目　录

第一章　HMM 基本理论

隐 Markov 模型(Hidden Markov Models，简称为 HMM)，作为语音信号的一种统计模型，今天正在语音处理各个领域中获得广泛的应用。而有关它的理论基础，却是在 1970 年前后由 Baum 等人建立起来的[30,31,32,33]，随后由 CMU 的 Baker 和 IBM 的 Jelinek 等人将其应用到语音识别之中[26,27,28,105,106]。由于 Bell 实验室 Rabiner 等人在 80 年代中期对 HMM 的深入浅出的介绍[144,101]，才逐渐使 HMM 为世界各国从事语音处理的研究人员所了解和熟悉，进而成为公认的一个研究热点[188,189]。本章是全书的基础，分四节介绍 HMM 的基本理论。首先在第一节中介绍 HMM 的基本思想，主要是回答什么是 HMM 这一问题，从 Markov 链着手，从分析有关 HMM 概念的实例开始，引出 HMM 的定义，并介绍 HMM 的参数。然后在第二节中给出 HMM 的基本算法，就是介绍将 HMM 应用到语音处理中经常会面临的三大基本问题的解决方案，亦即著名的 HMM 三大基本算法：前向-后向算法、Viterbi 算法和 Baum-Welch 算法。这些算法在本书以后的章节中会反复用到。接着，在第三节中介绍具体实现这些算法时应注意的一些问题，包括：初始模型选取；用多个观察值序列训练模型参数；为解决计算上的下溢问题而对算法加入比例因子的处理过程以及 Markov 链的形状。最后，第四节专门对 HMM 训练这一重要问题加以讨论，内容有：在训练数据不充分时的应付措施；怎样克服说话者的影响以及对经典训练算法加以改进的方法。

§1.1 HMM 基本思想

1.1.1 Markov 链

Markov 链是 Markov 随机过程的特殊情况，即 Markov 链是状态和时间参数都离散的 Markov 过程。从数学上，可以给出如下定义：

随机序列 X_n，在任一时刻 n，它可以处在状态 $\theta_1,\cdots,\theta_N$，且它在 $m+k$ 时刻所处的状态为 q_{m+k}的概率，只与它在 m 时刻的状态 q_m 有关，而与 m 时刻以前它所处状态无关，即有：

$$P(X_{m+k}=q_{m+k}/X_m=q_m,X_{m-1}=q_{m-1},\cdots,X_1=q_1)$$
$$=P(X_{m+k}=q_{m+k}/X_m=q_m)$$

其中，
$$q_1,q_2,\cdots,q_m,q_{m+k}\in(\theta_1,\theta_2,\cdots,\theta_N) \tag{1-1}$$

则称 X_n 为 Markov 链，并且称

$$P_{ij}(m,m+k)=P(q_{m+k}=\theta_j/q_m=\theta_i),$$
$$1\leqslant i,j\leqslant N,\quad m,k\text{ 为正整数} \tag{1-2}$$

为 k 步转移概率，当 $P_{ij}(m,m+k)$与 m 无关时，称这个 Markov 链为齐次 Markov 链，此时

$$P_{ij}(m,m+k)=P_{ij}(k) \tag{1-3}$$

以后若无特别申明，Markov 链就是指齐次 Markov 链。当 $k=1$ 时，$P_{ij}(1)$称为一步转移概率，简称为转移概率，记为 a_{ij}，所有转移概率 a_{ij}，$1\leqslant i,j\leqslant N$ 可以构成一个转移概率矩阵，即

$$\boldsymbol{A}=\begin{bmatrix}a_{11} & \cdots & a_{1N}\\ \cdots & & \cdots\\ a_{N1} & \cdots & a_{NN}\end{bmatrix} \tag{1-4}$$

且有
$$0\leqslant a_{ij}\leqslant 1 \tag{1-5}$$

$$\sum_{j=1}^{N}a_{ij}=1 \tag{1-6}$$

由于 k 步转移概率 $P_{ij}(k)$ 可由转移概率 a_{ij} 得到，因此，描述 Markov 链的最重要参数就是转移概率矩阵 A。但 A 矩阵还决定不了初始分布，即由 A 求不出 $q_1=\theta_i$ 的概率，这样，完全描述 Markov 链，除 A 矩阵之外，还必须引进初始概率矢量 $\pi=(\pi_1,\cdots,\pi_N)$，其中

$$\pi_i = P(q_1 = \theta_i), \quad 1 \leqslant i \leqslant N \tag{1-7}$$

显然有

$$0 \leqslant \pi_i \leqslant 1 \tag{1-8}$$

$$\sum_i \pi_i = 1 \tag{1-9}$$

实际中，Markov 链的每一状态可以对应于一个可观测到的物理事件。比如天气预测中的雨、晴、雪等，那么，这时它可称之为天气预报的 Markov 链模型。根据这个模型，可以算出各种天气(状态)在某一时刻出现的概率。

1.1.2 HMM 基本概念

HMM 是在 Markov 链的基础之上发展起来的。由于实际问题比 Markov 链模型所描述的更为复杂，观察到的事件并不是与状态一一对应，而是通过一组概率分布相联系，这样的模型就称为 HMM。它是一个双重随机过程，其中之一是 Markov 链，这是基本随机过程，它描述状态的转移。另一个随机过程描述状态和观察值之间的统计对应关系。这样，站在观察者的角度，只能看到观察值，不像 Markov 链模型中的观察值和状态一一对应，因此，不能直接看到状态，而是通过一个随机过程去感知状态的存在及其特性。因而称之为"隐"Markov 模型，即 HMM。现在来看一个著名的说明 HMM 概念的例子——球和缸(Ball and Urn)实验。

设有 N 个缸，每个缸中装有很多彩色的球，球的颜色由一组概率分布描述，如图 1.1 所示。实验是这样进行的，根据某个初始概率分布，随机地选择 N 个缸中的一个，例如第 i 个缸，再根据这个缸中彩色球颜色的概率分布，随机地选择一个球，记下球的颜

缸 1	缸 2	缸 N
$P(红)=b_{11}$	$P(红)=b_{21}$	$P(红)=b_{N1}$
$P(蓝)=b_{12}$	$P(蓝)=b_{22}$	$P(蓝)=b_{N2}$
$P(绿)=b_{13}$	$P(绿)=b_{23}$	$P(绿)=b_{N3}$
⋮	⋮	⋮
$P(黄)=b_{1M}$	$P(黄)=b_{2M}$	$P(黄)=b_{NM}$

图 1.1 球和缸实验

色，记为 O_1，再把球放回缸中，又根据描述缸的转移的概率分布，随机选择下一个缸，例如，第 j 个缸，再从缸中随机选一个球，记下球的颜色，记为 O_2，一直进行下去。可以得到一个描述球的颜色的序列 $O_1,O_2,\cdots$，由于这是观察到的事件，因而称之为观察值序列。但缸之间的转移以及每次选取的缸被隐藏起来了，并不能直接观察到。而且，从每个缸中选取球的颜色并不是与缸一一对应，而是由该缸中彩球颜色概率分布随机决定的。此外，每次选取哪个缸则由一组转移概率所决定。

1.1.3 HMM 定义

有了前面讨论的 Markov 链以及球和缸实验，就可以给出 HMM 的定义，或者说，一个 HMM 可以由下列参数描述：

a. N：模型中 Markov 链状态数目。记 N 个状态为 $\theta_1,\cdots,\theta_N$，记 t 时刻 Markov 链所处状态为 q_t，显然 $q_t\in(\theta_1,\cdots,\theta_N)$。在球与缸实验中的缸就相当于状态。

b. M：每个状态对应的可能的观察值数目。记 M 个观察值为 $V_1,\cdots,V_M$，记 t 时刻观察到的观察值为 O_t，其中 $O_t\in(V_1,\cdots,V_M)$。在球与缸实验中所选彩球的颜色，就是观察值。

c. π：初始状态概率矢量，$\pi=(\pi_1,\cdots,\pi_N)$，其中

$$\pi_i = P(q_1 = \theta_i), \quad 1 \leqslant i \leqslant N \tag{1-10}$$

在球与缸实验中指开始时选取某个缸的概率。

d. A：状态转移概率矩阵，$A=(a_{ij})_{N\times N}$，其中

$$a_{ij} = P(q_{t+1} = \theta_j / q_t = \theta_i), \quad 1 \leqslant i,j \leqslant N \tag{1-11}$$

在球与缸实验中指描述每次在当前选取的缸的条件下选取下一个缸的概率。

e. B：观察值概率矩阵，$B=(b_{jk})_{N\times M}$，其中

$$b_{jk} = P(O_t = V_k / q_t = \theta_j), \quad 1 \leqslant j \leqslant N, 1 \leqslant k \leqslant M \tag{1-12}$$

在球与缸实验中，b_{jk}就是第 j 个缸中球的颜色 k 出现的概率。

这样，可以记一个 HMM 为

$$\lambda = (N, M, \pi, A, B) \tag{1-13}$$

或简写为

$$\lambda = (\pi, A, B) \tag{1-14}$$

更形象地说，HMM 可分为两部分，一个是 Markov 链，由 π、A 描述，产生的输出为状态序列，另一个是一个随机过程，由 B 描述，产生的输出为观察值序列，如图 1.2 所示。T 为观察值时间长度。

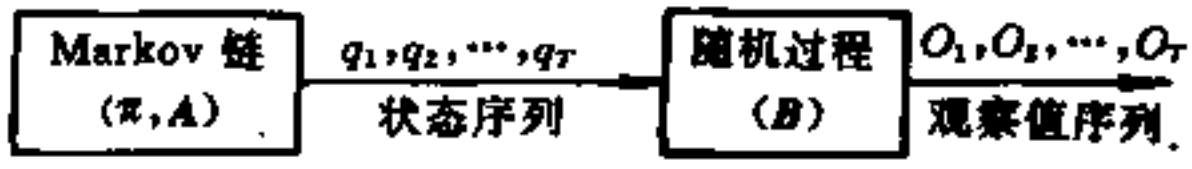


图 1.2　HMM 组成示意图

§1.2　HMM 基本算法

1.2.1　前向-后向算法

这个算法是用来计算给定一个观察值序列 $O=O_1, O_2, \cdots, O_T$ 以及一个模型 $\lambda=(\pi, A, B)$ 时，由模型 λ 产生出 O 的概率 $P(O/$

λ)。

根据图 1.2 所示 HMM 的组成，$P(O/\lambda)$最直接的求取方法如下：

对一个固定的状态序列 $S=q_1,q_2,\cdots,q_T$，有

$$P(O/S,\lambda)=\prod_{t=1}^{T}P(O_t/q_t,\lambda)=b_{q_1}(O_1)b_{q_2}(O_2)\cdots b_{q_T}(O_T) \tag{1-15}$$

其中
$$b_{q_t}(O_t)=b_{jk}|_{q_t=\theta_j,O_t=v_k},1\leqslant t\leqslant T \tag{1-16}$$

而对给定 λ，产生 S 的概率为

$$P(S/\lambda)=\pi_{q_1}a_{q_1q_2}\cdots a_{q_{T-1}q_T} \tag{1-17}$$

因此，所求概率为

$$P(O/\lambda)=\sum_{所有S}P(O/S,\lambda)P(S/\lambda)$$

$$=\sum_{q_1,q_2,\cdots,q_T}\pi_{q_1}b_{q_1}(O_1)a_{q_1q_2}b_{q_2}(O_2)\cdots a_{q_{T-1}q_T}b_{q_T}(O_T) \tag{1-18}$$

显而易见，上式的计算量是十分惊人的，大约为 $2TN^T$ 数量级，当 $N=5,T=100$ 时，计算量达 10^{72}，这是完全不能接受的。在此情况下，要求出 $P(O/\lambda)$还必须寻求更有效的算法，这就是 Baum 等人提出的前向-后向算法：

(1)前向算法

定义前向变量为

$$\alpha_t(i)=P(O_1,O_2,\cdots,O_t,q_t=\theta_i/\lambda),\quad 1\leqslant t\leqslant T \tag{1-19}$$

那么，有

a. 初始化：
$$\alpha_1(i)=\pi_i b_i(O_1),\quad 1\leqslant i\leqslant N \tag{1-20}$$

b. 递归：
$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}),$$
$$1\leqslant t\leqslant T-1,1\leqslant j\leqslant N \tag{1-21}$$

c. 终结：
$$P(O/\lambda)=\sum_{i=1}^{N}\alpha_T(i) \tag{1-22}$$

其中 $$b_j(O_{t+1}) = b_{jk}|_{O_{t+1}=v_k} \tag{1-23}$$

这种算法计算量大为减少，变为 $N(N+1)(T-1)+N$ 次乘法和 $N(N-1)(T-1)$ 次加法。同样，$N=5, T=100$ 时，只需大约 3 000 次计算(乘法)。这种算法是一种典型的格型结构，如图 1.3 所示。

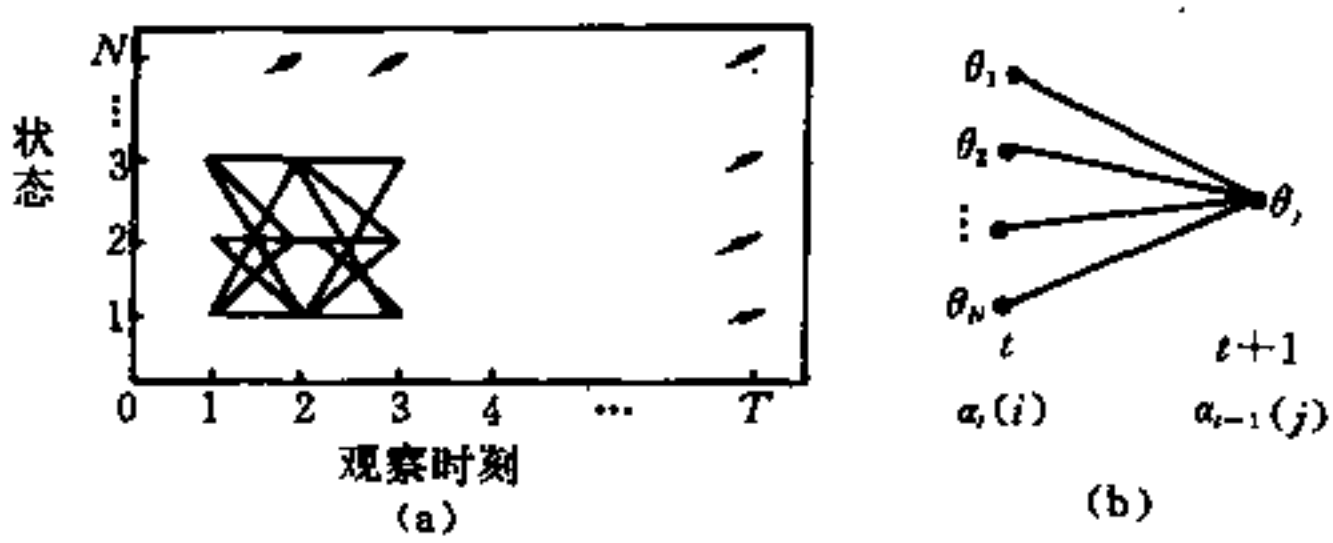


图 1.3 前向算法示意图

(a) 格型结构； (b) t 时刻递归关系

(2)后向算法

与前向算法类似，定义后向变量为

$$\beta_t(i) = P(O_{t-1}, O_{t+2}, \cdots, O_T / q_t = \theta_i, \lambda), \quad 1 \leqslant t \leqslant T-1,$$

其中 $$\beta_T(i) = 1 \tag{1-24}$$

类似，有

a. 初始化： $$\beta_T(i)=1, \quad 1 \leqslant i \leqslant N \tag{1-25}$$

b. 递归： $$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j),$$

$$t=T-1, T-2, \cdots, 1, \quad 1 \leqslant i \leqslant N \tag{1-26}$$

c. 终结： $$P(O/\lambda)=\sum_{i=1}^{N} \beta_1(i) \tag{1-27}$$

后向算法的计算量大约在 N^2T 数量级，也是一种格型结构。

1.2.2 Viterbi 算法

这个算法解决了给定一个观察值序列 $O=O_1, O_2, \cdots, O_T$ 和一

个模型 $\lambda=(\pi,A,B)$，在最佳的意义上确定一个状态序列 $Q^*=q_1^*,q_2^*,\cdots,q_T^*$ 的问题。

"最佳"的意义有很多种，由不同的定义可得到不同的结论。这里讨论的最佳意义上的状态序列 Q^*，是指使 $P(Q,O/\lambda)$[①] 最大时确定的状态序列 Q^*。Viterbi 算法可以叙述如下：

定义 $\delta_t(i)$为时刻 t 时沿一条路径 $q_1,q_2,\cdots,q_t$，且 $q_t=\theta_i$，产生出 $O_1,O_2,\cdots,O_t$ 的最大概率，即有

$$\delta_t(i)=\max_{q_1,q_2,\cdots,q_{t-1}} P(q_1,q_2,\cdots,q_t,q_t=\theta_i,O_1,O_2,\cdots,O_t/\lambda) \tag{1-28}$$

那么，求取最佳状态序列 Q^* 的过程为

a. 初始化：
$$\delta_1(i)=\pi_i b_i(O_1),\quad 1\leqslant i\leqslant N \tag{1-29}$$
$$\varphi_1(i)=0,\quad 1\leqslant i\leqslant N \tag{1-30}$$

b. 递归：
$$\delta_t(j)=\max_{1\leqslant i\leqslant N}[\delta_{t-1}(i)a_{ij}]b_j(O_t),2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{1-31}$$
$$\varphi_t(j)=\operatorname{argmax}_{1\leqslant i\leqslant N}[\delta_{t-1}(i)a_{ij}]^{②},2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{1-32}$$

c. 终结：
$$P^*=\max_{1\leqslant i\leqslant N}[\delta_T(i)] \tag{1-33}$$
$$q_T^*=\operatorname{argmax}_{1\leqslant i\leqslant N}[\delta_T(i)] \tag{1-34}$$

d. 状态序列求取：
$$q_t^*=\varphi_{t+1}(q_{t+1}^*),\qquad t=T-1,T-2,\cdots,1 \tag{1-35}$$

应当指出，Viterbi 算法的一个副产品 $P^*=\max_Q P(Q,O/\lambda)$和前向-后向算法计算出的$P(O/\lambda)=\sum_Q P(Q,O/\lambda)$之间的关系为

对语音处理应用而言，$P(Q,O/\lambda)$动态范围很大，或者说不同

---

① 此处及本书以后部分，为方便起见，将 $P(Q/O,\lambda)$记为 $P(Q,O/\lambda)$。

② 符号说明：如果 $i=I$ 时，$f(i)$达到最大值，那么 $I=\operatorname{argmax}_{1\leqslant i\leqslant N}[f(i)]$。

的 Q 使 $P(Q,O/\lambda)$的值差别很大，而 $\max_Q P(Q,O/\lambda)$事实上是 $\sum_Q P(Q,O/\lambda)$中举足轻重的唯一成分，因此，常常等价地使用 $\max_Q P(Q,O/\lambda)$和 $\sum_Q P(Q,O/\lambda)$，那么，Viterbi 算法也能用来计算 $P(O/\lambda)$。

此外，上述的 Viterbi 算法也是一种格型结构，而且类似于前向算法 。同样，由后向算法的思想出发，亦可推导出 Viterbi 算法的另一种实现方式。

1.2.3 Baum-Welch 算法

这个算法实际上是解决 HMM 训练，即 HMM 参数估计问题，或者说，给定一个观察值序列 $O=O_1,O_2,\cdots,O_T$，该算法能确定一个 $\lambda=(\pi,A,B)$，使 $P(O/\lambda)$最大。

显然，由(1-19)和(1-24)式定义的前向和后向变量，有：

$$P(O/\lambda)=\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j),\qquad 1\leqslant t\leqslant T-1 \tag{1-36}$$

这里，求取 λ，使 $P(O/\lambda)$最大，是一个泛函极值问题。但是，由于给定的训练序列有限，因而不存在一个最佳的方法来估计 λ。在这种情况下，Baum-Welch 算法利用递归的思想，使$P(O/\lambda)$局部极大，最后得到模型参数 $\lambda=(\pi,A,B)$。此外，用梯度方法也可达到类似目的。

定义 $\xi_t(i,j)$为给定训练序列 O 和模型 λ 时，时刻 t 时 Markov 链处于 θ_i 状态和时刻 $t+1$ 为 θ_j 状态的概率，即

$$\xi_t(i,j)=P(O,q_t=\theta_i,q_{t+1}=\theta_j/\lambda) \tag{1-37}$$

可以推导出：

$$\xi_t(i,j)=[\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)]/P(O/\lambda) \tag{1-38}$$

那么，时刻 t 时 Markov 链处于 θ_i 状态的概率为：

$$\xi_t(i) = P(O, q_t = \theta_i/\lambda) = \sum_{j=1}^{N} \xi_t(i,j)$$
$$= \alpha_t(i)\beta_t(i)/P(O/\lambda) \tag{1-39}$$

因此，$\sum_{t=1}^{T-1}\xi_t(i)$表示从 θ_i 状态转移出去的次数的期望值，而$\sum_{t=1}^{T-1}\xi_t(i,j)$表示从 θ_i 状态转移到 θ_j 状态的次数的期望值。由此，导出了 Baum-Welch 算法中著名的重估(reestimation)公式：

$$\bar{\pi}_i = \xi_1(i) \tag{1-40}$$

$$\bar{a}_{ij} = \sum_{t=1}^{T-1}\xi_t(i,j) \Big/ \sum_{t=1}^{T-1}\xi_t(i) \tag{1-41}$$

$$\bar{b}_{jk} = \sum_{\substack{t=1 \\ \text{且}O_t=v_k}}^{T}\xi_t(j) \Big/ \sum_{t=1}^{T}\xi_t(j) \tag{1-42}$$

具体推导过程见附录。那么，HMM 参数 $\lambda=(\pi,A,B)$的求取过程为：根据观察值序列 O 和选取的初始模型 $\lambda=(\pi,A,B)$，由重估公式(1-40)、(1-41)和(1-42)式，求得一组新参数 $\bar{\pi}_i$，$\bar{a}_{ij}$和 $\bar{b}_{jk}$，亦即得到了一个新的模型 $\bar{\lambda}=(\bar{\pi},\bar{A},\bar{B})$，可以证明，$P(O/\bar{\lambda})>P(O/\lambda)$，即由重估公式得到的 $\bar{\lambda}$ 比 λ 在表示观察值序列 O 方面要好。那么，重复这个过程，逐步改进模型参数，直到 $P(O/\bar{\lambda})$收敛，即不再明显增大，此时的 $\bar{\lambda}$ 即为所求之模型。

应当指出，HMM 训练，或称参数估计问题，是 HMM 在语音处理中应用的关键问题，与前面讨论的两个问题相比，这也是最困难的一个问题，Baum-Welch 算法只是得到广泛应用的解决这一问题的经典方法，但并不是唯一的，也远不是最完善的方法，本章最后一节还要对此问题作进一步讨论。

§1.3 HMM 算法实现中的问题

1.3.1 初始模型选取

根据 Baum-Welch 算法由训练数据得到 HMM 参数时，如重估公式(1-40)、(1-41)和(1-42)式所示，一个重要问题就是初始模型的选取。不同的初始模型将产生不同的训练结果。因为算法是使 $P(O/\lambda)$局部极大时得到的模型参数，因此，选取好的初始模型，使最后求出的局部极大与全局最大接近，是很有意义的。但是，至今这个问题仍没有完美的答案。实际处理时都是采用一些经验方法。一般认为[203]，π 和 A 参数初值选取影响不大，可以随机选取或均匀取值，只要满足(1-5)、(1-6)、(1-8)和(1-9)式要求的约束条件即可。但 B 的初值对训练出的 HMM 影响较大，一般倾向采取较为复杂的初值选取方法。基于这种考虑，一种典型的 HMM 参

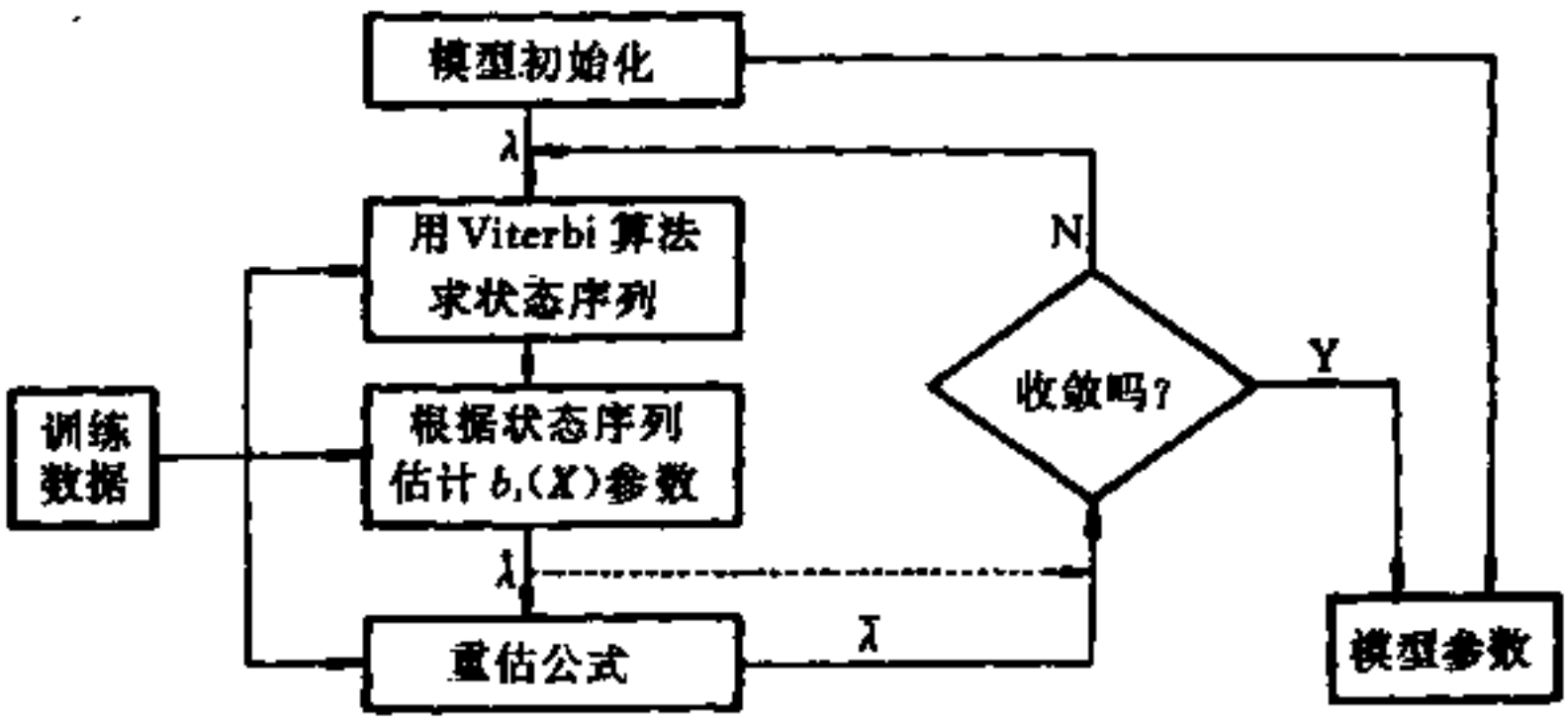


图 1.4 一种 HMM 参数估计方法示意图

数估计过程如图 1.4 所示[113,189,192]。这里，初始模型 λ 可以任意选取。但因为有 $P(O/\bar{\lambda})>P(O/\lambda)$，所以 $\bar{\lambda}$ 是 λ 改进后的模型。再将 $\bar{\lambda}$ 作为初值用重估公式，得到 $\bar{\bar{\lambda}}$。这样就避免了初值的选择不当，变

经典的 $\lambda\to\bar{\lambda}$ 为 $\lambda\to\hat{\lambda}\to\bar{\lambda}$。当然，沿图中虚线，不用重估公式，$\hat{\lambda}$ 也可近似作为模型参数。

当然，从以后的讨论会看到，HMM 有很多类型。因此，针对不同形式的 HMM，也可采取不同的有效的初值选取方法。

1.3.2 多个观察值序列训练

实际中，训练一个 HMM，经常是用到不止一个观察值序列，那么，对于 L 个观察值序列训练 HMM 时，要对 Baum-Welch 算法的重估公式(1-40)、(1-41)和(1-42)式加以修正。设 L 个观察值序列为 $O^{(l)}, l=1,\cdots,L$，其中 $O^{(l)}=O_1^{(l)},O_2^{(l)},\cdots,O_{T_l}^{(l)}$，假定各个观察值序列独立，此时，

$$P(O/\lambda)=\prod_{l=1}^{L}P(O^{(l)}/\lambda) \tag{1-43}$$

由于重估公式是以不同事件的频率为基础的，因此，对 L 个训练序列，重估公式修正为

$$\bar{\pi}_i=\sum_{l=1}^{L}\alpha_1^{(l)}(i)\beta_1^{(l)}(i)/P(O^{(l)}/\lambda),\quad 1\leqslant i\leqslant N \tag{1-44}$$

$$\bar{a}_{ij}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\alpha_t^{(l)}(i)a_{ij}b_j(O_{t+1}^{(l)})\beta_{t+1}^{(l)}(j)/P(O^{(l)}/\lambda)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\alpha_t^{(l)}(i)\beta_t^{(l)}(i)/P(O^{(l)}/\lambda)},$$

$$1\leqslant i,j\leqslant N \tag{1-45}$$

$$\bar{b}_{jk}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{(l)}(j)\beta_t^{(l)}(j)/P(O^{(l)}/\lambda)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{(l)}(j)\beta_t^{(l)}(j)/P(O^{(l)}/\lambda)},\quad \text{其中 } O_t=V_k$$

$$1\leqslant j\leqslant N,1\leqslant k\leqslant M \tag{1-46}$$

1.3.3 比例因子问题

在前向-后向算法和 Baum-Welch 算法中，都有 $\alpha_t(i)$和 $\beta_t(i)$的递归计算，因为所有量都小于 1，因此，$\alpha_t(i)$（随着 t 的增加）和 $\beta_t(i)$（随着 t 的减少）都迅速趋向于零，为了解决这种下溢(underflow)问题，必须采取增加比例因子(scaling)的方法，对有关算法加以修正，处理过程为[113,144]：

(1)对 α 的处理

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \leqslant i \leqslant N \tag{1-47}$$

$$\alpha_1^*(i) = \frac{\alpha_1(i)}{\sum_{i=1}^{N}\alpha_1(i)} \triangleq \frac{\alpha_1(i)}{\Phi_1}, \qquad 1 \leqslant i \leqslant N \tag{1-48}$$

$$\tilde{\alpha}_{t+1}(j) = \Big[\sum_{i=1}^{N}\alpha_t^*(i)a_{ij}\Big]b_j(O_{t+1}),$$

$$1 \leqslant j \leqslant N, t = 1,2,\cdots,T-1 \tag{1-49}$$

$$\alpha_{t+1}^*(j) = \tilde{\alpha}_{t+1}(j)/\sum_{j=1}^{N}\tilde{\alpha}_{t+1}(j) \triangleq \tilde{\alpha}_{t+1}(j)/\Phi_{t+1},$$

$$1 \leqslant j \leqslant N, t = 1,2,\cdots,T-1 \tag{1-50}$$

(2)对 β 的处理

$$\beta_T(i) = 1, \quad 1 \leqslant i \leqslant N \tag{1-51}$$

$$\beta_T^*(i) = 1, \quad 1 \leqslant i \leqslant N \tag{1-52}$$

$$\tilde{\beta}_t(i) = \sum_{j=1}^{N}a_{ij}b_j(O_{t+1})\beta_{t+1}^*(j),$$

$$1 \leqslant i \leqslant N, t = T-1,\cdots,1 \tag{1-53}$$

$$\beta_t^*(i) = \tilde{\beta}_t(i)/\Phi_{t+1},$$

$$1 \leqslant i \leqslant N, t = T-1,\cdots,1 \tag{1-54}$$

(3)常用计算公式的处理

对 α 和 β 做了上述处理之后，为了保持原有公式计算之结果不变，必须在常用计算公式中做相应处理，以消去比例因子的影响。

a. 概率 $P(O/\lambda)$的计算公式

由 α 的处理过程易推出：

$$\alpha_t^*(i) = \alpha_t(i)/\Phi_1\Phi_2\cdots\Phi_t \tag{1-55}$$

而
$$\Phi_t = \sum_{j=1}^{N}\bar{\alpha}_t(j) = \sum_{j=1}^{N}\Big[\sum_{i=1}^{N}\alpha_{t-1}^*(i)a_{ij}\Big]b_j(O_t)$$

$$= \sum_{j=1}^{N}\alpha_t(j)/\Phi_1\Phi_2\cdots\Phi_{t-1} \tag{1-56}$$

因此
$$\sum_{j=1}^{N}\alpha_t(j) = \Phi_1\Phi_2\cdots\Phi_t \tag{1-57}$$

即
$$P(O/\lambda) = \sum_{j=1}^{N}\alpha_T(j) = \Phi_1\Phi_2\cdots\Phi_T \tag{1-58a}$$

或
$$\lg P(O/\lambda) = \sum_{t=1}^{T}\lg\Phi_t \tag{1-58b}$$

b. 重估公式

由 β 的处理易知

$$\beta_t^*(i) = \frac{\beta_t(i)}{\Phi_{t+1}\Phi_{t+2}\cdots\Phi_T} \tag{1-59}$$

因此，重估公式(多个训练序列)时，(1-44)、(1-45)和(1-46)式变为

$$\bar{\pi}_i = \sum_{l=1}^{L}\alpha_1^{*(l)}(i)\beta_1^{*(l)}(i),$$

$$1 \leqslant i \leqslant N \tag{1-60}$$

$$\bar{a}_{ij} = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\alpha_t^{*(l)}(i)a_{ij}b_j(O_{t+1}^{(l)})\beta_{t+1}^{*(l)}(j)/\Phi_{t+1}}{\sum_{l=1}^{L}\sum_{t=1}^{T_l-1}\alpha_t^{*(l)}(i)\beta_t^{*(l)}(i)},$$

$$1 \leqslant i, j \leqslant N \tag{1-61}$$

$$\bar{b}_{jk}=\frac{\sum_{l=1}^{L}\sum_{\substack{t=1\\ \text{且}O_t=V_k}}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)},$$

$$1\leqslant j\leqslant N,1\leqslant k\leqslant M \tag{1-62}$$

(4)Viterbi 算法的处理

在原来的 Viterbi 算法(1-28)～(1-35)式中加入对数化处理即可,即

定义

$$\delta_t(i)=\max_{q_1,q_2,\cdots,q_{t-1}}\lg P(q_1,q_2,\cdots,q_t,q_t=\theta_i,O_1,O_2,\cdots,O_t/\lambda) \tag{1-63}$$

那么,初始化(1-29)式变为

$$\delta_1(i)=\lg\pi_i+\lg b_i(O_1),\qquad 1\leqslant i\leqslant N \tag{1-64}$$

递归运算(1-31)式变为

$$\delta_t(i)=\max_{1\leqslant i\leqslant N}[\delta_{t-1}(i)+\lg a_{ij}]+\lg[b_j(O_t)] \tag{1-65}$$

终结(1-33)式变为

$$\lg P^*=\max_{1\leqslant i\leqslant N}[\delta_T(i)] \tag{1-66}$$

这样,得到的是 P^* 的对数值而不是 P^*。应该指出,实际上为了避免计算出的概率值 $P(O/\lambda)$ 太小,而总是采用 $\lg P(O/\lambda)$。事实上,单个的概率值与通常意义上的概率有些区别,在以后的讨论中将会看到,在大多数语音处理应用场合中,单个具体的 $P(O/\lambda)$ 的值总是很小很小。因此,单个的 $P(O/\lambda)$ 数值本身并无多大意义,通常在应用中只是几个 $P(O/\lambda)$ 的相对大小才是有用的。

1.3.4 Markov 链的形状

如图 1.2 所示,HMM 由两部分组成,其一为 Markov 链,它由 π、A 描述,显然,不同的 π、A,决定了 Markov 链不同的形状。几种典型的 Markov 链如图 1.5 所示。它们各具特色。图 1.5(a)所

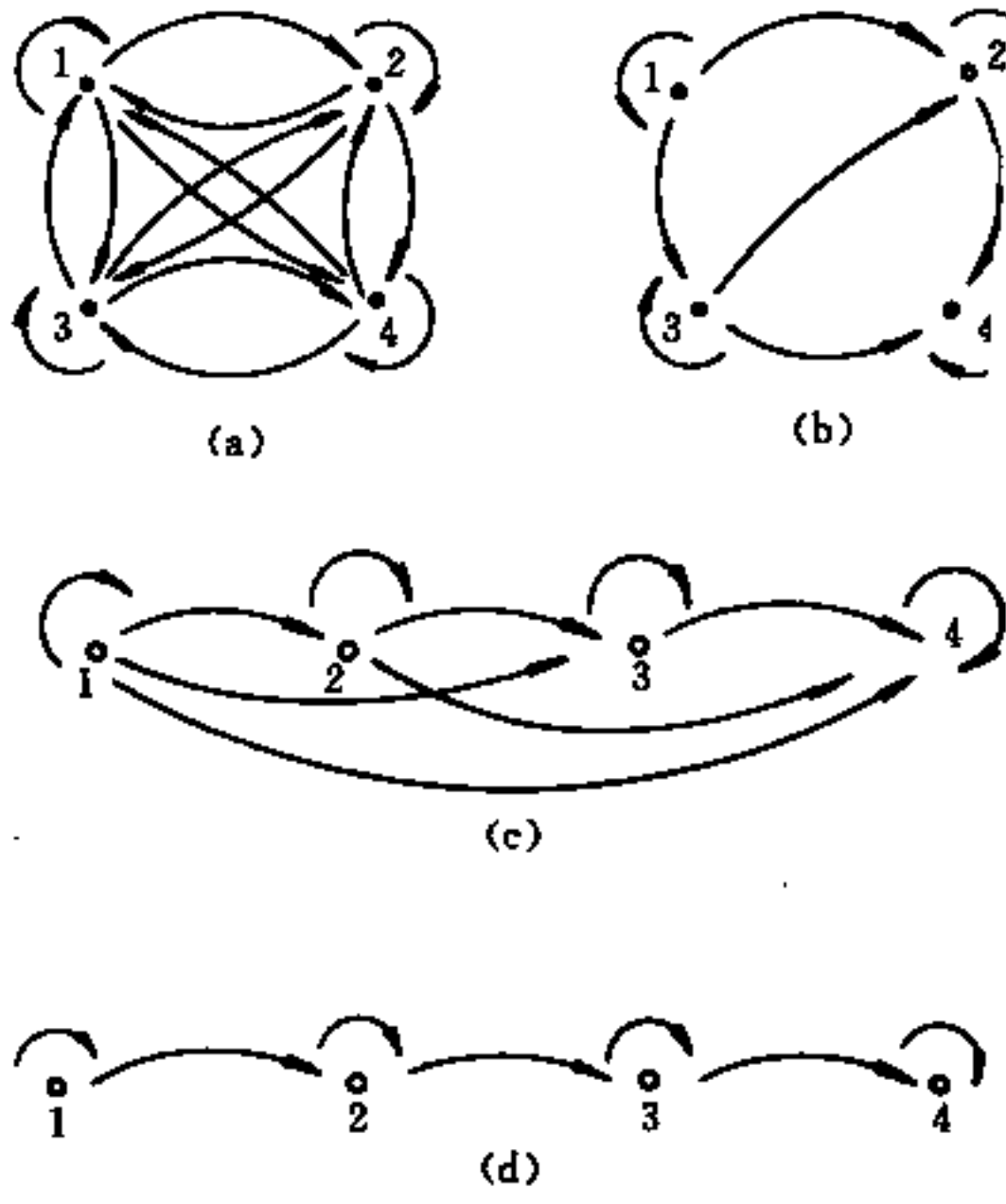


图 1.5　几种典型 Markov 链示意图($N=4$)

(a) A 矩阵没有零值的 Markov 链；

(b) A 矩阵有零值的 Markov 链；

(c)、(d) 左-右形式 Markov 链

示 Markov 链从任一状态出发，在下一时刻可到达任一状态，对应于 A 矩阵没有零值。图 1.5(b)所示 Markov 链则有些不同，比如，从状态 1 出发，下一时刻不可能到达状态 4，也就是说，A 矩阵含有零元素。图 1.5(c)和(d)是两种特殊的 Markov 链，其特点为：必定从状态 1 出发，沿状态序号增加的方向转移，最终停在状态 4。由这种 Markov 链构成的 HMM，一般称之为左-右模型(left-to-

right models)，在实际语音处理应用中被广泛采用，尤其是孤立词识别[144]。图 1.6 也是一个左-右模型，但它从左到右有多条转移路途，这种 Markov 链形状的 HMM 在连续语音识别中有成功的应用[148]。

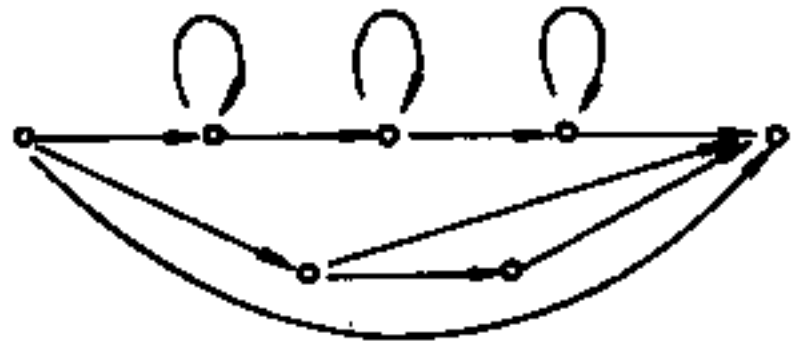


图 1.6　一种特殊 Markov 链示意图

§1.4　关于 HMM 训练的几点考虑

1.4.1　克服训练数据的不足

根据 HMM 的定义，一个 $\lambda=(\pi,A,B)$ 含有很多个待估计的参数，因此，为了得到满意的模型，必须要有很多的训练数据，这在实际中往往难以办到。另一方面，选择规模更小的模型，即减少 N 和 M 的值，也有实际的困难。因此，在很多场合中，都要对两个或多个针对同一事件的表示不同程度的细节和鲁棒性的模型进行合并。比如说，通常一些出现次数很少的观察值矢量没有包含在整个训练数据中，这样训练出的 HMM 参数中就会有不少为零的概率值。而事实上，在实际语音识别测试时，这些观察值矢量又可能出现，因而需要对训练好的模型进行平滑处理。而一般为了避免过分平滑，总是要合并平滑前后的模型参数，以获得较好的结果模型。

合并两个 HMM 的问题可以表述为

$$\lambda = w\lambda_1 + (1-w)\lambda_2 \tag{1-67}$$

其中，$\lambda=(\pi,A,B)$ 为结果模型，$\lambda_1=(\pi_1,A_1,B_1)$ 和 $\lambda_2=(\pi_2,A_2,$

B_2)为待合并的两个模型，$0 \leqslant w \leqslant 1$ 为合并时的系数。因此，问题的关键就是合并权值 w 的估计。

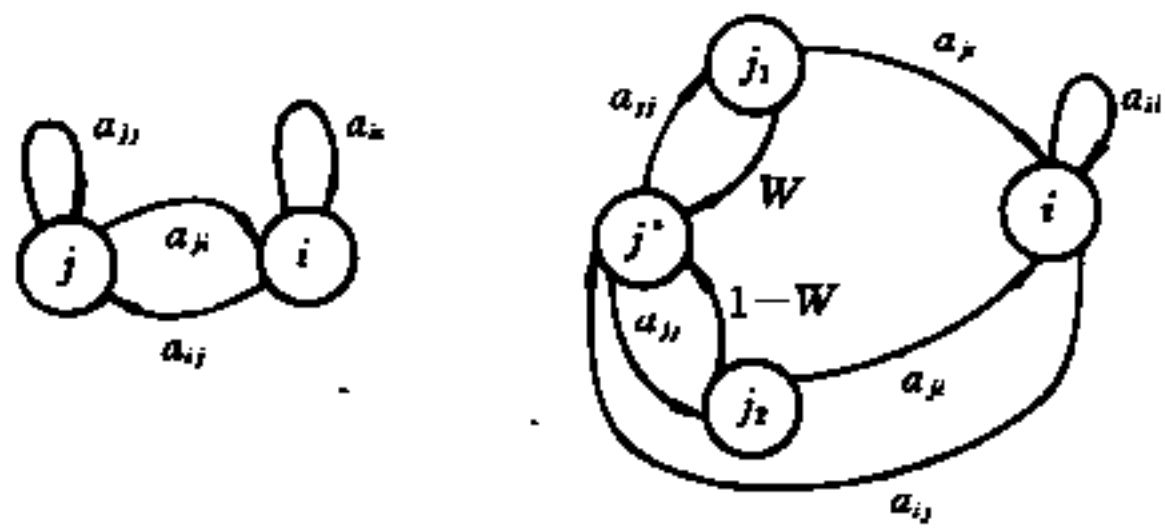
一种可能的方法是人工选择权值 w。Schwartz 等人就是根据训练量的多少和每个模型中概率分布的合适程度来人工选取权值的，并在语音识别中取得了成功[211]。但这种方法的局限也是明显的：过分依赖于人的经验判断，而且工作量也很大。

另一种估计 w 的方法就是著名的消去内插法(deleted interpolation)。这种方法最早由 Jelinek 提出[107]，随后被广泛使用在基于 HMM 的语音识别系统中[132,135]，它的基本思想为：

设 b_{jk}^1和 b_{jk}^2为 λ_1 和 λ_2 模型中状态 j 对应的观察值概率，b_{jk}为 λ 中状态 j 对应的观察值概率，那么，由(1-67)式有

$$b_{jk} = wb_{jk}^1 + (1-w)b_{jk}^2 \tag{1-68}$$

上式可以理解为 λ 模型中状态 j 被三个状态 j^*、j_1 和 j_2 所取代，



(a)　　(b)

图 1.7　消去内插法示意图

(a)λ 中状态 j，输出观察值概率为 b_{jk}；

(b)取代状态 j 的三个状态 j^*，j_1 和 j_2

如图 1.7 所示。其中，状态 j^* 无输出观察值概率，状态 j_1 和 j_2 的输出观察值概率分别为 b_{jk}^1和 b_{jk}^2，但从状态 j^* 转移到状态 j_1 和 j_2 的概率分别为 w 和 $1-w$，但不占用时间(这种转移称为零转移

(null transition))。那么，估计权值 w 的问题就转化为一个典型的 HMM 问题，因此，由 HMM 训练算法就可直接估计出权值 w。

但消去插值法的核心是使权值 w 的估计对未来的数据仍然有价值。因此，要求用以估计 λ_1 和 λ_2 的数据和用以估计 w 的数据不同。为了实现这个目的，一个合理的处理方法就是：将所有训练数据分成几部分，消去一部分数据用来估计 w，其余的数据训练 λ_1 和 λ_2。由于这种对总的训练数据的划分有很多种方式，由此得到很多 w 值，再用一个循环递归处理，可以求出所需的权值 w。

由上述讨论可知，消去插值法计算量很大。这是其不足之处。虽然近来有人给出了快速算法[24]，但在实际中完整地实现消去插值法来估计权值 w 仍然相当困难。另一方面，对合并两个模型，只用一个权值，也不是最好的选择。一个更好的合并方式是对模型中每个状态都选定一个权值。事实上，实际应用消去插值法时，也是将它修正简化，一方面减少计算量，另一方面，对每个状态都估计出一个权值[132,135]。

另外，从 Baum-Welch 算法的重估公式可以推导出一种 HMM 相对可靠性度量方法，这样，就可以得到待合并的两个或多个模型各自的相对可靠程度，由此确定合并时的权值。这种估计权值的方法可以简述为[242]：

根据重估公式，考虑有 L 个观察值序列训练模型 $\lambda=(\pi,A,B)$，于是有

$$\bar{a}_{ij}=\frac{\text{从状态 } i \text{ 到 } j \text{ 的转移次数的期望值}}{\text{从状态 } i \text{ 转移出去的次数的期望值}}$$

$$=\frac{\sum_{l=1}^{L}\text{第 } l \text{ 个训练序列从状态 } i \text{ 到 } j \text{ 的转移次数}}{\sum_{l=1}^{L}\text{第 } l \text{ 个训练序列状态 } i \text{ 的状态数目}}$$

$$\triangleq\frac{\sum_{l=1}^{L}\text{trans-counts}(i,j,l)}{\sum_{l=1}^{L}\text{state-counts}(i,l)} \tag{1-69}$$

$$\bar{b}_{jk} = \frac{\text{位于状态 } j \text{ 且观察到输出矢量 } k \text{ 的次数的期望值}}{\text{位于状态 } j \text{ 的次数的期望值}}$$

$$= \frac{\sum_{l=1}^{L} \text{第 } l \text{ 个训练序列位于状态 } j \text{ 输出矢量为 } k \text{ 的矢量个数}}{\sum_{l=1}^{L} \text{第 } l \text{ 个训练序列位于状态 } j \text{ 的状态数目}}$$

$$\triangleq \frac{\sum_{l=1}^{L} \text{vect-counts}(k,j,l)}{\sum_{l=1}^{L} \text{state-counts}(j,l)} \tag{1-70}$$

注意到：

$$\text{state-counts}(j,l)\big|_{j=i} = \text{state-counts}(i,l) \tag{1-71}$$

这样，(1-69)和(1-70)式可重写为

$$\bar{a}_{ij} = \sum_{l=1}^{L} R_{il} \cdot \frac{\text{trans-counts}(i,j,l)}{\text{state-counts}(i,l)} = \sum_{l=1}^{L} R_{il} \cdot \bar{a}_{ijl} \tag{1-72}$$

$$\bar{b}_{jk} = \sum_{l=1}^{L} R_{jl} \cdot \frac{\text{vect-counts}(k,j,l)}{\text{state-counts}(j,l)} = \sum_{l=1}^{L} R_{jl} \cdot \bar{b}_{jkl} \tag{1-73}$$

其中

$$R_{jl} = \frac{\text{state-counts}(j,l)}{\sum_{l'=1}^{L} \text{state-counts}(j,l')} \tag{1-74}$$

分析(1-72)和(1-73)式可知，当用 L 个训练序列获取 HMM 参数时，在每次迭代时，可以分别用每个训练序列获取相应的 HMM 参数，再加以合并，而且，合并的权值仅仅取决于状态数目。因此，可以认为，正是状态数目描述了 HMM 的相对可靠程度。因此，当需要合并 L 个 HMM 时，对任一状态 j，合并的权值可由(1-74)式求出。

由于这种估计权值的方法由 Baum-Welch 算法中重估公式导出，因而在最大似然意义上是(局部)最佳的，而且，对每个状态都选取一个合并的权值，而不是对整个待合并的 HMM 选取权值，这样使合并的结果模型更好。显然，使用这种方法估计权值，在训练各个 HMM 时，除了保存模型参数之外，还应保存相应的状态

数目，因此，需占用较多的存储空间。这种方法得到了实验支持[242]。

1.4.2 处理说话者的影响

由于语音动态范围很大，不同的说话者的语音，甚至同一说话者在不同时间和场合说的话，都有较大的不同，因此，训练 HMM 时，充分考虑到说话者的影响，对于估计出好的 HMM 参数是十分重要的。这个问题在这里可以表述为：设有训练数据集 D_A 所训练出的模型 $\lambda=(\pi,A,B)$，从训练过程可知，λ 反映了 D_A 的特性。如果又新增加了一个训练数据集 D_B，我们希望经过一个处理过程，使 D_B 的特性也能反映在结果模型之中。D_B 相对于 D_A 来说，可以是不同说话者的语音，也可以是同一说话者在不同的时间所发的语音，因此，这个问题对语音识别，尤其是非特定人语音识别是很有意义的。

根据 Baum-Welch 算法，一个直接的处理方法就是用 D_A 和 D_B 一起，重新训练一个模型。但这样做，一方面不经济，没有利用已训练好的模型 λ 的信息，另一方面，实际实现起来有困难，因为在很多实际场合中并没有保留训练数据集 D_A，而只保存了反映其特性的而且占用少得多存储空间的模型 λ。那么，另一个简单的处理方法容易想到：以 λ 为初始模型，用数据 D_B，通过重估公式进行若干次迭代，得到新模型 λ''。但显然，这个 λ'' 只能反映数据集 D_B 的特性，而不可能同时反映出 D_A 的特性。

针对这个问题，经过分析 Baum-Welch 算法，我们给出了一种处理说话者影响的方法，它在小词汇表语音识别[264]和大词汇表语音识别中都有成功的应用[78,250]，其基本思想为：

由重估公式(1-69)和(1-70)式可知：在迭代中，L 个训练序列的信息，是由 L 个序列分别计算出的转移次数、矢量数、状态数目通过分子分母分别相加来反映在迭代后的新模型参数之中，那么，将 L 个训练序列分成两个训练数据集，不妨也记为 D_A 和 D_B，其

中所含训练序列数分别为 L_1 和 L_2，显然 $L_1+L_2=L$，那么，改写重估公式(1-69)和(1-70)式，有

$$\bar{a}_{ij}=\frac{\sum_{l_1=1}^{L_1}\text{trans-counts}^{(A)}(i,j,l_1)+\sum_{l_2=1}^{L_2}\text{trans-counts}^{(B)}(i,j,l_2)}{\sum_{l_1=1}^{L_1}\text{state-counts}^{(A)}(i,l_1)+\sum_{l_2=1}^{L_2}\text{state-counts}^{(B)}(i,l_2)}$$

$$=\frac{\text{trans-counts}^{(A)}+\text{trans-counts}^{(B)}}{\text{state-counts}^{(A)}+\text{state-counts}^{(B)}} \tag{1-75}$$

$$\bar{b}_{jk}=\frac{\sum_{l_1=1}^{L_1}\text{vect-counts}^{(A)}(k,j,l_1)+\sum_{l_2=1}^{L_2}\text{vect-counts}^{(B)}(k,j,l_2)}{\sum_{l_1=1}^{L_1}\text{state-counts}^{(A)}(j,l_1)+\sum_{l_2=1}^{L_2}\text{state-counts}^{(B)}(j,l_2)}$$

$$=\frac{\text{vect-counts}^{(A)}+\text{vect-counts}^{(B)}}{\text{state-counts}^{(A)}+\text{state-counts}^{(B)}} \tag{1-76}$$

这样，说话者的影响可处理如下：在得到训练数据集 D_A、训练产生模型 λ 时，不仅保存 $\lambda=(\pi,A,B)$的结果参数，还保存相应的转移次数、矢量数和状态数目，即 $\text{trans-counts}^{(A)}$、$\text{vect-counts}^{(A)}$ 和 $\text{state-counts}^{(A)}$。在得到新的训练数据集 D_B 时，以 λ 为初始模型，得到新的模型 λ_B 以及新的转移次数、矢量数和状态数目，即 $\text{trans-counts}^{(B)}$、$\text{vect-counts}^{(B)}$ 和 $\text{state-counts}^{(B)}$，那么，由(1-75)式和(1-76)式求出的 $\bar{a}_{ij}$和 $\bar{b}_{jk}$，就是既反映了数据集 D_A 的特性、又反映了数据集 D_B 的特性的所求模型 λ^* 的参数。

此外，这种处理说话者影响的方法还可推广为一种 HMM 模块化的训练方法[266]：对于几个训练数据集，如 D_A、D_B 和 D_C，由 Baum-Welch 算法分别产生模型 λ_A、λ_B 和 λ_C，以及保存相应的转移次数、矢量数和状态数目，根据(1-75)和(1-76)式的思想，只要将相应的参数分子分母分别相加，即可十分灵活地得到反映 D_A+D_B、D_A+D_C 或 D_B+D_C、$D_A+D_B+D_C$ 的特性的各个模型参数来。这样，使 HMM 参数估计的过程具有良好的自适应性和很强的自

学习能力：只要增加新的训练数据，产生的最后模型就能反映这新增数据的信息。

处理说话者的影响，一直是语音识别研究中所关注的问题，也称之为说话者自适应问题。不少学者在这方面都做了很有成效的工作，例如，Shikano 等提出了以 VQ 技术为基础的系统对说话者自适应的方法[213]，Sugawara 等提出了离散 HMM 系统自适应方法[216]，后来，他们又提出了一个分两阶段自适应的方法[175]，Martin 等人也提出了一个针对高斯型概密 HMM（下一章将介绍这种形式的 HMM）系统的自适应方法[160]。Nakamura 等人提出了一种基于 Fuzzy 理论和 VQ 的自适应方法，这种新方法不依赖于识别系统的具体算法，他们将其用于 HMM 系统和神经网络（NN）系统，均获得了成功[171]。Lee 等人针对连续 HMM 的说话者自适应问题，提出了一个 Bayesian 学习算法[130,131]。除此之外，还有很多有代表性的工作[76,89]。

1.4.3 改进经典训练算法

HMM 训练问题，是 HMM 在语音处理各个领域中成功应用的关键问题。因此，这个问题得到很多人的关注就是理所当然的了。经典算法 Baum-Welch 算法，实际上是 HMM 的最大似然（Maximum Likelihood，ML）参数估计方法，即给定训练序列 O，使 $P(O/\lambda)$最大时求出 λ。而最大似然估值并不是唯一的，也不是在所有情况下都适用的准则。正因为这样，人们提出了很多改进的途径。其中最有代表性的就是基于最大互信息（Maximum Mutual Information，MMI）准则的估计方法[23,44,65,70,163]。因为当事先假定的模型不正确时，MMI 估值器优于 ML 估值器[170]。

对训练序列 O 和模型 λ，互信息定义为

$$I(\lambda,O)=\lg\frac{P(O,\lambda)}{P(O)P(\lambda)}=\lg\frac{P(O/\lambda)}{P(O)}$$

$$=\lg P(O/\lambda)-\lg P(O)$$

$$= \lg P(O/\lambda) - \lg \sum_{\lambda'} P(O/\lambda')P(\lambda') \tag{1-77}$$

所谓最大互信息准则就是使 $I(\lambda,O)$ 最大，从而求取 λ。

但目前对 MMI 估值还没有找到类似于 ML 估值中的前向-后向算法那样有效的方法，因此，使 $I(\lambda,O)$ 最大一般采用经典的最大梯度法。尽管如此，人们还是做了不少工作。例如，MMI 训练的一种有效实现方法[52]；加快梯度法收敛速度的措施[177]等等。

另一种改进的算法是将训练和最后对模型的测试联系起来，利用识别结果来修正训练的模型，比如纠错训练法[25]，它的基本思想为：对训练数据集 D，先用经典的 Baum-Welch 算法训练一组模型，再用模型对训练集 D 做测试识别，若其中词 w 被误识成 ω，则调整模型参数，使产生 w 的概率增大，产生 ω 的概率减小，从而有可能消去这一错误。一直持续这个过程，使最后得到的一组模型最好地表达了训练数据集。显然这种方法的关键之处是怎样调整模型参数，下面简要介绍一种可行的方法[254]：

以模型参数 a_{ij} 为例，根据(1-69)式重估公式可以写为

$$\bar{a}_{ij} = \frac{\text{trans-counts}(i,j)}{\text{state-counts}(i)} \tag{1-78}$$

这样，设训练数据 $O^{(l)}$，$l=1,\cdots,L$，由经典方法分别训练成 L 个模型 λ_l，$l=1,\cdots,L$，而且，测试识别时 $O^{(p)}$ 误识成了 $O^{(q)}$，即有

$$P(O^{(p)}/\lambda_p) < P(O^{(p)}/\lambda_q) \tag{1-79}$$

因此，应调整模型参数 λ_p，提高 $P(O^{(p)}/\lambda_p)$，使(1-79)式不再成立，从而减少这一错误。调整过程为：用数据 $O^{(p)}$ 对 λ_p 用重估公式做一次迭代，得到相应的转移次数和状态数目，即 trans-counts$(i,j)^{(p)}$、state-counts$(i)^{(p)}$。再由数据 $O^{(p)}$ 对 λ_q 也做一次迭代，得到 trans-counts$(i,j)^{(q)}$ 和 state-counts$(i)^{(q)}$，那么，调整后的模型 λ_p 的参数为

$$\bar{a}_{ij} = \frac{\text{trans-counts}(i,j) + \delta_1 t_d}{\text{state-counts}(i) + \delta_2 s_d} \tag{1-80}$$

其中 $t_d = \text{trans-counts}(i,j)^{(p)} - \text{trans-counts}(i,j)^{(q)}$

$$s_d = \text{state-counts}(i)^{(\rho)} - \text{state-counts}(i)^{(\varphi)}$$

δ_1 和 δ_2 为修正因子，在[0,1]范围内取值。实验表明，采用纠错训练之后，比单纯用 Baum-Welch 算法训练对识别率有明显的提高。进一步的理论分析，可以推出上述方法的一般形式[1]。它是通过选取目标函数为

$$P(\lambda/O) = \frac{P(O/\lambda)P(\lambda)}{\sum_{l=1}^{L} P(O/\lambda_l)P(\lambda_l)} \tag{1-81}$$

其中，λ_l，$l=1$，…，L，为系统要建立的模型。用拉格朗日方法，使 $P(\lambda/O)$最大，在前向-后向算法基础上导出 λ 的参数估计公式。由于一般认为 $P(\lambda)$只与语言学模型有关，与要估计的模型参数无关，因此，比较(1-77)和(1-81)式，发现它们很相似，只不过，(1-81)式对 $P(O)$的计算做了特别的处理，从这个意义上讲，这个方法是以 MMI 准则训练 HMM 的一个特殊的实现方式。

HMM 的纠错训练和类似的使错误最小的训练，是一个广为关注的课题[51,71,152]。

应当指出，虽然对经典的 Baum-Welch 算法的改进工作还有很多，例如，分段 k 平均训练算法[114]（最佳化的目标函数为 $\max P(O,S/\lambda)$以取代经典方法中的如(1-18)式所示的 $\sum P(O,S/\lambda)$，即 $P(O/\lambda)$）；类似于神经网络训练而提出的竞争训练法[234]，为了提高 HMM 对模式的辨识能力而用训练数据再产生一个从属模型的方法[99]；离散 HMM 训练时将 VQ 一起优化的方法[120,178]；不仅估计 HMM 参数，而且也优化 HMM 结构的方法[48,231]；分解 HMM 成几个更准确模型，再用线性插值的方法[19]；用统计的矩阵量化(MQ)取代一般 VQ 来改进离散 HMM 的方法[176]；用最可能的状态序列取代经典 ML 方法中所有状态序列的方法[162]等等。但这方面的工作还没有完结，寻求更有效的 HMM 参数估计方法仍是一个活跃的研究方向。

第二章　各具特色的 HMM

第一章讨论的 HMM，由于其观察值是 M 个离散可数的观察值中的一个，因而称之为离散 HMM，某个状态 j 对应的观察值的统计特性是由一组概率 $b_{jk}, k=1,\cdots,M$ 来描述的。本章介绍在此基础上提出的各式各样有代表性的 HMM。由于离散 HMM 由图 1.2 所示两部分组成，因此，这里讨论的各具特色的 HMM，大多可以认为是从不同角度出发为了更好解决语音处理中的问题而对这两部分进行不同的修正而产生的。首先，第一节讨论修正图 1.2 第二部分由 B 描述的随机过程而得到的连续和半连续 HMM。所谓连续 HMM，指观察值为一个连续随机变量 X，因此，某个状态 j 对应的观察值统计特性由一个观察值概率密度函数 $b_j(X)$ 表示，这里对后续章节中引用较多的连续 HMM 的一种有代表性的形式——线性预测 HMM 也进行了较为细致的讨论（一般认为，连续 HMM 比离散 HMM 性能更好[75]）。而半连续 HMM，可以认为是 HMM 的一种一般形式，连续 HMM 和离散 HMM 都是其特例。而且，半连续 HMM 在一定程度上兼有二者的长处。接着，在第二节中给出了对图 1.2 第一部分由 π、A 描述的 Markov 链加以修正而得到的三种主要的 HMM，即利用 Gibbs 分布取代 Markov 链的 HMM；在 Markov 链中加入状态驻留时间参数的 HMM 以及二阶 HMM。最后，第三节对其它有代表性的 HMM 进行了简要综述。

§2.1 连续和半连续 HMM

2.1.1 连续 HMM

在离散 HMM 基础之上给出的连续 HMM，仍可记为 $\lambda=(\pi, A, B)$。其中 B 不再是一个矩阵，而是一组观察值概密函数，即 $B=\{b_j(X), j=1,\cdots,N\}$。对这种形式的 HMM，前向-后向算法公式(1-20)～(1-27)式和 Viterbi 算法公式(1-29)～(1-35)式仍可使用，但用来估计模型参数的 Baum-Welch 算法中的重估公式，虽然在估计 π、A 参数时仍适用，但估计描述 $b_j(X)$的参数，必须对 $b_j(X)$加以一定的限制才成立，Baum 等人给出重估公式时，要求 $b_j(X)$为一个对数凹函数[33]。后来，Liporace 引用 Fan 提出的表达定理，重新定义了辅助函数，将 $b_j(X)$的限制拓宽为椭圆对称多变量分布函数，即有[149]

$$b_j(X) = |R_j|^{-1/2} h_j(g_j(X)) \tag{2-1}$$

其中

$$g_j(X) = (X-\eta_j)^T R_j^{-1}(X-\eta_j) \tag{2-2}$$

为一个正定二次型。Juang 等人进一步放宽了 $b_j(X)$的限制，认为重估公式的成立基础可以扩展为：$b_j(X)$是对数凹函数、椭圆对称函数以及它们的线性组合[108,110]。由于这种线性组合可以逼近很多有实际意义的函数形式，从而为连续 HMM 在语音处理中的应用打下了坚实的基础。

当然，不同形式的 $b_j(X)$是由不同的参数来描述的，而估计这种参数的重估公式也是不一样的。这里以一种广泛应用的概密函数——高斯型 $b_j(X)$为例[108,110]：

$$\begin{aligned} b_j(X) &= \sum_{k=1}^{K} c_{jk} b_{jk}(X) \\ &= \sum_{k=1}^{K} c_{jk} N(X, \mu_{jk}, \Sigma_{jk}), \quad 1 \leqslant j \leqslant N \end{aligned} \tag{2-3}$$

其中，$N(X,\mu_{jk},\Sigma_{jk})$为多维高斯概密函数，$\mu_{jk}$为其均值矢量，$\Sigma_{jk}$为方差矩阵，$K$ 为组成 $b_j(X)$的混合概密个数，c_{jk}为组合系数，且

$$\sum_{k=1}^{K} c_{jk} = 1 \tag{2-4}$$

设观察值训练序列为 $O=O_1,O_2,\cdots,O_T$，则可推导出重估公式为

$$\tilde{c}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)}{\sum_{t=1}^{T}\xi_t(j)} \tag{2-5}$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)O_t}{\sum_{t=1}^{T}\gamma_t(j,k)} \tag{2-6}$$

$$\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)(O_t-\bar{\mu}_{jk})(O_t-\bar{\mu}_{jk})^T}{\sum_{t=1}^{T}\gamma_t(j,k)} \tag{2-7}$$

其中
$$\gamma_t(j,k) = \sum_{i=1}^{N}\alpha_{t-1}(i)a_{ij}c_{jk}b_{jk}(O_t)\beta_t(j)/P(O/\lambda) \tag{2-8}$$

$\xi_t(j)$如(1-39)式定义。具体推导过程见附录。

另一种广泛应用的概密函数 $b_j(X)$为线性预测型或称高斯自回归型，相应的连续 HMM 称为线性预测 HMM，由于后续章节中多次引用到这种 HMM，本节第二部分将专门讨论它。

对高斯型 $b_j(X)$，如(2-3)式定义，在最简单的情形 $K=1$ 时，可以写为

$$b_j(X) = k_1\exp[-D_1(X;\mu_j)] \tag{2-9}$$

其中
$$k_1 = \frac{1}{(2\pi|\Sigma_j|)^{1/2}} \tag{2-10}$$

$$D_1(X;\mu_j) = (X-\mu_j)\Sigma_j^{-1}(X-\mu_j)^T \tag{2-11}$$

这是，μ_j 为均值矢量，Σ_j 为方差矩阵，$D_1(X;\mu_j)$事实上是矢量 X 和 μ_j 之间的 Mahalanobis 距离[220]。后面将会看到，线性预测型 $b_j(X)$亦能表示成类似(2-9)式的形式，因此，可以给出连续 HMM 的一种一般性概密函数形式[251]

$$b_j(X) = k_0\exp[-D(X;B_j)] \tag{2-12}$$

其中，k_0 为某个常数，目的是使 $b_j(X)$为一概密函数，即

$$\int_X b_j(X)\mathrm{d}X = 1 \tag{2-13}$$

对语音处理应用而言，k_0 的具体数值多少没什么意义，只需知道它是一个常数即可。B_j 为描述 $b_j(X)$的参数，它是观察值 X 相对应的语音某个特征矢量，那么，$D(X;B_j)$就是这种特征矢量的距离量度。下面以语音倒谱系数特征矢量和相应的 Euclidean 距离为例，给出估计这种 $b_j(X)$参数的重估公式[247]：

倒谱系数是描述语音的特征之一，可由著名的 LPC 系数导出[198]

$$c(1) = a(1) \tag{2-14}$$

$$c(i) = a(i) + \sum_{k=1}^{i-1}(1 - k/i)a(k)c(i-k),$$

$$1 \leqslant i \leqslant k \tag{2-15}$$

其中，$a(i), i=1,\cdots,p$，为 LPC 系数，$c(i), i=1,\cdots,p$，为倒谱系数，p 为 LPC 分析阶数，此时，

$$b_j(X) = k_0\exp[-D(X;C_j)] \tag{2-16}$$

其中，$C_j=(C_{j1},\cdots,C_{jp})^T$ 为 p 阶倒谱系数，是描述 $b_j(X)$的参数，设 $C_x=(C_{x1},\cdots,C_{xp})$为语音帧 X 的倒谱系数，那么

$$D(X;C_j) = (C_x - C_j)^T(C_x - C_j) = \sum_{m=1}^{p}(C_{xm} - C_{jm})^2 \tag{2-17}$$

与推导离散 HMM[144]和连续 HMM[108,110]的重估公式一样，可以得到参数 C_j 的重估公式

$$\overline{C}_{jm}=\frac{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)C_{o_tm}}{\sum_{t=1}^{T}\alpha_t(j)\beta_t(j)},\quad 1\leqslant j\leqslant N,1\leqslant m\leqslant p \tag{2-18}$$

其中，C_{o_tm}为语音第 t 帧 O_t 的倒谱系数 C_{o_t}中的第 m 个值。当训练序列为 L 个时，(2-18)式易推广，而且，具体实现时，其下溢问题也能用 1.3.3 中的比例因子方法加以解决。这样的 HMM 在孤立词识别中取得了成功。显然，(2-12)式的概密亦即推广到多个概密函数的混合，即

$$b_j(X)=\sum_{k=1}^{K}c_{jk}b_{jk}(X) \tag{2-19}$$

其中，$b_{jk}(X)$具有(2-12)式的形式。

2.1.2 线性预测 HMM

线性预测 HMM(Linear Predictive HMM)是具有线性预测型观察值概密函数 $b_j(X)$的一种连续 HMM，最早由 Poritz 提出，并用于说话人辨识[187]，后来 Juang 和 Rabiner 对其进行了深入研究并给出了由多个概密函数混合而成的 $b_j(X)$，在孤立词识别中得到了成功的应用[109,113]。作者在博士论文中对这种 HMM 在语音识别、压缩和增强等多方面的应用进行了系统的研究[237]。

线性预测 HMM 是以语音处理中著名的 LPC 分析理论为基础的。而线性预测的概念也广泛用于信息与控制中的估值和系统辨识问题，因而人们也常常用系统辨识的术语描述 LPC 方法。由于在时不变的条件下，线性预测表示的是系统辨识中常用的自回归(AR)模型[227]，因此，这种 HMM 也称为高斯自回归 HMM (Gaussian Autoregressive HMM)。

LPC 的基本概念是，一个语音抽样能够用过去若干个语音抽样的线性组合来逼近，通过使实际语音抽样和线性预测值之间差值平方和(在一个短时区间内)达到最小值，能够唯一决定一组预测器系数(也就是线性组合中所用的加权系数)，这就是众所周知

的 LPC 系数[198]。求取 LPC 系数，目前已有几十种算法，其中，以自相关法、协方差法和格型法最为知名。本书中所有实验中求取 LPC 系数时都是采用自相关法，此时 LPC 系数 $a_i, i=1,\cdots,p$，是下列方程的解：

$$\sum_{k=1}^{p} a_k R_n(|i-k|) = R_n(i), \qquad i = 1,\cdots,p \qquad (2\text{-}20)$$

其中，$R_n(i)$为语音第 n 帧的自相关函数。(2-20)式亦有多种有效解法，一般采用的是 Robinson 递归法[3,159]。由此可见，求取某帧语音的 LPC 系数，等价于求取其自相关函数，因为一旦获知自相关函数，由(2-20)式，LPC 系数也可以求出了。

在 LPC 分析的一个短时区间内，一帧语音，记为 $S^T=(s_1,\cdots,s_K)$，K 为帧长，作为如图 2.1 所示的一个高斯 AR 模型的输出，有下列关系成立[109]：

$$x_i = -\sum_{j=1} a_j x_{i-j} + e_i \qquad (2\text{-}21)$$

$$s_i = \sigma x_i \qquad (2\text{-}22)$$

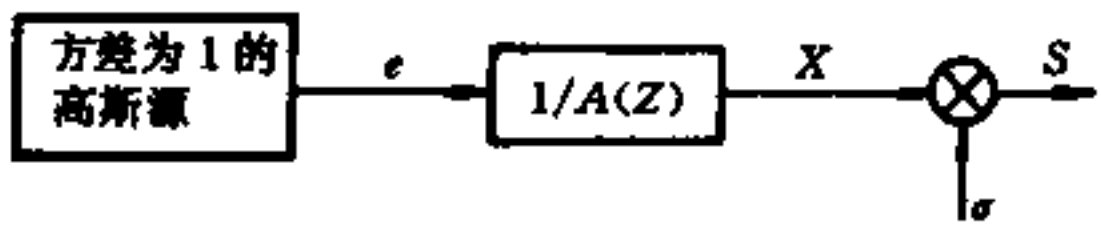


图 2.1　高斯 AR 模型示意图

图 2.1 中，$A(Z)=1+a_1Z^{-1}+\cdots+a_pZ^{-p}$，其中，$a_1,\cdots,a_p$ 为 LPC 系数。

当把 S 作为一个随机过程(矢量)的一个实现，在已知自相关矩阵 C_i 时，有下面的高斯概密函数：

$$f(S/C_i) \doteq (2\pi)^{-K/2}|C_i|^{-1/2}\exp\left[-\frac{1}{2}S^TC_i^{-1}S\right] \qquad (2\text{-}23)$$

经过推导[109]，有

$$|C_i| = (\sigma^2)^K \qquad (2\text{-}24)$$

$$S^TC_i^{-1}S = \delta(\sigma^{-1}S;a) \qquad (2\text{-}25)$$

其中，$a^T=(a_0,a_1,\cdots,a_p)$，$a_0=1$，为 LPC 系数，记 $R_a(i)$为 a^T 的自相关函数，$R_s(i)$为 S^T 的自相关函数，则

$$\delta(\sigma^{-1}S;a)=[R_a(0)R_s(0)+2\sum_{i=1}^{p}R_a(i)R_s(i)]/\sigma^2 \quad (2\text{-}26)$$

其中，σ^2 就是 LPC 分析时的预测残差(平均)能量。因此，(2-23)式可写为

$$f_s(S/a,\sigma^2)=(2\pi)^{-K/2}(\sigma^2)^{-K/2}\exp[-\frac{1}{2}\delta(\sigma^{-1}S;a)] \quad (2\text{-}27)$$

对于一帧语音 S^T 的归一化语音 $X^T=S^T/\sigma$，根据概率论公式，有

$$f(X/a)=\sigma^K f_s(\sigma X/a,\sigma^2)$$

$$=(2\pi)^{-K/2}\exp[-\frac{1}{2}\delta(X;a)] \quad (2\text{-}28)$$

其中 $$\delta(X;a)=R_x(0)R_a(0)+2\sum_{i=1}^{p}R_x(i)R_a(i) \quad (2\text{-}29)$$

$R_x(i)$为 X^T 的自相关函数，即归一化语音帧的自相关函数。

当给出一帧语音(归一化)$X^{(0)}$时，在 LPC 理论中用自相关法确定 LPC 系数 $a^{(0)}$，等价于使 $f(X^{(0)}/a)$最大时确定的 a 值，或者说，等价于使 $\delta(X^{(0)};a)$最小时的 a 值。如果记 $a^{(0)}$为表示 $X^{(0)}$的 LPC 系数，a 表示 X 的 LPC 系数，那么

$$f(X^{(0)}/a)=(2\pi)^{-K/2}\exp\left[-\frac{1}{2}\delta(X^{(0)};a)\right] \quad (2\text{-}30)$$

其中，$\delta(X^{(0)};a)$就是度量 LPC 系数相似程度的著名的 Itakura 距离[103,194,198]。当然，对 LPC 系数而言，也还有其它可供选择的距离量度[245]。(2-30)式揭示了距离量度和概密函数的联系。

有了上述讨论，可以给出线性预测 HMM 的概密函数为

$$b_j(X)=(2\pi)^{-K/2}\exp[-\frac{1}{2}\delta(X;a_j)] \quad (2\text{-}31)$$

其中，K 为归一化语音帧的帧长(点数)，a_j 为描述 $b_j(X)$的参数，

是一组 LPC 系数，记 $R_a(i)$和 $R_x(i)$分别为 a_j^T 和 X^T 的自相关函数，则有

$$\delta(X;a_j) = R_a(0)R_x(0) + 2\sum_{i=1}^{p} R_a(i)R_x(i) \qquad (2\text{-}32)$$

p 为 LPC 分析阶数。现在的关键问题就是在给定训练序列 $O=O_1,\cdots,O_T$ 时，由 Baum-Welch 算法中的重估公式思想推导出估计描述 $b_j(X)$的参数——LPC 系数 a_j 的重估公式，或者说估计出 a_j 相对应的等价的某个自相关函数 R_j 的重估公式。

经过推导，见附录有关部分，自相关函数 R_j 的重估公式为

$$\overline{R}_j(i) = \sum_{t=1}^{T} P(O,q_t = \theta_j/\lambda)R_t(i), i = 0,1,\cdots,p \qquad (2\text{-}33)$$

对于实际中的 L 个训练序列 $O^{(l)}$，$l=1,\cdots,L$，$\overline{R}_j(i)$的重估公式为

$$\overline{R}_j(i) = \sum_{l=1}^{L}\sum_{t=1}^{T_l} P(O^{(l)},q_t = \theta_j/\lambda)R_t^{(l)}(i),$$
$$i = 0,1,\cdots,p, j = 1,\cdots,N \qquad (2\text{-}34)$$

其中，$R_t^{(l)}(i)$是 $O^{(l)}$中第 t 个值 $O_t^{(l)}$ 的自相关函数，T_l 为 $O^{(l)}$的长度。

在实际编程时，依照已有资料建议[113]，$\overline{R}_j(i)$采用增加一个比例系数的公式

$$\overline{R}_j(i) = \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} P(O^{(l)},q_t = \theta_j/\lambda)R_t^{(l)}(i)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l} P(O^{(l)},q_t = \theta_j/\lambda)}$$
$$= \frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \alpha_t^{(l)}(j)\beta_t^{(l)}(j)R_t^{(l)}(i)/P(O^{(l)}/\lambda)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l} \alpha_t^{(l)}(j)\beta_t^{(l)}(j)/P(O^{(l)}/\lambda)} \qquad (2\text{-}35)$$

这样，为了防止计算上的下溢，最后的实现公式为

$$\overline{R}_j(i)=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)R_t^{(l)}(i)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)},$$

$$1\leqslant j\leqslant N, i=0,1,\cdots,p \tag{2-36}$$

其中，$\alpha_t^*(j)$和$\beta_t^*(j)$分别如(1-55)和(1-59)式定义。

当然，线性预测 HMM 的观察值概率函数 $b_j(X)$也可推广为[113]

$$b_j(X)=\sum_{k=1}^{K}c_{jk}b_{jk}(X) \tag{2-37}$$

其中 $b_{jk}(X)$如(2-31)式所定义的形式，c_{jk}为组合系数，满足(2-4)式的约束。这样的 HMM 性能更好。

2.1.3 半连续 HMM

半连续 HMM(Semi-continuous HMM)是在对离散 HMM 和连续 HMM 思想进行综合后提出来的，最早由 Huang 等人提出并进行了系统研究[93,95,96,97,98]，稍后 Bellgarda 和 Nahamoo 也独立地提出了这一思想[35]。

离散 HMM 处理的观察值必须在 M 个离散值组成的集合之内。因此，对语音信号段分帧产生的特征参数序列，比如 LPC 系数必须经过矢量量化(VQ)之后，才能用于离散 HMM 的训练和识别[196,197]，此时，VQ 的码本(codebook)中 M 个码字(codeword)，就是 HMM 相应的 M 个观察值集合。这样的处理过程就产生了离散 HMM 的两个缺陷：首先，将语音数据，确切地说，是各帧的特征参数，经过 VQ，造成了一些信息丢失。此外，VQ 的码本训练和离散 HMM 训练是两个分开的最优化过程，而不是一起进行优化训练。

另一方面，连续 HMM 虽然能直接处理语音数据，但为了取得语音识别的良好性能，必须要较多的概率函数进行混合，以得到

某个状态对应的观察值概密函数，对如(2-3)式所示的高斯型概密来说，K 值就要比较大。这样，不仅模型复杂，运算量大，而且，需要更多的训练数据才能得到可靠的模型参数。

半连续 HMM 就是为了克服离散 HMM 和连续 HMM 的上述不足之处而提出来的，其基本思想为：

离散 HMM 使用 VQ 产生 M 个码字组成的码本，这 M 个码字实际上就是将训练矢量空间划分为 M 个部分，这种划分是丢失语音特征信息的原因，因此，用 M 个高斯型概密函数取代 M 个码字，这样，训练矢量空间的划分就不会损失信息，而且，VQ 的训练和 HMM 的训练可在一个优化过程中完成。此时，状态 θ_i 对应的观察值概密函数为

$$b_i(\boldsymbol{X}) = \sum_{j=1}^{M} f(\boldsymbol{X}/v_j) b_i(j) \tag{2-38}$$

其中，$f(\boldsymbol{X}/v_j)$就是第 j 个码字 v_j 对应的高斯概密函数；$b_i(j)$就是 $P(v_j/q_t=\theta_i)$。实际上，由于码本中码字较多，即 M 值较大，因此，为了减少计算量，(2-38)式一般简化为

$$b_i(\boldsymbol{X}) = \sum_{v_j \in \eta(\boldsymbol{X})} f(\boldsymbol{X}/v_j) b_i(j) \tag{2-39}$$

其中，$\eta(\boldsymbol{X})$表示那些使 $f(\boldsymbol{X}/v_j)$足够大的码字的集合，这样 $\eta(\boldsymbol{X})$就比 M 小了很多。

从半连续 HMM 的观察值概密表达式(2-39)式可知，它事实上是离散 HMM 和连续 HMM 的一种一般形式。当 $\eta(\boldsymbol{X})$只选一个使 $f(\boldsymbol{X}/v_j)$最大的码字，即与 $\boldsymbol{X}$ 最接近的码字，此时，半连续 HMM 就变为离散 HMM。另一方面，半连续 HMM 也是连续 HMM，只不过，N 个状态的概密函数共同拥有 M 个概密函数，而且 $b_i(j)$就是混合若干概密函数时的加权系数。

剩下的关键问题就是在一次优化过程中训练 VQ 和 HMM，这可由与 Baum-Welch 算法类似的重估公式思想来解决：设训练序列为 $O=\boldsymbol{O}_1,\boldsymbol{O}_2,\cdots,\boldsymbol{O}_T$，可以推导出：

$$\bar{b}_i(j)=\frac{\sum_{t=1}^{T} r_t(i,j)}{\sum_{t=1}^{T}\xi_t(i)},\quad 1\leqslant i\leqslant N,1\leqslant j\leqslant M \tag{2-40}$$

对高斯型概密函数 $f(X/v_j)$的均值矢量 μ_j 和方差矩阵 Σ_j 有

$$\bar{\mu}_j=\frac{\sum_{t=1}^{T} r_t(j)O_t}{\sum_{t=1}^{T} r_t(j)},\quad 1\leqslant j\leqslant M \tag{2-41}$$

$$\bar{\Sigma}_j=\frac{\sum_{t=1}^{T} r_t(j)(O_t-\bar{\mu}_j)(O_t-\bar{\mu}_j)^T}{\sum_{t=1}^{T} r_t(j)},\quad 1\leqslant j\leqslant M \tag{2-42}$$

其中

$$r_t(i,j)=\begin{cases}\pi_i b_i(j) f(O_1/v_j)\beta_1(i)/P(O/\lambda), & t=1\\ \sum_k \chi_{t-1}(k,i,j), & 1<t\leqslant T\end{cases} \tag{2-43}$$

而

$$\chi_t(i,j,k)=\alpha_t(i)a_{ij}b_j(k)f(O_{t+1}/v_k)\beta_{t+1}(j)/P(O/\lambda) \tag{2-44}$$

$$r_t(j)=\sum_i r_t(i,j),\quad 1\leqslant t\leqslant T \tag{2-45}$$

$\xi_t(i)$由(1-39)式定义。

具体的推导过程以及详尽的分析，可参见文献[92,94]。比较(2-5)和(2-40)式、(2-6)和(2-41)式、(2-7)和(2-42)式，发现它们是相似的。这对于理解半连续和连续 HMM 的联系是有帮助的。类似于离散 HMM 和连续 HMM，半连续 HMM 的重估公式也能推广到 L 个训练序列的情形。在语音处理中的应用表明，半连续 HMM 比离散和连续 HMM 有更好的性能，例如音素分类实验[93]。从已有的报道看，半连续 HMM 是经典离散 HMM 的一种较为成功的修正形式，在语音识别中也有较为广泛的应用[166]。

§2.2 对 Markov 链修正后的 HMM

2.2.1 利用 Gibbs 分布取代 Markov 链的 HMM

Gibbs 分布在图象纹理分析和图象恢复中有广泛的应用[61,80]。这里，它被用来描述 HMM 的状态序列，以取代由 π、A 表示的 Markov 链，从而得到 HMM 的一种较为一般的形式。

在 HMM 中，状态序列 $S=q_1,q_2,\cdots,q_T$ 产生的概率，如(1-17)式所示，由 π 和 A 来表征。而用 Gibbs 分布，产生 S 的概率为

$$P(S/\lambda)=\exp[-U(S)]/Z \tag{2-46}$$

其中，$U(S)$为能量函数，而 Z 为一个归一化的项，即

$$Z=\sum_{\text{所有}S}\exp[-U(S)] \tag{2-47}$$

因此，Gibbs 分布由其能量函数 $U(S)$确定。对于一维一阶邻域的 Markov 随机场，能量函数 $U(S)$的一般形式为

$$U(S)=\sum_{t=1}^{T}h(q_t)+\sum_{t=2}^{T}g(q_{t-1},q_t) \tag{2-48}$$

其中，h 和 g 为任意实函数，这样，如果用 Gibbs 分布表示 Markov 链，有

$$U(S)=\sum_{j=1}^{N}\hat{\pi}_jJ_j(q_1)+\sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N}\hat{a}_{ij}J_{ij}(q_{t-1},q_t) \tag{2-49}$$

其中

$$J_j(q_1)=\begin{cases}1, & \text{如果 } q_1=\theta_j\\ 0, & \text{其它}\end{cases} \tag{2-50}$$

$$J_{ij}(q_{t-1},q_t)=\begin{cases}1, & \text{如果 } q_{t-1}=\theta_i,q_t=\theta_j\\ 0, & \text{其它}\end{cases} \tag{2-51}$$

$$\sum_{j=1}^{N}e^{-\hat{\pi}_j}=1 \tag{2-52}$$

$$\sum_{j=1}^{N} e^{-\hat{a}_{ij}} = 1, \qquad i = 1, \cdots, N \tag{2-53}$$

此时，参数 $\hat{\pi}_j$ 和 $\hat{a}_{ij}$就描述了能量函数，或者说描述了 Gibbs 分布。显然，与经典 Markov 链参数 π、A 的关系为

$$\hat{\pi}_j = -\lg \pi_j, \qquad j = 1, \cdots, N \tag{2-54}$$

$$\hat{a}_{ij} = -\lg a_{ij}, \qquad i, j = 1, \cdots, N \tag{2-55}$$

因此，如果用(2-49)式的能量函数 $U(s)$表示的 Gibbs 分布来描述 HMM 的状态序列，与由 π、A 表示的 Markov 链来描述状态序列，是完全等价的。但是，由于 $U(s)$的一般形式如(2-48)式所示，因此，描述状态序列的 Gibbs 分布还有很多可能的选择。例如，可选取在图象纹理分析时使用的能量函数[60]，在一维最近邻域情形，这种能量函数可以简化为：

$$U(s) = \sum_{j=1}^{N} \sum_{t=1}^{T} \hat{a}_j J_j(q_t) + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=2}^{T} \hat{\beta}_{|j-i|} J_{ij}(q_{t-1}, q_t) \tag{2-56}$$

其中，$\hat{a}_j$ 称为外部场参数，$\hat{\beta}_{|j-i|}$称为结合强度参数，它们共同描述了能量函数 $U(s)$，或其相应的 Gibbs 分布。

那么，利用 Gibbs 分布的 HMM 的参数集就是$(\hat{\alpha}, \hat{\beta}, B)$，其中，$\hat{\alpha}$ 和 $\hat{\beta}$ 是描述 Gibbs 分布的参数。Zhao 等人选取线性预测型观察值概密函数，如(2-31)式所示，对这种形式的修正 HMM 进行了细致的研究，并在孤立词识别中取得了成功[235,236]。他们的主要工作为：利用 Baum-Welch 算法的思想，导出了估计 $\hat{\alpha}$ 和 $\hat{\beta}$ 参数的重估公式。此外，类似前向-后向算法，定义了前向函数和后向函数，有效地解决了概率 $P(O/\lambda)$的计算问题，因为

$$P(O/\lambda) = \sum_{\text{所有}S} P(O/S, \lambda) P(S/\lambda) \tag{2-57}$$

在 $P(S/\lambda)$如(2-46)式所示的情况下，直接计算 $P(O/\lambda)$将因计算量太大而不可能做到。

2.2.2 在 Markov 链中考虑状态驻留时间的 HMM

经典 HMM 中的 Markov 链由 π、A 表征，因此，在状态 θ_i 上相继产生 d 个观察值的概率为

$$p_i(d) = (a_{ii})^d(1 - a_{ii}) \tag{2-58}$$

这个概率值 $p_i(d)$描述了状态 θ_i 的驻留时间(state duration)。显然，这是一个指数分布，且其最大值出现在 $d=0$ 处。这与语音的物理事实不相符合，因为在 HMM 应用于语音处理中时，状态一般总与一定的语音单位相对应，而这些语音单位都具有相对稳定的分布。针对经典 HMM 的这个缺陷，自 80 年代中期以来，很多研究人员提出了各自的改进措施，基本思想都是在 Markov 链中考虑状态驻留时间的非指数分布 $p_i(d)$(指数分布的 $p_i(d)$就是经典 HMM 中 Markov 链所具有的特性)。或者说，对描述 Markov 链的参数集 π、A 进行修正，增加一项描述状态驻留时间的概率值 $p_i(d)$。

一种最直接的方法就是所谓非参数方法[113,192,198]，即在 Markov 链参数中，令 $a_{ii}=0$，同时，增加状态驻留时间概率分布 $p_i(d), d=1,\cdots,D$，其 D 为所有状态可能停留的最长时间值，那么，这种 HMM 产生的输出观察值序列的过程为：由 π_i 选择初始状态 q_1，根据 $p_{q_1}(d)$确定状态驻留时间 d_1，产生 d_1 个观察值 O_1，$O_2,\cdots,O_{d_1}$，其概率为 $\prod_{t=1}^{d_1} b_{q_1}(O_t)$，再根据 $a_{q_1q_2}$，选择下一个状态 q_2，重复这个过程，直到整个观察值序列 $O=O_1,O_2,\cdots,O_T$ 产生完毕。

为了计算 $P(O/\lambda)$，类似(1-19)式，定义前向变量为

$$\alpha_t(i) = P(O_1,O_2,\cdots,O_t,t \text{ 时结束于状态 } \theta_i/\lambda) \tag{2-59}$$

那么，$\alpha_t(j)$的递归公式变为

$$\alpha_t(j) = \sum_{i=1}^{N}\sum_{d=1}^{D}\alpha_{t-d}(i)a_{ij}p_j(d)\prod_{s=t-d+1}^{t} b_j(O_s) \tag{2-60}$$

因此，与经典 HMM 一样，有

$$P(O/\lambda) = \sum_{i=1}^{N} a_T(i) \tag{2-61}$$

为了训练这种修正 HMM，导出估计其参数的重估公式，还必须定义另外三个前向、后向变量：

$$\hat{\alpha}_t(i) = P(O_1, O_2, \cdots, O_t, \text{状态 } \theta_i \text{ 始于 } t+1/\lambda) \tag{2-62}$$

$$\beta_t(i) = P(O_{t+1}, \cdots, O_T/\theta_i \text{ 止于 } t, \lambda) \tag{2-63}$$

$$\hat{\beta}_t(i) = P(O_{t+1}, \cdots, O_T/\theta_i \text{ 始于 } t+1, \lambda) \tag{2-64}$$

显然，α 和 $\hat{\alpha}$、β 和 $\hat{\beta}$ 的关系为

$$\hat{\alpha}_t(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} \tag{2-65}$$

$$\alpha_t(i) = \sum_{d=1}^{D} \hat{\alpha}_{t-d}(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(O_s) \tag{2-66}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \hat{\beta}_t(j) \tag{2-67}$$

$$\hat{\beta}_t(i) = \sum_{d=1}^{D} \beta_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(O_s) \tag{2-68}$$

由此推导出的重估公式为

$$\overline{\pi_i} = \frac{\pi_i \beta_0(i)}{P(O/\lambda)} \tag{2-69}$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t(j)} \tag{2-70}$$

$$\overline{b_{jk}} = \frac{\sum_{\substack{t=1 \\ \text{且} O_t = V_k}}^{T} \left[\sum_{\tau<t} \hat{\alpha}_\tau(j) \beta_\tau(j) - \sum_{\tau<t} \alpha_\tau(j) \beta_\tau(j) \right]}{\sum_{k=1}^{M} \sum_{\substack{t=1 \\ \text{且} O_t = V_k}}^{T} \left[\sum_{\tau<t} \hat{\alpha}_\tau(j) \beta_\tau(j) - \sum_{\tau<t} \alpha_\tau(j) \beta_\tau(j) \right]} \tag{2-71}$$

$$\overline{p_i}(d)=\frac{\sum_{t=1}^{T}\hat{\alpha}_t(i)p_i(d)\beta_{t+d}(i)\prod_{s=t+1}^{t+d}b_i(O_s)}{\sum_{d=1}^{D}\sum_{t=1}^{T}\hat{\alpha}_t(i)p_i(d)\beta_{t+d}(i)\prod_{s=t+1}^{t+d}b_i(O_s)} \tag{2-72}$$

增加 $p_i(d)$参数的 HMM，比经典的 HMM 有更好的性能，这是以计算量和存储空间加大为代价的。特别是，估计出可靠的参数 $p_i(d),i=1,\cdots,N,d=1,\cdots,D$，要求训练数据量很多才能做到。

为了使 $p_i(d)$在有限的训练数据时也易估计，人们提出了另外一种在 Markov 链中考虑状态驻留时间的方法，即所谓参数方法：不去直接估计 $p_i(d)$的具体数值，而是假定 $p_i(d)$服从某种分布，通过估计描述这种分布的参数，来实现估计 $p_i(d)$的目的。例如，假定 $p_i(d)$服从 Gamma 分布[142,143]

$$p_i(d)=\frac{\eta_i^{v_i}d^{v_i-1}e^{-\eta_i d}}{\Gamma(v_i)} \tag{2-73}$$

这样，估计 $p_i(d)$的任务就转变为估计 Gamma 分布的参数 v_i 和 η_i。对于这种修正的 HMM， 描述 Markov 链的参数集就是(π,A,V,η)，其中 $V=(v_1,\cdots,v_N),\eta=(\eta_1,\cdots,\eta_N)$。根据前向-后向算法和 Baum-Welch 算法的思想，容易推导出概率 $P(O/\lambda)$的计算公式以及估计各参数的重估公式。为了解决这种 HMM 的重估公式等在计算上的下溢问题，除了用在 1.3.3 中介绍的增加比例因子方法之外，还可以用 Askar 和 Derin 基于后验概率用新的递归形式对前向-后向概率和重估公式进行运算的方法[20]，而且，重估公式更简单[85]。此外，假定 $p_i(d)$服从 Poisson 分布，也获得了成功[207]。

用参数方法获取 $p_i(d)$，虽然避免了直接估计 $p_i(d)$数值的非参数方法对训练数据量的太苛刻的要求，但计算量更大，而且，人为假定 $p_i(d)$服从某种分布，并不适用于所有状态。在此情况下，只估计各状态 θ_i 驻留时间的上限和下限参数，分别记为 u_i 和 l_i，对经典 HMM 进行修正，就是一个比较满意的选择[84]。这种修正

HMM 的参数集就为(π, A, L, U, B)，其中，π、A、B 为经典 HMM 的参数，$L=(l_1,\cdots,l_N)$，$U=(u_1,\cdots,u_N)$为新增加的描述各状态驻留时间最小值和最大值(下限和上限)的参数。为了推导的方便，记 $d(i)$为 θ_i 状态的驻留时间长度。那么，对于一个状态序列 $S=q_1, q_2,\cdots,q_T$ 而言，如果假定 Markov 链从 θ_1 状态出发，终止于 θ_N 状态，即如图 1.5(c)或(d)所示的左-右模型，那么，有

$$S=\underbrace{\theta_1\cdots\theta_1}_{d(1)}\underbrace{\theta_2\cdots\theta_2}_{d(2)}\cdots\underbrace{\theta_N\cdots\theta_N}_{d(N)} \tag{2-74}$$

这种修正 HMM 中的经典参数 π、A、B 仍可采用原有方法估计出，新增参数 L 和 U 的估计方法为：

设 d_{ik}为第 k 个训练序列 $O^{(k)}=O_1^{(k)},O_2^{(k)},\cdots,O_{T_k}^{(k)}$由 Viterbi 算法求出的最佳状态序列 Q_i^* 中状态 θ_i 的驻留时间，设共有 k 个训练序列，则

$$l_i=\min_{k=1}^{K}\{\max[d_{ik}/T_k,1/T_k]\}\quad \cdot,N \tag{2-75}$$

$$u_i=\max_{k=1}^{K}\{\max[d_{ik}/T_k,1/T_k]\},\quad i=\ :,\cdots,N \tag{2-76}$$

从上述上限和下限估计中可知，这种考虑 Markov 链中状态驻留时间的方法和采用参数方法估计 $p_i(d)$是同一类方法，因此，对训练数据量的要求并不高，但避免了参数方法估计出的 $p_i(d)$有时出现 d 太小或太大的可能性。事实上，这种估计 Markov 链中状态驻留时间上下限(归一化值)参数的方法，仍是一种参数方法，只不过假定 $p_i(d)$在某个上下限内服从均匀分布而已。实验表明，这是一种有效的考虑状态驻留时间的方法。

无论用非参数方法还是参数方法去考虑 Markov 链状态驻留时间，都改进了经典 HMM 在语音处理中应用的效果。对半连续 HMM，在 Markov 链中加入状态驻留时间参数，也能有很好的改进效果[92,93]。

值得指出的是，考虑状态驻留时间的影响，对汉语语音识别也是十分重要的，因为 HMM 的状态一般对应于某些基本语音单

元，比如有可能是辅音，而汉语辅音长度分布相对集中[5]，并不是经典 HMM 隐含的指数分布。针对这一问题，清华大学王作英等人提出了用非齐次 Markov 链去取代经典 HMM 中由 π、A 描述的 Markov 链的思想对经典 HMM 加以修正，并导出了相应的简单有效的训练和识别算法，在此基础上，构成了一个大词汇量汉语孤立词识别系统(赛德 919 系统)[6,7]。最近，战普明等人采用 KL (Kullback-Leibler)鉴别信息给出了这种修正 HMM 的距离量度[8]，并成功地用于语音粗分类[9]。这种修正 HMM 的基础是：非齐次 Markov 链的状态转移概率可以从事先假定的 $p_i(d)$分布得到，即有

$$\begin{aligned} a_{ii}(k) &= p_i(d \geqslant k / d \geqslant k-1) \\ &= \frac{p_i(d \geqslant k)}{p_i(d \geqslant k-1)} \end{aligned} \tag{2-77}$$

$$a_{ij}(k) = (1 - a_{ii}(k)) p_{i+1}(d=0) \cdots p_{j-1}(d=0) p_j(d \geqslant 1) \tag{2-78}$$

其它研究工作也表明，非齐次 HMM 的确是考虑 Markov 链驻留时间的一个有效的方法[203]。

2.2.3 二阶 HMM

所谓二阶 HMM，就是用二阶 Markov 链取代经典 HMM 中由 π、A 描述的 Markov 链而得到的一种修正 HMM[80,121,225]。而二阶 Markov 链，是指 Markov 链当前所处的状态，不仅与前一时刻的状态有关，而且还与前两个时刻的状态有关，即状态转移概率变为

$$\begin{aligned} a^{*}_{ijk} &= p(q_t = \theta_k / q_{t-1} = \theta_i, q_{t-2} = \theta_j) \\ & 1 \leqslant i, j, k \leqslant N \end{aligned} \tag{2-79}$$

相应地，初始概率除了矢量 $\pi = \{\pi_1, \cdots, \pi_N\}$之处，还有初始概率矩阵，记为 $\hat{A} = \{\hat{a}_{ij}, 1 \leqslant i, j \leqslant N\}$，其中

$$\hat{a}_{ij} = p(q_2 = \theta_j / q_1 = \theta_i)$$

$$1 \leqslant i,j \leqslant N \tag{2-80}$$

这样，描述二阶 Markov 链的参数为：π、$\hat{A}$、A^*，其中 $A^* = \{a_{ijk}, 1 \leqslant i,j,k \leqslant N\}$。

类似经典 HMM，可将三大基本算法推广为适用二阶 HMM 的情形。这种 HMM 不仅改善了语音识别的性能，而且，在手写体文字识别中也有成功的应用[123]。

§2.3 其它有代表性的 HMM 简要综述

自 80 年代中期以来，随着 HMM 在语音处理领域各方面的广泛应用，人们在经典离散 HMM 的基础上提出了各种形式的修正 HMM，除了已经给出的连续 HMM、半连续 HMM 和对 Markov 链修正后的 HMM 之外，这里将简要介绍另外几种有代表性的 HMM。

帧间线性预测 HMM[116]：

在 2.1.2 中介绍的线性预测型 HMM，是假定在语音帧内的一段信号是高斯 AR 模型产生的。这里为了很好处理语音信号本质上的非平稳性，假定一帧一帧的语音(特征)矢量序列是由一个非平稳的高斯 AR 模型产生，而且，AR 模型参数的变化由 Markov 链控制。这种 HMM 的基本思想为：设 Markov 链 $q_{t-1} = \theta_i, q_t = \theta_j$，那么，时刻 t 的语音(特征)矢量帧可以写成

$$\boldsymbol{Y}_t = \boldsymbol{A}(i,j) + \boldsymbol{B}_1(i,j)\boldsymbol{Y}_{t-1} + \cdots + \boldsymbol{B}_p(i,j)\boldsymbol{Y}_{t-p} + \boldsymbol{E}_t \tag{2-81}$$

其中，$\boldsymbol{Y}_t$ 和 $\boldsymbol{A}(i,j)$ 为 $d \times 1$ 矢量，$\boldsymbol{B}_1(i,j), \cdots, \boldsymbol{B}_p(i,j)$ 为 $d \times d$ 矩阵，预测误差 $\boldsymbol{E}_t$ 为 $d \times 1$ 随机矢量，且有均值为 0、方差为 $\boldsymbol{\Sigma}(i,j)$ 的高斯概密。显然，当 $\boldsymbol{B}_1(i,j), \cdots, \boldsymbol{B}_p(i,j)$ 均为 0 时，这种 HMM 就是 2.1.1 中介绍的通常意义上的高斯型连续 HMM。

利用前向-后向算法和 Baum-Welch 算法的思想，可以导出这种 HMM 的参数估计公式，即从 $\boldsymbol{Y}_{-p+1}, \cdots, \boldsymbol{Y}_0, \boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_T$ 中可以估

计出 $A(i,j)$，$B_1(i,j)$，…，$B_r(i,j)$ 以及 $\Sigma(i,j)$，也包括描述 Markov 链的 π、A 参数。大词汇量孤立词识别实验表明，这种 HMM 比通常的高斯型连续 HMM 要好。

状态插值 HMM[58]：

在大词汇量语音识别中，通常选择音素为基本语音单元。但同一音素在不同的上下文关系(context)中表现出来的特性很不一样，这就是所谓的音变(acoustic variability)问题。这里介绍的状态插值 HMM 提供了解决音变问题的一条新的途径。

状态插值 HMM 是根据语音感知机理中的轨迹(locus)理论而提出来的，它为元音提供了一个上下文相关的统计模型。其基本思想为：在辅音-元音(CV)转移过程中，元音 V 有 K 个转移状态，其均值矢量为 $m_V(k)$，$k=0,1,\cdots,K-1$，有一个稳定状态，均值矢量为 μ_V，辅音 C 的特征矢量为 v_c，称之为辅音的轨迹。设元音的状态插值 HMM 中每一状态的观察值概密为高斯函数，而且，所有状态的概密函数共享一个方差矩阵 Σ(实验表明，每个状态用不同的方差矩阵与共用一个方差矩阵没什么区别)，但元音 V 的 K 个转移状态的均值矢量是通过插值产生

$$m_V(k) = \lambda_k v_c + (1-\lambda_k)\mu_V \tag{2-82}$$

其中，λ_k 为插值权值，由先验知识选取，比如在元音两端各有一个转移状态时，可分别选取权值为 1/3 和 2/3。这样，模型参数就是辅音 C 的轨迹 v_C，$C=1,\cdots,\bar{C}$，元音 V 的稳定状态的均值矢量 μ_V，$V=1,\cdots,\bar{V}$ 以及所有元音所有状态共享的方差矩阵 Σ，其中，$\bar{C}$ 为辅音个数，$\bar{V}$ 为元音个数。这些参数可以由 Baum-Welch 算法类似的重估公式得到。此外，对元音-辅音(VC)转移过程也有类似的结果。

有辨别力的 HMM(Discriminte HMM)[41]：

一般认为，HMM 处理语音信号中统计和序列方面信息的能力很强，但它对模式的辨别能力较差。为此，对经典 HMM 加以修正而提出了有辨别力的 HMM，其基本思想为：对于测试序列 O 和

系统模型 λ_i，根据 Viterbi 得分的计算思想，有

$$p(\lambda_i/O) = \max_{q_1,q_2,\cdots,q_T} p(q_1,q_2,\cdots,q_T,\lambda_i/O)$$

$$= \max_{q_1,q_2,\cdots,q_T} [p(q_1,q_2,\cdots,q_T/O)p(\lambda_i/O,q_1,q_2,\cdots,q_T)] \tag{2-83}$$

这意味着识别分成两步来完成：第一步，声学译码过程，将输入的矢量量化语音序列转化成状态序列；第二步，进行音韵学和词汇方面的处理，而且，当状态序列已知之后，模型 λ_i 只和状态序列有关，而和 O 不再有关，这样

$$p(\lambda_i/O,q_1,q_2,\cdots,q_T) = p(\lambda_i/q_1,q_2,\cdots,q_T) \tag{2-84}$$

另一方面，第一项改为

$$p(q_1,q_2,\cdots,q_T/O) = p(q_1/O)p(q_2/O,q_1)\cdots$$
$$p(q_T/O,q_1,q_2,\cdots,q_{T-1}) \tag{2-85}$$

对于(2-85)式，假定每项仅与前一个状态有关，与序列 O 中宽度为 $2p+1$ 的一段有关，即有

$$p(q_t/O,q_1,q_2,\cdots,q_{T-1}) = p(q_t/O_{t-p},\cdots,O_{t+p},q_{t-1}) \tag{2-86}$$

这就是这种修正 HMM 的基础，它考虑了输入的上下文相关信息。而且，每一个观察值矢量是和一个转移相联系，并没有像经典离散 HMM 那样定义分开的状态转移概率和状态产生观察值概率。此外，这种 HMM 没有经典 HMM 那样的计算上的下溢麻烦(参见 1.3.3)。

提高 HMM 的模式分类辨别能力，是一个很受重视的问题，本书 4.1.2 还会涉及到。事实上，由此提出的修正 HMM 形式还有不少，例如基于成对的 Bayes 分类器的 HMM[115]。

第三章　HMM 在语音处理中的应用

70 年代，刚刚出现的 HMM 就被用于解决连续语音识别中的问题[26,28,105,106]。80 年代中期，由于 Bell 实验室 Rabiner 等人对 HMM 理论的详细介绍[144,191]以及成功用于孤立词识别的报道[192,196,197]，使得 HMM 成为各国从事语音处理的研究人员共同关注的一个焦点，在以后几年间，HMM 几乎在语音处理的各个方面都获得了极其广泛的应用，甚至在信号处理的相关学科中也能觅到 HMM 的踪影，例如，图象处理[62]、文字识别[123]、频率跟踪[29]（包括从周期图（periodogram）数据中跟踪调频信号的瞬时频率[226]）以及自然声音（natural sound）的建模和分类[230]等等。

本章分三节介绍 HMM 在语音处理各方面的应用，这些方面包括语音识别、语音增强和语音压缩。除了简要综述国内外有代表性的工作之外，本章着重介绍作者（包括所指导的硕士生、本科生）在过去几年间所做的工作。限于篇幅，本章对 HMM 在语音综合和说话人识别中的应用[153,218]，没有进一步介绍。

§3.1　语音识别

3.1.1　孤立词与连接词识别

孤立词识别，是语音识别最基本的一个研究课题，几乎所有语音处理中的方法，包括 HMM，都曾尝试过解决这一问题。绝大多数孤立词识别系统都采用了模式匹配的原理，系统构成如图 3.1 所示。

预处理包括滤波、放大、A/D 转换、端点检测等。而特征提取指求取表示孤立词特征的那些参数，例如短时幅度、能量、过零率、

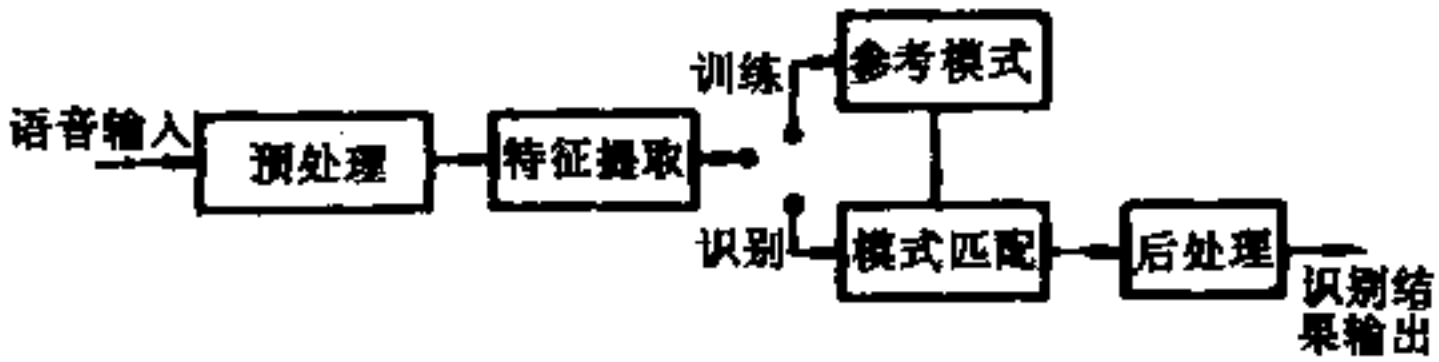


图 3.1　孤立词识别系统构成原理方框图

自相关函数、LPC 系数等等。每个参考模式对应系统词汇表中每个孤立词的特征参数。模式匹配就是度量待识别词的特征构成的测试模式与系统所存储的每个参考模式之间的距离，典型方法就是著名的动态时间弯析(Dynamic Time Warping，DTW)[195,209]。

基于 HMM 的孤立词识别系统的基本思想为：在训练阶段，用 HMM 的训练算法(例如 Baum-Welch 算法)，建立系统词汇表中每个词 W_i 对应的 HMM，记为 λ_i(对应图 3.1 中的参考模式)；在识别阶段，用前向-后向算法或 Viterbi 算法求出各个概率 $P(O/\lambda_i)$值，其中，O 为待识别词的观察值序列(对应图 3.1 中的模式匹配)；后处理就是选取最大 $P(O/\lambda_i)$值所对应的词 W_i 为 O 的识别结果。但是，对于不同类型的 HMM，送入 HMM 处理的观察值序列 O 有所不同，例如，对离散 HMM，一般求出语音特征参数(例如 LPC 倒谱系数)之后，还必须做矢量量化(Vector Quantization，VQ)，这样，O 就是由 VQ 码字序号组成的序列[196,197]；对线性预测 HMM，语音信号经过预处理即可，不必求取特征参数。O 其实就是一帧一帧的语音数据序列[109,113,187]；对于高斯型 HMM，语音信号经预处理、特征提取之后，成为一帧一帧的语音特征参数(如 LPC 倒谱)序列，这就是相应的观察值序列 O[108,110,192]。

当识别系统词汇表很小时，比如经常研究的 0～9 这十个数字音的识别问题，系统除了采用图 3.1 所示的形式之外，也可使用其它更灵活的方式。比如采用判决树的方式[248]。所谓判决树，就是根据大量训练数据的统计特性，选取典型语音特征，例如过零率、能量、基音周期等，对输入待识别语音进行分类，分别确定出判决十

个音的条件，形成一个有十个终节点的判决树来完成识别。这种方法尤其对非特定人识别系统有利。

但是，当识别系统词汇表比较大时，系统就必须在图 3.1 的基础之上考虑更多细致的问题。对汉语语音识别而言，词汇表选取汉语全部无调音节 400 多个或有调音节 1 200 多个构成的所谓全音节识别系统，就是一个典型的例子。首先，由于词汇表大，每词一个 HMM，总的系统训练好的模型参数占的存储空间过大，不利于系统的实时和实用化，因此，一种可行的方法是对 HMM 观察值概密函数(连续高斯型)进行量化，大大减少数据量。而且，识别时分级处理也是必要的。比如，第一级从无调音节中选出 40 个，第二级根据汉语四声的特点，结合有调音节，进一步识别出结果(6 个候选)[6,7]。另外，对于汉语全音节识别而言，选择组成音节的更小的语音单元，即汉语声母和韵母，作为系统 HMM 训练的基本单元，也是一个很好的选择[2]。这是因为，相对汉语 400 多个无调音节和 1 200 多个有调音节而言，总数仅 60 个左右的声母和韵母，其 HMM 能够更充分地得到训练，系统模型参数也将占用更少的存储空间。相应地，在图 3.1 所示的系统基本构成中，在预处理和特征提取之后，还必须做声母和韵母的分离工作。系统将分别进行声母、韵母的训练和识别。后处理之前，还应该运用汉语音节规则，将声母、韵母单独的识别结果组合成音节，从而得到最终的识别结果。实验表明，以声母、韵母为基本单元的全音节识别，是一个可行的方案[11]。当然实用的大词汇量识别系统在音节串识别之后，还必须进行音节(拼音)-汉字的转换。我国大陆以外的地区也出现了一些基于 HMM 的很出色的全音节识别系统[139]。

在词汇表很大时(例如 75 000 词)，选取音素作为 HMM 训练单元，是一个必然的选择[59]。当然，适当考虑音素的上下文关系，有利于系统性能的改善。

对于汉语语音识别来说，四声的辨识是一个很重要的问题。几乎所有的四声辨识方法都是从语音基音周期着手的。HMM 方法

也是如此。但在选择 HMM 处理的观察值序列方面，却各有不同。例如，可选为基音序列的增量调制[50]。但是，对于非特定人的四声识别，HMM 观察值序列最好这样选取[232]：设 $q_1, q_2, \cdots, q_N$ 为一个对数归一化的基音周期序列

$$q_i = \lg(p_i/p_0) \tag{3-1}$$

其中，p_i 为基音周期，p_0 是某个说话者对某个声调的平均基音周期。那么，定义观察值矢量为

$$u_k = \begin{bmatrix} c_k \\ d_k \end{bmatrix} \tag{3-2}$$

其中

$$c_k = q_{i+1} + q_i$$

$$d_k = q_{i+1} - q_i$$

$$i = 2k - 1, k = 1, 2, \cdots, N/2 \tag{3-3}$$

显然，c_k 是两个值的均值，而 d_k 则是增量的度量。由于 c_k 受 p_0 的影响，因而这样对非特定人识别有益。将观察值矢量作矢量量化处理，结合离散 HMM，对四声中每一声调都选择一个 HMM，构造一个如图 3.1 所示的基本系统，即可完成四声的识别。

HMM 在连接词的识别中也取得了很大的成功[195,199]。所谓连接词识别，就是指系统内存储的训练好的模式（或 HMM）是针对孤立的每个词的，但识别的语音却是由这些词构成的词串。例如，研究最多、也是最有实用价值的连接数字音识别，就是系统内存有 0～9 这十个音的参考模式，但要识别的数字串可能是 567，144 等。对于连接数字音的识别，一般的方法是根据统计模式识别的原理而得到的，并且识别率很高[46,200,201]。识别时，就是寻找参考模式串和待识别的数字串的最佳匹配，而且，系统的训练和孤立词识别有些不同，各个词的对应模式要由数字串数据训练[190,202]。

一个 HMM 连接数字音识别系统，如图 3.2 所示。识别时关键的三个步骤为

(1)LPC 谱分析

对语音信号 $s(n)$，求出加权的 LPC 倒谱和加权的增量倒谱矢

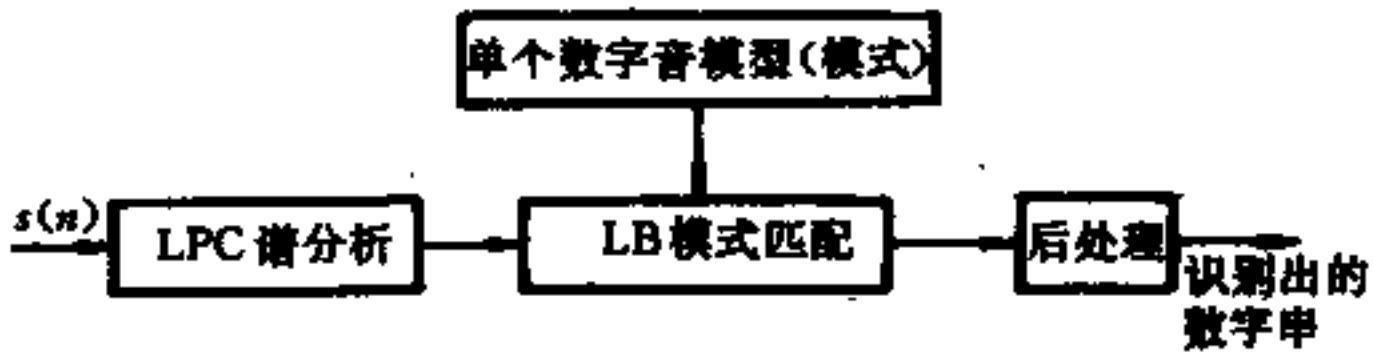


图 3.2　HMM 连接数字音识别系统方框图

量。

(2)LB 模式匹配

LB(Level Building)算法是这一步的关键。而这一步的处理也是整个系统的核心。对于输入的未知语音信号的谱矢量序列，由 LB 算法，与系统存储的单个数字音模式(HMM)的组合进行匹配(每一层用 Viterbi 匹配)。这一步的输出就是几个候选的不同长度的数字串(即每串中数字个数也可能不同)。

(3)后处理

对得到的几个候选数字串作进一步有效性测试，排除一些不可能的候选者，选出一个最佳的数字串作为整个系统的识别输出结果。

系统训练过程也比较复杂，由于是从连接数字串中产生各个数字音的模型，第一步就是要从这些数字串中分割出单个的数字。基于分割式 k 均值训练过程如图 3.3 所示。假定每个数字音的初始模型已知，训练数据就是各种不同长度的数字串语音，由 LB 分割算法，将数字串最佳分割成各个数字，再由 HMM 训练算法产生各个数字的新的模型，以替代初始模型。这一过程可迭代进行，直至收敛。关于 LB 算法的详尽介绍，可参见文献[196,199]。这里，表示每个数字音的 HMM，如图 3.4 所示。其中，观察值概密函数为混合高斯型，如(2-3)式所示。

无论是孤立词识别还是连接词识别，都假定待识别的语音由系统词汇表内的词组成，这对于实用化的系统而言，与真正面对的

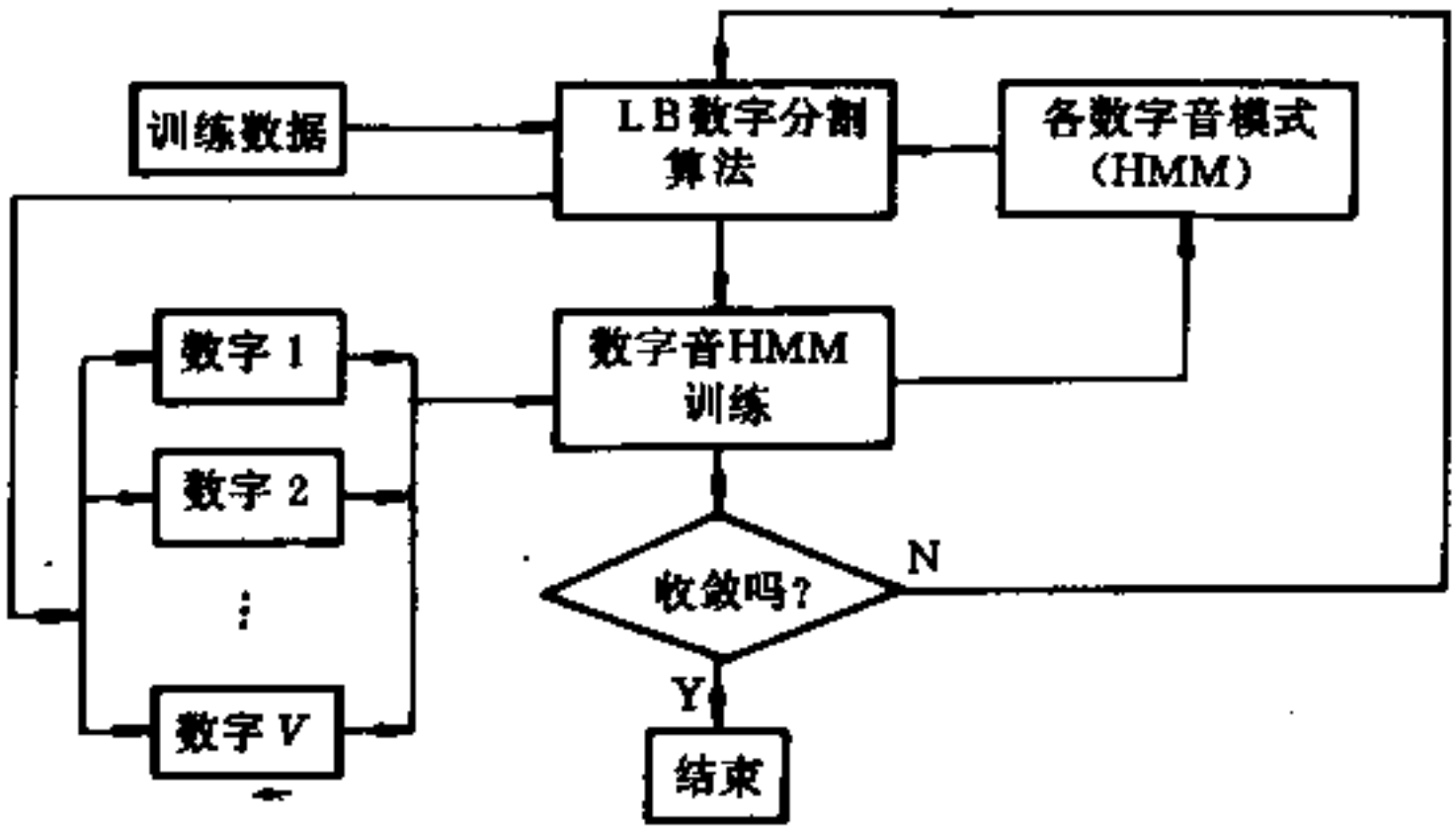
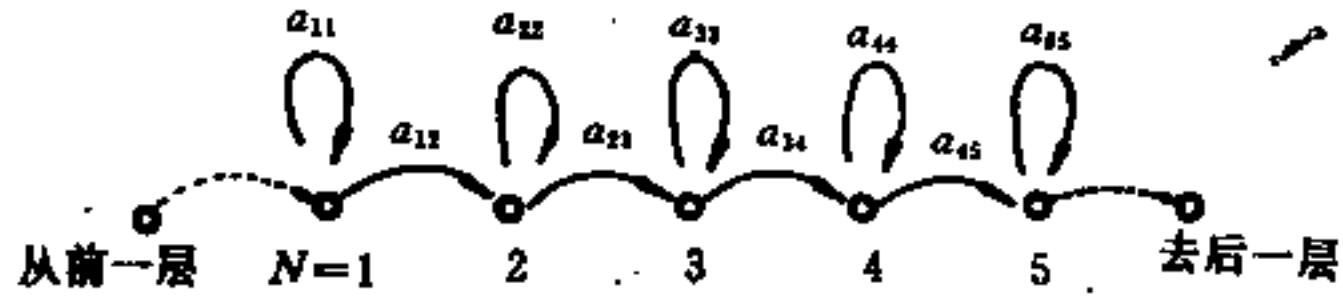


图 3.3　分割式 k 均值训练过程方框图



观察值概率函数	$b_1(0)$	$b_2(0)$	$b_3(0)$	$b_4(0)$	$b_5(0)$
能量概率	$p_1(E)$	$p_2(E)$	$p_3(E)$	$p_4(E)$	$p_5(E)$
状态驻留时间概率	$\hat{p}_1(d)$	$\hat{p}_2(d)$	$\hat{p}_3(d)$	$\hat{p}_4(d)$	$\hat{p}_5(d)$

图 3.4　表示数字音的 HMM 形式及参数

实际情景并不一致。比如电话自动查询，问话者很难严格按要求说出规定的词，这样在问话中除了规定的词外，还混杂了别的词，如果仍用孤立词或连接词算法来识别，必然会产生很多错误。因此，从未严格限制的语音中识别出所需的关键词，就是一个很有意义的课题。虽然已有一些采用 DTW 算法的系统[38,91,169]，但使用 HMM 算法的系统取得了更好的效果[226]。这里的 HMM 算法是在上述的连接词识别过程的基础之上修正而得到的。不仅建立了关

键词的 HMM，而且，对不需要的语音和背景信号，也建立相应的 HMM。识别时，假定每个发音中仅有一个关键词，但有别的语音和背景信号，受一个简单的语法约束，如图 3.5 所示。即无关语音（或背景）后面，可以是关键词，也可以是无关语音（或背景）。这样，图 3.5 中 0 是起始节点，1 是终止节点。采用 HMM 的关键词识别研究，还有很多出色的工作[205,229]。

图 3.5　用于识别混在无关语音和背景中的关键词的语法

3.1.2　音素 HMM 连续语音识别

连续语音识别，是语音识别中一个难度相当大的课题。这是因为连续语音句子中各单词发音没有明显的间隔或声学标志，因此，分割起来格外困难。另外，连续语音中各单词之间 发音会相互影响，如连读等。人们对连续语音识别进行了大量的研究，取得了很大的进展，并在解决这一难题的过程中，引用了很多复杂而有效的方法，例如，将语音学（phonetics）、音素学（phonemics）、词素学（morphemics）、韵律学（prosodics）、句法（syntax）和语义学（semantics）等知识和人工智能相结合的语音理解方法[55,118]；采用 Fuzzy Set 的方法[56]；神经网络方法[222]等等。

HMM 出现之后，人们对 HMM 在连续语音识别中的应用非常重视，进行了广泛的研究，取得了很大的进展，从较早的 Dragon 系统等有代表性的工作[26,28,105,106]，到最近的大量研究成果报道[81,88,168,185]，包括一些知名度很高的工作，例如美国 BBN 的 BYBLOS 系统[53]；日本京都 ATR 做的 HMM-LR 系统[87,117]；德国的 SPICOS 系统及改进后的 10 000 词系统[172,173,215]；英国 RSRE 研制的 ARM 系统[208]；美国 SRI 的 DECIPHER 系统[54]；

美国 MIT 的 Lincoln 实验室的系统[180]；日本的 NINJA 系统[104]；美国 CMU 的 SPHINX 系统[132,133,134,135,136]等等。1990 年欧洲七国合作进行的旨在识别和合成七种语言的语音的庞大课题 Polyglot (Esprit 2104)，计划做三类连续语音识别系统，其中之一也是采用 HMM 方法[221](作者当时在法国 LIMSI-CNRS 参加了 HMM 连续语音识别系统的研制工作[241])。因此，可以这样说，HMM 是解决连续语音识别最主要和最有希望的方法之一。在我们实验室里，基于矢量量化(VQ)和离散 HMM，也建立了一个大词汇量连续语音识别系统——HUST 系统，下面将给出较为详细的介绍：

(1)HMM 训练的基本语音单元选择

通过 3.1.1 孤立词和连接词识别的讨论可知，在小词汇量时，一般 HMM 被用来表示词(words)，但当词汇量增大时，容易混淆的词汇显著增加，此外，由于不可能要求训练数据中每个词出现很多重复，因此，难以得到可靠的词 HMM。这样，必须选取比词更小的单元作为 HMM 训练的基本语音单元。另一方面，在确定基本语音单元时，通常要考虑以下两个方面[133]：

a. 一致性，即同一基本语音单元在不同情况下出现时应具有相似性；

b. 可训练性，即基本语音单元在训练数据中要有足够的重复出现次数。显然，这两点是相互矛盾的：一些较大的单元，如音节、半音节等，具有较好的一致性，但却难以训练；而一些较小的单元，如音素，虽然容易训练，但它的一致性却较差。权衡利弊，在 HUST 系统中采用了音素作为 HMM 训练的基本语音单元。

(2)实验用语音数据库[238]

由于目前尚无合适的大词汇量连续语音库，为了能评价 HUST 系统的性能，我们建立了一个 1 000 词汇的汉语连续语音库，HUST 系统所进行的各项实验都是在该连续语音库上完成的。

我们根据文献[12]的统计结果，用出现频度最高的 1 020 个

汉语词汇构成语音库的字典，即词汇表，并请现代汉语方面的专家根据这字典中各词汇的出现频度，造了 581 个句子，这些句子覆盖了字典中的所有 1 020 个词汇。所有 581 个句子都由一成年男性以正常语速朗读并录音。

系统的音素是参照《汉语拼音方案》选取的，由声母和韵母构成，共 59 个，其中声母 21 个，加上 1 个零声母(@)，韵母 35 个加上儿化音(er)和静音(#)(静音表示没有语音)。

整个语音数据库由两部分组成：语音信息描述部分和语音数据部分。前者提供各句子有关的音素、词汇等方面的描述，后者提供各句子的语音数据。

最基本的语音描述信息是以下三个文件：音素文件(列出语音库中的所有基本语音单元——音素的名称和标号)，词汇字典文件(列出语音库中的所有词汇的名称及其相应的音素描述)和句子文件(列出语音库中的所有句子)。

若要进行全音节识别，则需要一个全音节字典文件(列出语音库中的所有音节的名称及其相应的音素描述)。

若要进行音素识别，则需要一个音素字典文件。

此外，为了给连续语音识别系统训练提供一组初始化参数，语音库还根据语音信号的波形(手工)做了一组音素初始化参数文件(共由 34 个句子组成，分别描述这 34 个句子中各音素的起止帧位置信息)。

为了得到所需各种语音信息描述文件，这里建立了一个强有力的工具软件，可以进行如下处理：

a. 字典校正。由于原始字典文件中可能会出现相同的词(词及其音素描述完全相同)，因而必须将重复的词删除。此外，由于汉语中普遍存在着一词多音的情况，因而有必要将多音词区别开来，避免在训练和识别时造成混淆。语音库的词汇字典文件原有 1 020 个词汇，经校正后实际为 941 个词汇。

b. 统计所有音素在字典中的出现情况。

c. 统计在一组句子中词汇和音素的出现情况。

d. 统计字典中的所有词汇在一组句子中的出现情况。

e. 根据字典，由原始句子产生训练时需要的句子音素描述文件。

f. 根据字典，将原始句子文件中以连续汉字形式输入的句子，分割成有间隔的以词汇形式组成的句子。

g. 根据字典和以词汇形式组成的句子文件产生识别时将用到的 word-pair 句法文件。

h. 根据全音节字典文件，将以音素形式组成的句子文件转换为以全音节形式组成的句子文件。

i. 根据字典和以词汇形式组成的句子文件，产生任意音素指定最小出现次数的系统训练时所需的初始化句子子集。

录在磁带上的语音，经 Compaq 486 计算机内的一块 TMS320C25 板采样，10kHz 采样率，12bit 量化，然后用一个系统转移函数为 $H(Z)=1-0.97Z^{-1}$ 的滤波器预加重。对于用一个帧宽为 200 点(即 20ms)、帧移动步长为 100 点的汉明窗截取出的每一帧语音，用自相关法求取它们的 12 阶 LPC 系数，再求出相应的 12 阶 LPC 倒谱系数，然后，对 12 阶 LPC 倒谱系数进行双线性变换。最后，利用 Kohonen 的自组织特征映射(SOFM)神经网络对每一帧的 12 阶双线性变换倒谱系数进行矢量量化。矢量量化的码本大小为 256，该码本是 554 个句子的近 160 000 帧 12 阶双线性变换倒谱系数训练产生的。整个语音在经过处理后，词汇字典实际为 941 个词汇，全音节字典实际为 408 个全音节，音素字典实际为 59 个音素，音素文件实际为 59 个音素，实际采用的句子为 554 个。

(3)基本识别系统

HUST 系统采用离散的与上下文无关的音素 HMM。每个 HMM 具有 3 个状态，左-右形式，允许自环和跳跃性状态转移。HUST 系统训练和识别时的输入语音数据均是已经预处理后的

各个句子的 VQ 码字文件。在训练时用的是 Viterbi 算法，在识别时则用状态 Viterbi 算法处理状态级数据，用词汇 Viterbi 算法处理词汇级数据。

系统的训练大致可分两步：初始化和对训练句子进行分割处理。初始化时，产生系统的 59 个初始音素 HMM 参数。这些参数分成 *A*、*B* 两个矩阵，*A* 矩阵为三维：59×3×3(59 个音素，每个音素 HMM 有 3 个状态，每个状态只有 3 个可能的状态转移——自环、转到下一个和下两个状态)，对于单个的 3 状态 HMM，可认为还有两个虚拟的状态存在。*A* 矩阵初值为等概率，存放取整后的对数概率值。*B* 矩阵也为三维：59×3×256(59 个音素，每个音素 HMM 有 3 个状态，每个状态观察值为 256 个码字之一)。*B* 矩阵初值可根据手工分割的 34 个句子估计出，也是存放取整后的对数概率值。对训练句子进行分割处理，是由 Viterbi 算法完成的。分割时，读入一个训练句子的语音数据(VQ 码字序列)之后，还要读入其相应的音素描述信息。例如，我们实验用语音数据库中的第一个句子“一只白鸟飞了”的音素描述信息为

@ # @ I ZH I B AI N IAO F EI L E @ #

这里，@表示零声母，#表示静音(没有语音的空白段)。根据句子的音素描述信息，每个音素对应一个 HMM，对应 3 个状态，依次排列，得到一系列状态(上例中，有 16 个音素对应的 48 个状态)。由初始 *A*、*B* 矩阵，可知这些状态之间的各个状态转移概率(自环、转到下一个和下两个状态)以及各状态的输出观察值矢量的概率。那么，对于读入的这个句子的一串 VQ 码字(观察值序列)，由 Viterbi 算法，可求出最佳累积概率以及相应的最佳状态序列，前者将影响训练过程的迭代次数，后者就对这一句子产生了最佳分割。分割完这一句子之后，由分割结果去修改三组计数器的值：每个音素的每个状态停留次数(帧数)，每个音素每个状态观察到每个 VQ 码字的数目，每个音素每个状态自环、转向下一个和下两个状态的次数。当所有训练句子处理完后，这三组计数值将产生这一

次迭代的新的 A、B 矩阵值，为下一次迭代作好了准备。

训练部分的算法描述为：

a. 读入系统中所有音素的名称和标号，并返回音素 HMM 的个数；

b. 利用语音库中专门用于初始化的句子对 HMM 参数 A、B 作初始化；

c. 读入训练句子的目录文件；

d. 读入一个句子的语音以及相应的音素描述信息；

e. 用 Viterbi 算法分割句子；

f. 若训练句子的目录文件已读完就去 g，否则去 d；

g. 利用分割结果，更新 HMM 参数 A、B。若新旧参数之间的差距大于指定的门限则去 c，否则去 h；

h. 存放各个音素的 HMM 参数；

i. 结束。

系统的识别，采用了著名的经典识别算法——时间(帧)同步 Viterbi(beam)搜索算法[27,138,211]。不考虑句法时，识别过程很类似连接词识别中的一次通过算法(one-pass algorithm)。将词汇表中所有词对应的所有音素的所有状态排成一排(词之间有虚拟状态)，由训练时产生的 A、B 矩阵，就可知这些状态之间的转移概率和输出观察值矢量的概率。像系统训练过程一样，读入一个待识别句子的语音数据，由 Viterbi 算法可得到一个最佳状态序列，相应地也就得到了该句子的识别结果。由于 Viterbi 算法每次用输入的待识别句子的一帧数据处理完排在一起的所有状态，因此，识别过程是时间(帧)同步的。但是，在考虑句法时，识别过程就复杂一些。这里的句法指 word-pair 句法和 bigram 句法，前者列出了词汇表中每一个词后可跟的其它词，后者不仅如此，而且还给出了该词所跟各词的可能性(概率)。在构造基本识别系统时，没有足够大的语料库来统计可信的句法，因此，只使用了根据实验用语音库中的 581 个句子统计而成的一个 word-pair 句法，但程序略为修改一下

就能使用 bigram 句法。这时，对于输入的待识别句子的一帧语音数据，先由状态 Viterbi 算法处理各词内各状态之间的转移，再将每个词最后一个状态处的累积概率和回溯路径带入相应的词汇 Viterbi 算法处理过程中，在句法信息帮助下，处理词汇之间的转移，完成之后，保存最终回溯时需要的路径信息，而且，将这里得到的相应累积概率再返回状态 Viterbi 算法中，为下一帧输入语音数据的处理作好准备。对所输入句子的所有帧语音数据处理之后，再回溯，就得到该句子的识别结果。

识别部分算法描述为：

a. 读入系统中所有音素的名称和标号，并返回音素 HMM 的个数；

b. 读入语音库的字典，得到系统中各词汇的名称及其音素描述信息，返回词汇个数；

c. 读入句法，返回句法节点个数；

d. 读入所有音素的 HMM 参数(A、B 矩阵)；

e. 读入待识别句子的目录文件；

f. 读入一个待识别句子的语音；

g. 对于该句子的每一帧语音：

a)对于字典中的每一词汇的所有音素的所有状态进行状态 Viterbi 处理；

b)对于字典中的每一词汇进行词汇 Viterbi 处理(利用句法信息)；

h. 对经过帧同步过程处理后的结果进行回溯，输出识别后的句子；

i. 若识别句子的目录文件未读空则去 f，否则去 j；

j. 结束。

(4)识别系统性能评估

性能评估的目的是得到系统对句子识别输出结果中正确和各种错误的百分比。为此，我们以比较处理两个句子(一个是标准句

子，另一个称之为目标句子——系统对标准句子语音的识别输出）的算法为基础编制了一个连续语音识别系统性能评估软件[244]。这个软件的核心是比较处理两个句子。所谓比较处理两个句子，就是通过对它们的比较，输出有关系统性能的重要数据，包括正确识别的词及数量、替代误识词及数量、消去词及数量、插入词及数量。由于各种识别系统性能差异很大，语言中句子也有很大随机性，因此，系统识别输出结果千差万别。这样，比较处理两个句子的算法必须能广泛适用于各种情形。这里的算法是受到人的判决过程启发而得到的。当人比较两个句子的相似程度时，往往是先确定两个句子中相同词的对应关系，并在此基础上得到哪些词被误识成另外一些词（替代错误）；哪些词原来没有而识别输出后多出了（插入错误）；哪些词原来有而输出识别结果时没有了（消去错误），根据这些结果就可以判断识别系统性能的优劣。因此，评估软件首先把句子转化成链表以便于使用，然后取目标句子链表中每个节点和标准句子每个节点的单词域相比较，若单词相同（可有多个），则把两个句子相同单词之间的匹配对应关系转化成一个树形链表结构。然后就对树进行遍历，可得到两个句子之间单词的所有对应关系，从中选择一种对系统性能最有利的关系为合理的对应。当确定了两个句子正确识别单词的对应关系之后，其余的各种识别错误就容易统计了。比较处理两个句子的算法共分六个步骤：

a. 把句子转化成链表

这里采用链表结构是为了适应各种长度的句子，充分利用内存，且便于进行顺序搜索。链表节点采用如下所示的结构，共分四个域：len 为单词长度，location 为单词在句子中的位置（由 1 开始），word 为指向单词的指针，next 为指向下一节点的指针。

```
typedef struct sentenceptr{
        int len;
        int location;
        char * word;
```

```
        struct sentenceptr * next;
}SENTENCEPTR;
```

b. 寻找对应关系

取目标句子链表中每个节点的单词与标准句子链表中的每个节点的单词相比较，若相同(可有多个)，则形成一个树节点，并调用建树过程把此节点插入树中所有合适的位置(合适的位置可有多个，下面将会讨论到)。树节点采用下面定义的结构：sl，dl 分别是单词在标准句子和目标句子中的位置，brother 和 son 分别是指向树形链表结构中的兄弟节点和子节点的指针。

```
typedef struct treeptr{
        int sl, dl;
        struct treeptr * brother, * son;
}TREEPTR;
```

c. 建树过程

这里的任务是把上一步中得到的节点正确插入到树形结构的所有适当位置。树形链表结构中节点的位置反映了两个句子中相同单词对之间的关系。两个句子中匹配词对(单词相同，位置不一定一致)之间的关系可分两种：一种是相容关系，如图 3.6(a)中的 A↔A，B↔B 和 C↔C 这三个匹配词对；另一种为不相容关系，即不能共存的匹配词对，如图 3.6(b)中的 A↔A，A↔A，(c)中的 A↔A，B↔B，(d)中的 A↔A，A↔A。不相容关系中的匹配词对只能存在一种。

节点在树形链表中的位置反映了节点代表的单词对之间的关系。互为兄弟节点的单词对具有不相容的关系，互为父子节点的单词对具有相容关系。判断两个节点是否相容可通过比较节点中单词的坐标位置来进行，如有两个代表匹配单词对的节点 P1 和 P2，如果有

$$P1.sl < P2.sl, \text{且 } P1.dl < P2.dl$$

则这两个节点表示的匹配单词对是相容的，P2 可插入到 P1 的

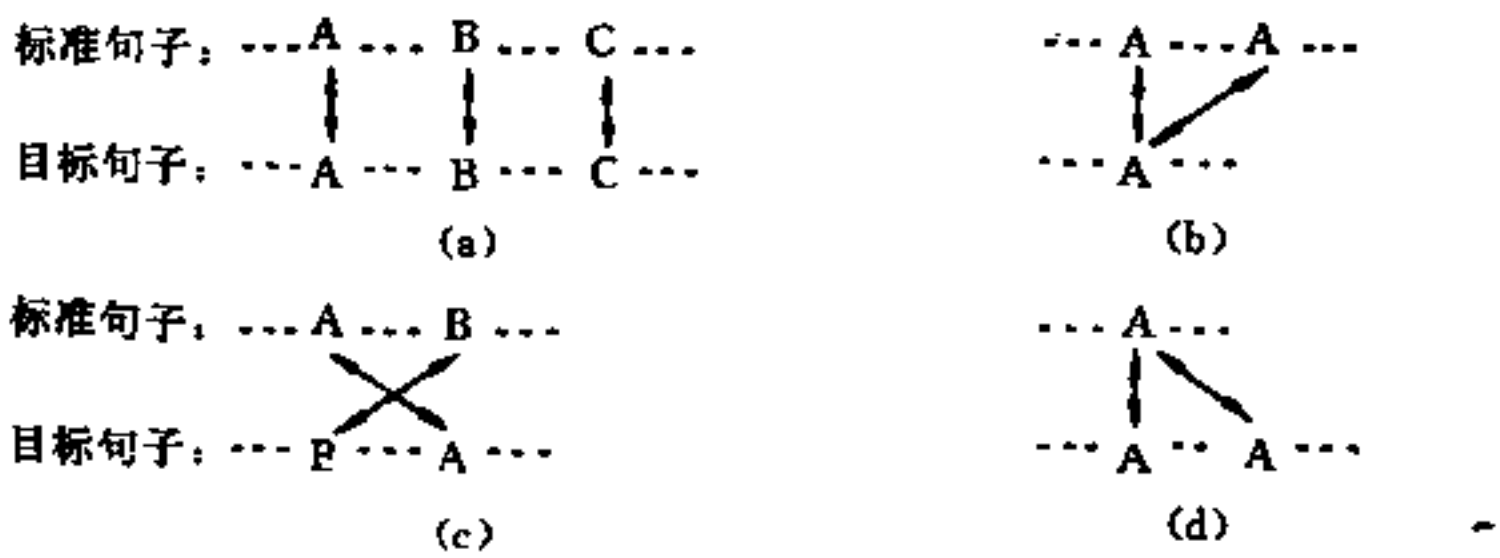


图 3.6　词对的相容和不相容关系

(a) 相容关系；(b)、(c)、(d) 不相容关系

son 指针下面，否则，P1 和 P2 不相容，P2 可成为 P1 的兄弟节点，如果 P2 经判断没有插入到 P1 的兄弟节点的子指针之下的话。图 3.7 给出了两个句子转化成树形链表结构的例子。

建树中采用了类似于广度优先搜索算法的递归算法，先沿兄弟节点进行搜索，直到指针为空。再沿子节点方向搜索，直到把要插入的节点插入到合适位置。

d. 遍历过程

当树形链表结构建好后，对树进行遍历就可得到两个句子中单词的各种对应关系。从建树过程可知，沿着树节点的子指针方向往下搜索到底，就可得到一种对应关系。若把其中经过的某节点换成其兄弟节点，再沿着这节点的子指针往下搜索又可得到另一种对应关系。按这种方法进行递归搜索，就可得到两个句子中单词对应的所有情形，从中选择一种对系统性能最有利的对应关系为合理的结果。对图 3.7 所示的例子中的树进行搜索，可得到如图 3.8 所示的四种对应关系。显然，其中的图 3.8(c)和(d)为对系统有利的对应关系，可从中选择一种作为这两个句子真实的对应关系。

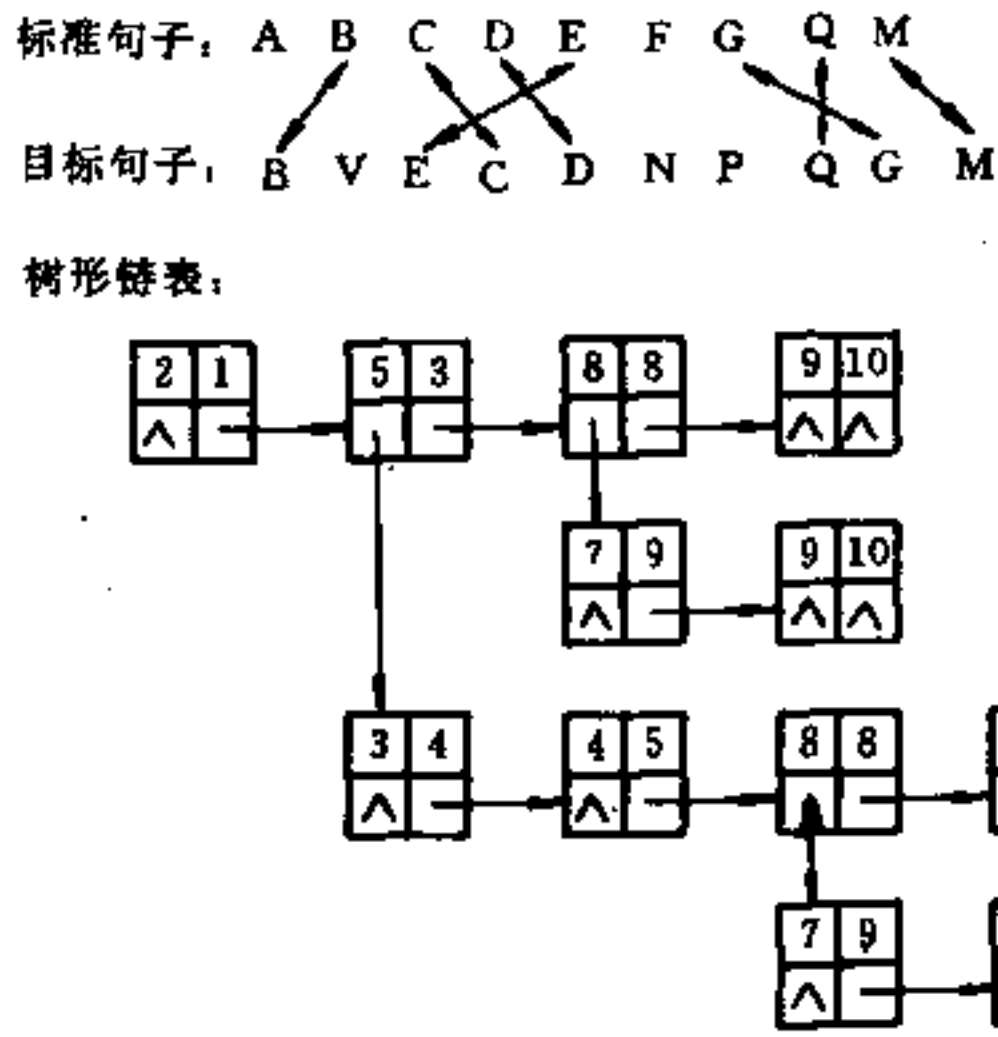


图 3.7　从句子到树形链表结构举例

注：Λ表示空指针；字母表示词

搜索采用递归方法，使用类似于深度优先搜索算法的方法，先沿子节点方向搜索，再转入兄弟节点。每进入一个节点就做一个标记，退出时删除标记。标记为一整型数组 result，其下标为节点的 dl，数组值为 sl，即有

$$result[dl]=sl$$

上式表示目标句子中坐标为 dl 的单词和标准句子中坐标为 sl 的单词相对应。若有 result[dl]=0，表示目标句子中坐标为 dl 的单词和标准句子中的单词无对应关系。例如，对图 3.8(c)，有：result [1] =2; result [4] = 3; result [5] = 4 ; result [8] = 8; result [10] =9;其余值为零。

e. 对应关系合理性判断

搜索得到的对应关系并不都是合理的，例如下面两个句子：

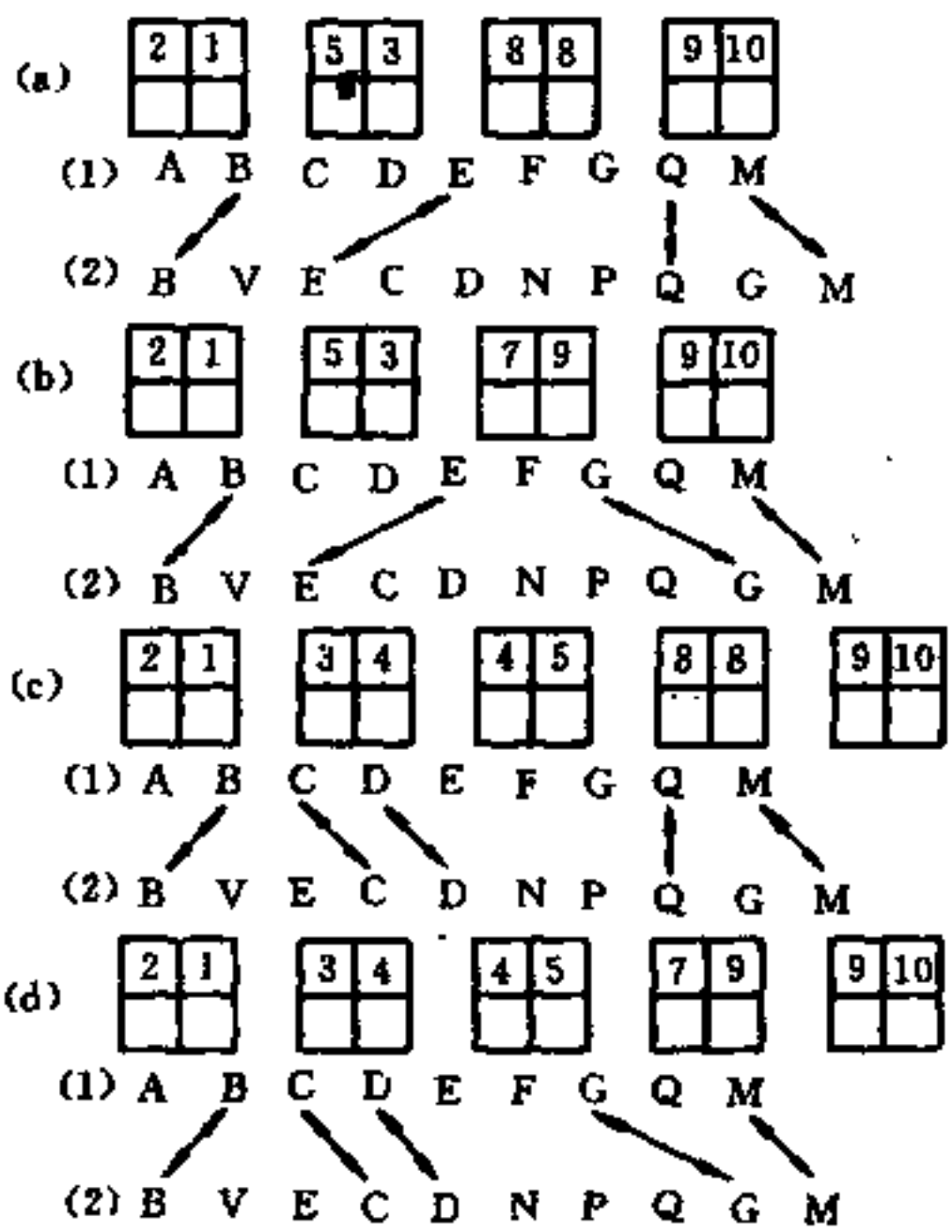


图 3.8　两个句子单词间所有可能对应关系

(a)、(b)、(c)、(d)对应关系

注：(1)为标准句子；(2)为目标句子

标准句子：ABCDEFGHIJK

目标句子：KLMNOPQRSTU

若认为目标句子中的 K(坐标位置为 1)是对标准句子中坐标位置为 11 的 K 的正确识别，显然是不合理的。如果那样认为，就将得出结论：A 到 J 为消去错误，L 到 U 为插入错误，一共有 20 个错误。合理的判断应该是：K 到 U 是对 A 到 K 的替代错误，这样，一共有 11 个错误。当然，合理的评价对系统性能是有利的。

在软件中，判断单词之间的对应关系是否合理，采用的方法

为：若有三个相邻的对应，即 dl_1、dl_2、dl_3 分别与 sl_1、sl_2、sl_3 对应，显然，对于 dl_1 到 dl_3 和 sl_1 到 sl_3 这部分单词识别错误的总和不会大于 (sl_3-sl_1) 和 (dl_3-dl_1) 的最大者，否则，dl_2 与 sl_2 的对应关系是不合理的，应予以取消，即令：result$[dl_2]=0$，这样，错误最多只会是 (sl_3-sl_1-1) 和 (dl_3-dl_1-1) 两者的最大值。这样的结果显然对系统性能评价是最为有利的。用这种方法逐一判断每三个相邻的匹配对，从而减少一些不合理的对应关系，客观地评价系统性能。

f. 统计识别性能数据

在确定了两个句子正确识别的单词间的对应关系后，就可确定其它的识别错误了。若 dl_1 和 dl_2 为 result 数组中相邻的两个非零值，且有：$dl_1<dl_2$，result$[dl_1]=sl_1$，result$[dl_2]=sl_2$，则坐标位置为 dl_1 和 dl_2 的单词为正确识别的单词；若 $dl_2-dl_1<sl_2-sl_1$，则目标句子中坐标位置为 dl_1+1 到 dl_2-1 之间的单词是对标准句子中坐标位置为 sl_1+1 到 $sl_1+dl_2-dl_1-1$ 之间单词的替代错误识别；对标准句子中坐标位置为 $sl_1+dl_2-dl_1$ 到 sl_2-1 之间单词的消去错误识别。若 $dl_2-dl_1>sl_2-sl_1$，则目标句子中坐标为 dl_1+1 到 $dl_1+sl_2-sl_1-1$ 之间的单词为标准句子中 sl_1+1 到 sl_2-1 之间单词的替代错误识别。此时，目标句子中坐标位置为 $dl_1+sl_2-sl_1$ 到 dl_2-1 的单词为插入识别错误。对于图 3.8(c)中的结果，有

B↔B，C↔C，D↔D，Q↔Q，M↔M 为正确识别；

E↔N，F↔P 为替代错误识别；

A、G 为消去错误识别；

V、E、G 为插入错误识别。

应该指出，对于一般连续语音识别系统输出结果句子，极少出现树形链表分叉的情况，建好的树往往是一条单链表。但为了应付各式各样的复杂情形，本软件还是采用了上面讨论过的广泛适用的通用算法。

处理部分在主程序中以循环调用比较两个句子的函数来实现对整个文件的处理。每处理完两个相应句子的比较，就根据处理结

果修改用于统计整个系统识别性能的各个计数器的值，并相应修改替代误识词队列、消去词队列和插入词队列。

当然，上面的算法是假设目标句子中至少有一个词识别正确的情况下而提出的，但如果整个句子都识别错了，软件将直接统计这个句子的识别结果数据。

(5)实验结果[238]

我们利用语音库中实际采用的 554 个句子中的 476 个句子对系统进行了训练，然后用系统对所有 554 个句子(包括未参加训练的 78 个句子)分别用词汇字典，全音节字典和音素字典进行了识别。最后，用刚刚介绍的性能评估软件计算出了识别率及各种错误百分比。

a. 1 000 词汇连续语音句子中词识别

采用大小为 944 的词汇字典，在无句法和有 word-pair 句法两种情况下获得的实验结果参见表 3.1。

表 3.1　1 000 词汇连续语音句子中词识别结果

句法	句子总数	词汇总数	正确	消去错误	替代错误	插入错误
无	554	3 824	24.6%	15.2%	60.2%	6.1%
word-pair	554	3 824	77.8%	5.1%	17.1%	8.6%

这里所说的无句法是指句子中每个词后面允许跟字典中的任何一个词。

所采用的 word-pair 句法是根据 581 个句子产生的，它给出了字典中任一个词的后面可跟的其它词。

b. 连续语音全音节识别

采用大小为 408 的全音节字典，在无句法和有 syllable-pair 句法两种情况下获得的实验结果参见表 3.2。

表 3.2　连续语音全音节识别结果

句法	句子总数	全音节总数	正确	消去错误	替代错误	插入错误
无	554	4 098	29.3%	8.2%	62.5%	6.5%
syllable-pair	554	4 098	62.9%	5.7%	31.4%	6.9%

这里，所用的 syllable-pair 句法是根据 581 个句子产生的。

c. 连续语音音素识别

采用大小为 59 的音素字典，在无句法和有 phone-pair 句法两种情况下进行了识别，实验结果参见表 3.3。phone-pair 句法是根据以下原则产生的：

a)所有声母的后继音素依照全音节字典产生；

b)任一韵母的后继可以是全部声母；

c)音节结构为声母＋韵母结构。

表 3.3　连续语音音素识别结果

句法	句子总数	音素总数	正确	消去错误	替代错误	插入错误
无	554	12 826	33.9%	22.5%	43.6%	7.9%
phone-pair	554	12 826	39.4%	20.2%	40.4%	6.1%

(6)改进系统的努力方向

a. 突破 1 000 词汇的限制：对识别更大词汇量(如 50 000 词汇)的连续语音，除了采用更大词汇量的字典之外，另一个可行的方法，是以我们进行的连续语音全音节识别工作为基础，再加上汉语四声识别和拼音(音节)-汉字转换部分。目前，我们采用最长词匹配技术做的拼音-汉字转换部分，正确率约 92.5%[240]。

b. 采用多码本技术：由于用单码本量化要比用多码本量化产生大得多的误差[86]，在考虑进一步引入差分倒谱和能量时，决定采用三个码本：一个 12 阶双线性变换 LPC 倒谱系数码本，一个 12 阶差分双线性变换 LPC 倒谱系数码本和一个能量及差分能量码本。

c. 采用改进的离散 HMM 取代目前的经典 HMM 去表示音素：比如考虑了 HMM 中的状态驻留情况(state duration)的 HMM，因为经典 HMM 假定状态驻留时间分布为指数分布，与大多数物理真实情况不符，改进后的 Gamma 分布或别的分布，均有利于性能改善，参见 2.2.2。

d. 采用更完善的句法：比如从更大语料库中总结的 word-pair 句法，或 bigram 句法等。

e. 克服说话人的影响，向非特定人识别系统过渡，参见 1.4.2。模型训练时采用纠错训练等措施，参见 1.4.3。

f. 采用上下文相关(context-dependent)的音素作为 HMM 基本训练单元：与目前使用的上下文无关(context-independent)音素单元比起来，对系统识别率也有明显的提高[243]。所谓上下文相关音素，就是考虑一个音素与其左或(和)右相邻音素的相关情况后选取的音素。这样，对于 N 个音素(上下文无关)，就可能存在 N^2 个左或右上下文相关(left or right context dependent)音素，可能存在 N^3 个左和右上下文相关(left and right context dependent)音素。选取这样的语音基本单元的依据是：在人的语音产生过程中，音素之间是高度相关的。如果只考虑左上下文相关音素作为 HMM 训练的基本语音单元，音素的总数就从 59 个激增至 1 000 个左右，比如音素“ei”就成为“b:ei”、“f:ei”等等。对原来的系统的修正主要有三部分，即读入音素名称和标号部分，读入训练句子相应的音素描述信息部分以及识别时读入词汇表(字典)中各词汇的音素描述部分。

g. 对识别时采用的帧同步 Viterbi(beam)搜索算法进一步优化，注意路径的裁剪，减少搜索空间，缩短句子的识别时间。

(7)具有开发价值的原型样机系统

在前面介绍的工作基础之上，受(1994～1995)国家 863 计划项目资助，我们正在研制一个具有开发价值的原型样机系统——采用 HMM 方法的 50 000 词书面语连续语音识别系统，其基本

框架为[259]：系统是在一台 Compaq486/33 微机上实现。为了减少处理时间，使系统达到实时性要求，在微机内插了一块 ADSP21020 开发/高速处理多功能卡，这样，系统的大致硬环境如图 3.9 所示。系统识别的基本单位为连续语音句子，语音输入时，

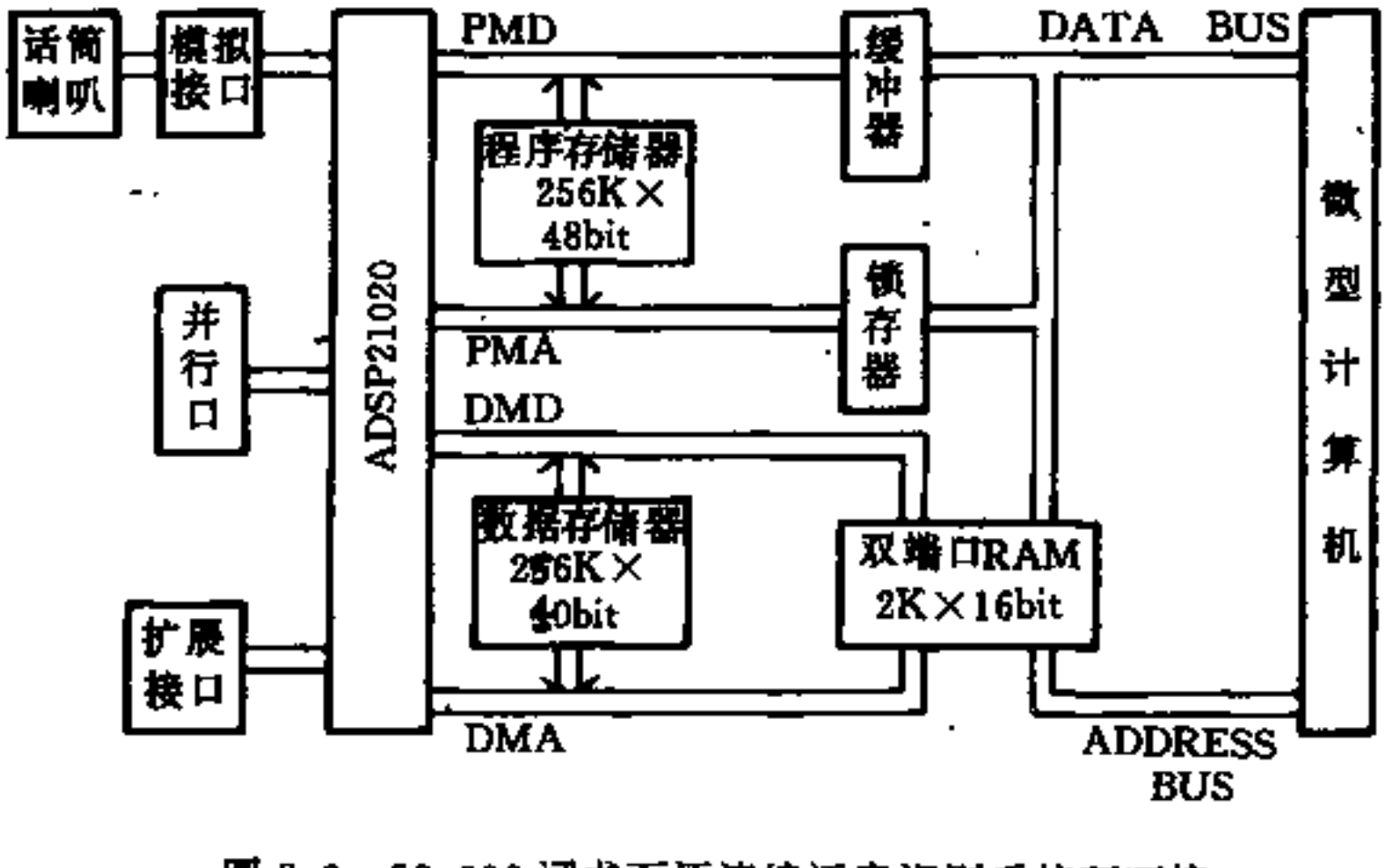


图 3.9　50 000 词书面语连续语音识别系统硬环境

句子与句子之间应略作停顿。在一个句子持续时间内，ADSP21020 在以 16kHz 频率进行采样(16bit 量化)的同时(间隔内)，将完成(上一帧的)预处理，即加窗、12 阶 LPC 分析、求 12 阶倒谱系数、矢量量化。在两个句子之间的停顿时间内，已矢量量化成码字(序号)串的句子，经过一个以离散 HMM 为核心，由 ADSP21020 汇编实现的识别模块，被识别成无调拼音形式的句子，再通过汉语四声识别部分和拼音-汉字转换单元(也由 ADSP 21020 汇编实现)，得到这个句子的汉字形式，识别结果通过双端口 RAM 送入 PC 机。在 ADSP21020 处理下一个句子的时间内，PC 机将这一句子识别结果写入文件中并显示在屏幕上。拼音-汉字转换单元的词汇表为从 1993 年全年的《人民日报》中得到的 50 000 词汇。系统识别所用句法也是从 1993 年全年《人民日报》

中统计生成的无调音节的 bigram 句法。

3.1.3 大型 HMM 音素识别

一般情况下,HMM 的状态数仅为 3～5 个,在孤立词识别中,HMM 被用来表示词(word),在连续语音识别中,尤其是大词汇量时,HMM 常被用来表示比词更小的语音单元,如音素以及考虑了上下文相关信息(context)的音素[102,132,136,137]。所谓大型 HMM,就是状态数多达几十个的 HMM。这方面的一个有代表性的工作是由 George 工学院的 Clements 教授和他的博士生们所提出的大型 HMM,其中,Farges 用大型 HMM 描述英语连续语音本身,通过 HMM 的状态序列来合成语音,实现了极低比特率的语音编码[72,73]。而 Pepper 则用大型 HMM 的相应状态序列作为音素识别系统的输入,即完成了预处理的工作[183],随后又研究了大型 HMM 的结构能分解成一组表示英语音素的子结构的方法[184]。大型 HMM 另一有影响的工作是由 Bell 实验室的 Levinson 等人提出的 CVDHMM(Continuously Variable Duration HMM)[142],这种 HMM 具有 43 个状态,每个状态对应于一个英语音素,且状态的观察值概密函数为高斯型,此外,假定状态驻留时间服从 Gamma 分布。这种 HMM 在英语音素识别中取得了和经典方法不相上下的识别率[143,154]。

音素识别是连续语音识别中的一个基本课题和重要基础。这里借鉴了 CVDHMM 的基本思想,在我们已经做的有关线性预测 HMM 的工作基础之上[237],提出了用于汉语音素识别的一个大型线性预测 HMM,一个状态对应于一个汉语音素,由于汉语有 22 个声母(包括零声母,这里记为@,35 个韵母,加上两个特殊韵母 er 和 #,其中 # 为表示无声时的韵母。因此,这个大型 HMM 一共有 59 个状态。而且,根据汉语语音中声母、韵母的统计特性,包括声母音长的特性[5,13],假定了各状态驻留时间服从其相应的某个区间内的均匀分布,避免了 Gamma 分布中出现驻留时间太长或

太短的可能性。下面首先给出大型线性预测 HMM 的参数估计方法以及用它来完成汉语音素识别的工作原理，然后再介绍一些实验结果[258]。

(1)大型线性预测 HMM

线性预测 HMM，已在 2.1.2 中介绍过。对于经典的线性预测 HMM 的参数估计，已有成熟的方法。但经典 HMM 的 Markov 链中状态 i 驻留时间 d 的分布 $p_i(d)$是一个指数分布，在很多情形下与真实情况不相符合，为此，人们提出了不少的改进措施，包括 CVDHMM 中对 $p_i(d)$服从 Gamma 分布的假定。此外，人们也做了 $p_i(d)$服从 Poisson 分布的尝试[207]，以及对在最大值 D 之内所有 $p_i(d)$，$1\leqslant d\leqslant D$，数值直接估计实验[113,192,198]。虽然所有非指数分布的改进都有助于性能的提高，但直接估计 $p_i(d)$的方法要求很多训练数据，而假定 $p_i(d)$服从 Gamma 分布或 Poisson 分布，虽对训练数据要求降低，但会出现 d 太小或太大的可能性，参见 2.2.2。

这里提出的大型线性预测 HMM，由于一个状态对应于一个汉语音素(声母或韵母)，因而具有 59 个状态，而且，适当考虑 $p_i(d)$也成了一个重要的问题。因为状态就是音素，状态驻留时间就是在语音中的音素持续时间，对汉语而言，声母持续时间有稳定的统计特性[5]。再利用 HMM/BSD(HMM with Bounded State Duration)的思想[84]，这里假定大型线性预测 HMM 每个状态对应的驻留时间是某一个相应区间内的均匀分布，由这个区间的起止点两个值描述。

那么，这里提出的大型线性预测 HMM 的参数集由下列各部分组成：初始状态概率矢量 $\pi=\{\pi_i,1\leqslant i\leqslant 59\}$，状态转移概率矩阵 $A=\{a_{ij},1\leqslant i,j\leqslant 59\}$，各状态驻留时间均匀分布上下限 $U=\{u_i,1\leqslant i\leqslant 59\}$，$L=\{l_i,1\leqslant i\leqslant 59\}$，描述各状态的观察值概密函数的参数，即 59 组 LPC 系数 $B=\{a_j,1\leqslant j\leqslant 59\}$或等价的 59 个自相关函数 $R=\{R_j,1\leqslant j\leqslant 59\}$，因为由 LPC 理论中的自相关法，可从自相

关函数中求取相应的 LPC 系数。

这里，大型线性预测 HMM(状态 1 到 22 为声母，状态 23 到 59 为韵母)的参数估计方法如下：

首先来看初始状态概率 $\pi=\{\pi_i,1\leqslant i\leqslant 59\}$的估计：在我们的系统中，每个连续语音的句子总是以“无声”开始和结束，而无声音节由零声母(状态 1)和无声韵母(状态 59)组成，因此，有

$$\pi_i=\begin{cases}1, & i=1\\ 0, & \text{其它}\end{cases} \tag{3-4}$$

而状态转移概率矩阵 $A=\{a_{ij},1\leqslant i,j\leqslant 59\}$可以估计如下：由于汉语音节的结构为声母-韵母结构，考虑状态和音素(声母、韵母)一一对应，因此，由声母转移到声母，韵母转移到韵母的概率为零，而由声母转移到韵母的概率，也就是汉语音节中声母和韵母结合的概率，已有资料给出了其统计值[14]，此外，由韵母转移到声母，意味着一个音节结束另一个音节开始，因此，其概率相当于汉语音节中声母出现的概率，这也有统计结果[14]，但由于这里考虑了无声韵母，它仅和零声母有关系，这样，就影响了零声母和其它韵母的结合概率。这里，利用实验中 500 个汉语连续语音句子中无声音节占有全部零声母音节的百分比，对零声母转移到各韵母的概率进行了修正。这样，就得到了状态转移概率矩阵 $A=\{a_{ij},1\leqslant i,j\leqslant 59\}$的全部参数。

接下来可以考虑状态驻留时间上下限的估计问题，即求取 $U=\{u_i,1\leqslant i\leqslant 59\}$和 $L=\{l_i,1\leqslant i\leqslant 59\}$。相应的 i 状态驻留时间分布为

$$\dot{p}_i(d)=\begin{cases}1/(u_i-l_i+1), & l_i\leqslant d\leqslant u_i\\ 0, & \text{其它}\end{cases} \tag{3-5}$$

由于已知声母持续时间的统计结果[5,13]，所以 $u_i,l_i(1\leqslant i\leqslant 22)$就得到了。必须指出，状态 1 对应的零声母，本来不占用时间。但为了处理方便，本文的系统中假定 $u_1=l_1=1$。所有 u_i 和 l_i 的单位是帧，这里，每帧语音长 10ms。对于状态 59 对应的无声韵母，类似于

零声母进行了处理。但是，对于状态 23 到 58 相应的韵母，u_i, l_i(23 $\leqslant i \leqslant$ 58)的求取就复杂一些：首先，根据人工分割成音素串的 144 个句子来选取 u_i, l_i(23$\leqslant i \leqslant$58)的初值，再用一个迭代过程，根据更多的语音数据(500 句)来改进初值。初值的选取方法为：对人工分割成音素串的 144 个句子，将其中各韵母的持续帧数记录下来，其最大最小值就是相应的 u_i 和 l_i 的初值。改进这些初值的迭代过程是和改进描述状态观察值概密函数的参数 $\boldsymbol{R}=\{R_j, 1\leqslant j\leqslant 59\}$ (与 $\boldsymbol{B}=\{a_j, 1\leqslant j\leqslant 59\}$等价)的初值的迭代一起进行的。而 $\boldsymbol{R}$ 的初值可选取如下：对人工分割成音素串的 144 个句子中各音素相应的各帧语音归一化自相关函数求和，就是该音素相应的自相关函数 R_j 的初值。这样，就有了这个大型线性预测 HMM 的全部参数，只不过，u_i, l_i(23$\leqslant i \leqslant$58)和 R_j(1$\leqslant j \leqslant$59)是根据人工分割成音素串的 144 个句子估计出来的初值，必须由一个迭代过程来改进它们，这个迭代过程可描述为：

对于 500 个连续语音句子组成的训练数据集，已知每个组成句子的音素串，只是不知道音素串中每个音素在给定的句子对应语音数据中的起止帧位置。那么，根据初始模型参数值，由简化的 Viterbi 算法(淡化了状态驻留时间分布的影响，实质上与一个 DTW 算法相同)可得到每一音素在句子中的起止帧位置，这样，可得到改进后的参数

$$\bar{u}_i = \max_l\{\text{状态 } i \text{ 在 } Q^{(l)} \text{ 中的最长持续时间}\}$$

$$23 \leqslant i \leqslant 58 \tag{3-6}$$

$$\bar{l}_i = \max_l\{\text{状态 } i \text{ 在 } Q^{(l)} \text{ 中的最短持续时间}\}$$

$$23 \leqslant i \leqslant 58 \tag{3-7}$$

$$\bar{R}_j = \sum_l \sum_{t=1}^{T_l} R_t^{(l)} \mid q_t^{(l)} = j$$

$$1 \leqslant j \leqslant 59 \tag{3-8}$$

其中，$Q^{(l)}$为第 l 个句子对应的状态序列。$R_t^{(l)}$ 为第 l 个句子中第 t 帧的归一化自相关函数。T_l 为第 l 个句子的帧数，$q_t^{(l)}$ 为 $Q^{(l)}$中第 t

个状态。改进后的大型线性预测 HMM 可再一次作为初始模型，重新开始新一轮迭代，直至得到满意的大型线性预测 HMM 参数。

用于汉语音素识别的过程为：对一个待识别的观察值序列 $O=O_1,O_2,\cdots,O_T$，由已训练好的大型线性预测 HMM 参数，根据修改后的 Viterbi 算法，可以求出一个最大似然意义上最优的状态序列 $Q=q_1,q_2,\cdots,q_T$，由于一个状态对应于一个汉语声母或韵母，那么，Q 序列实际上就是识别出的连续语音句子 O 对应的音素序列，即完成了音素识别的工作。所谓修改后的 Viterbi 算法，就是在经典 Viterbi 算法中考虑 Markov 链的状态驻留时间的概率分布 $p_i(d)$[84,154]，那么，初始值修正为

$$\delta_1(i)=\lg\pi_i+\lg p_i(1)+\lg b_i(O_1) \tag{3-9}$$

这里

$$\delta_1(i)=\begin{cases}\lg b_1(O_1), & i=1\\ -\infty, & \text{其它}\end{cases} \tag{3-10}$$

递归式修正为

$$\delta_t(j)=\max_{1\leqslant i\leqslant N}\max_{d<t}\Big[\delta_{t-d}(i)+\lg a_{ij}+\lg p_j(d)+\sum_{k=1}^{d}\lg b_j(O_{t-d+k})\Big]$$
$$2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{3-11}$$

而且

$$\psi_t(j)=(i,d)=\arg\max_{i,d}\Big[\delta_{t-d}(i)+\lg a_{ij}+\lg p_j(d)+\sum_{k=1}^{d}\lg b_j(O_{t-d+k})\Big]$$
$$2\leqslant t\leqslant T,1\leqslant j\leqslant N \tag{3-12}$$

因此，求取状态序列时，$\psi_t(j)$中的 d 值决定回溯的步长，i 值决定在此步长内状态序列的值。

(2)实验结果

实验是在与 3.1.2 中配有相同语音接口的一台 Compaq 486 计算机上完成的。采样频率为 10kHz，每帧 10ms，根据线性预测 HMM 要求，语音数据做了归一化处理，即各帧语音除以其相应 LPC 分析的残差能量 σ。训练用语音为 500 个连续语音句子。初步测试时，做音素识别的句子为从 500 句子中选出的 50 个句子。首先，估计出大型线性预测 HMM 的参数，再进行音素识别，实验初步结论如表 3.4 所示。表中的百分比亦是根据系统性能评估软件得到的。

表 3.4　50 句连续语音句子音素识别初步结果

系统	正确	消去错误	替代错误	插入错误
大型线性预测 HMM（一个音素一个状态）	44.8%	40.9%	14.3%	2.5%

初步实验结果表明，这里给出的大型线性预测 HMM，为解决难度很大的汉语连续语音音素识别问题提供了一个可供选择的途径。比起经典的一个音素建立一个 HMM 的识别系统（参见 3.1.2），本文的方法在初步实验中显示的识别率也略有提高（参见表 3.3）。尤其是训练好的模型占用存储空间少（这里是 17 228 byte，而经典的一个音素一个 HMM 的系统模型占用 91 686 byte），识别过程简便，主要涉及到 Viterbi 算法的计算，而 Viterbi 算法的快速并行算法及专用芯片已有很多可用的成果，因此，本文提出的方法对系统实用化和完成实时处理是有利的。

目前，将大型线性预测 HMM 用于汉语音素识别的工作正在做更多的实验，以便进一步改进系统性能。此外，利用大型线性预测 HMM 完成汉语连续语音音节识别和大词汇表时词的识别工作也已展开。

§3.2 语音增强

3.2.1 加性高斯白噪声中的语音增强方法

噪声，不仅影响到语音识别效果，而且，与语音压缩编码质量关系重大。因此，对有噪声语音的增强研究，一直是语音处理领域关注的一个重要课题。研究者们已经提出了不少方法和建立了很多实用系统[146]。经典语音增强方法大多是从统计信号处理的角度提出来的，例如，使用谱相减的方法[37]；对有噪语音建一个全极点模型以估计其参数的方法[145]；使用自适应梳状滤波器方法[147]；以及最近提出的采用最小均方误差短时谱估计的增强方法[66]等。神经网络技术也被应用于语音去噪增强处理，例如，基于著名的 BP 网络学习算示，建立一个四层神经网络来消除语音中的噪声的方法[217]。HMM 在语音增强方面亦得到了成功的应用，最有代表性的工作是由 Ephraim 等人做的[63,64,67,68,69]。这里只介绍作者提出的一种加性高斯白噪声中的语音增强方法[265]。

(1)问题

设 Y_t 为有噪声语音帧，S_t 为无噪声语音帧，n_t 为高斯白噪声帧，且有

$$Y_t = S_t + n_t, \quad t = 1, \cdots, T \tag{3-13}$$

要在最小均方误差条件下，从 Y_t 中估计出语音 $\hat{S}_t$。这是一个经典问题。显然，Y_t 彼此相关，即为有色的。如果将语音 S_t 作为白噪声通过一个状态变量模型表示的线性离散系统的输出序列，那么，增强之后的语音 $\hat{S}_t$ 的求取过程就是著名的 Kalman 滤波。这里将 S_t，$t=1,\cdots,T$，作为线性预测 HMM 输出的观察值序列，利用一种新颖的白化方法以及最小均方误差波形估计原理，导出了语音增强的一种新方法，亦可称为一种滤波方法。

(2)白化方法[241]

在信号检测和通信理论中，离散情况下高斯有色噪声白化问题一直为人们所重视。经典的方法是用协方差矩阵表示有色噪声的先验知识，然后，采用白化滤波器来使高斯有色噪声白化。但是，在绝大多数情况下，确定白化滤波器涉及到很多复杂方程求解，因而十分困难。

这里利用线性预测 HMM 表示高斯有色噪声先验知识，把高斯有色噪声序列作为线性预测 HMM 输出的观察值序列，从而使有色噪声的相关性隐含在线性预测 HMM 的 Markov 链中。据此，提出了一种高斯有色噪声白化新方法。这种方法也是后面的语音增强方法的基础。

首先，要获取噪声的先验知识。就是根据若干高斯有色噪声序列，估计出一个线性预测 HMM 的参数，这实际上是线性预测 HMM 训练问题，已有经典算法解决。

将高斯有色噪声序列记为 $n_c=n_{c1},n_{c2},\cdots,n_{cT}$，根据估计出的一组线性预测 HMM 参数，可采用下面所述方法进行白化处理：

作为线性预测 HMM 输出的观察值序列，n_c 可有很多不同的 Markov 状态序列相对应。但是，在最大似然的条件下，由 Viterbi 算法可以求出一个 Markov 状态序列，记为 $Q=q_1,q_2,\cdots,q_T$，显然，有

$$f(\boldsymbol{n}_{ct}/q_t=\theta_j)=(2\pi)^{-K/2}|\boldsymbol{C}_{nj}|^{-1/2}\exp\left[-\frac{1}{2}\boldsymbol{n}_{ct}^T\boldsymbol{C}_{nj}^{-1}\boldsymbol{n}_{ct}\right]$$

$$t=1,2,\cdots,T \tag{3-14}$$

其中，$\boldsymbol{C}_{nj}$为先验模型中描述 θ_j 状态对应之观察值概密函数 $b_j(\boldsymbol{X})$ 的参数，为一协方差矩阵。此时，$b_j(\boldsymbol{X})$采用如(2-23)式所示原始高斯型描述方式。

或者，写(3-14)式为下面形式：

$$f(\boldsymbol{n}_{ct}/q_t=\theta_j)=(2\pi)^{-K/2}\exp\left[-\frac{1}{2}\delta(\boldsymbol{n}_{ct};\boldsymbol{a}_{nj})\right] \tag{3-15a}$$

其中

$$\delta(\boldsymbol{n}_{ct};\boldsymbol{a}_{nj})=R_n(0)R_a(0)+2\sum_{i=1}^{p}R_n(i)R_a(i) \tag{3-15b}$$

且 n_{ct}为归一化的噪声帧。$R_n(i)$为 n_{ct}之自相关函数，$R_a(i)$为 a_{nj}之自相关函数，这时，对应于 $b_j(X)$采用如(2-31)式所示一般形式。a_{nj}为描述 $b_j(X)$的 LPC 系数。显然，a_{nj}和 C_{nj}同为描述 $b_j(X)$的参数，必定存在某种等价关系，在以后的讨论中将会涉及。

因此，根据 HMM 特性，有下式成立：

$$f(n_c/Q) = f(n_{c1}, n_{c2}, \cdots, n_{cT}/q_1, q_2, \cdots, q_T)$$
$$= f(n_{c1}/q_1) f(n_{c2}/q_2) \cdots f(n_{cT}/q_T) \qquad (3\text{-}16)$$

(3-16)式意味着，对有色噪声序列 $n_c = n_{c1}, \cdots, n_{cT}$，它的各个值 n_{ct}之间本是相关的。但是，在已知 Q 的情况下，n_{ct}之间彼此独立了，亦即白化了。此时，可以将 n_c 作为白色噪声处理，且各个值的统计特性由下式描述：

$$f_w(n_{ct}) = f(n_{ct}|q_t = \theta_j)$$
$$= (2\pi)^{-K/2} |C_{nj}|^{-1/2} \exp\left[-\frac{1}{2} n_{ct}^T C_{nj}^{-1} n_{ct}\right] \qquad (3\text{-}17)$$
$$t = 1, 2, \cdots, T$$

综上所述，对一个高斯有色噪声序列 n_c，在给定 Q 的条件下，可以把 n_c 作为高斯白噪声序列处理，且 n_{ct}的统计特性由(3-17)式表示。C_{nj}可以通过 a_{nj}计算出来，而 a_{nj}用经典线性预测 HMM 训练方法，在建立噪声先验知识模型时已经求出。

从前面的讨论中不难看出，一个高斯有色噪声序列 n_c，作为线性预测 HMM 输出之观察值序列，有很多不同的 Markov 状态序列对应，亦即 n_c 可能等价地看做很多个不同的白色噪声序列处理。但是，由 Viterbi 算法求出状态序列 Q 的情况下，将 n_c 等价地看做(3-17)式描述的白色噪声序列处理，从最大似然意义上讲，这是一个最佳的结果。

(3)最小均方误差波形估计原理[214]

设 $Z_t, t=1, \cdots, T$，为一高斯白噪声序列，即 Z_t 各值彼此独立，且有

$$Z_t = S_{wt} + n_t, \quad t = 1, \cdots, T \qquad (3\text{-}18)$$

其中，$S_{wt}, t=1,\cdots,T$，为白色的信号序列，$n_t, t=1,\cdots,T$ 为白色的噪声序列，那么，从有噪序列 Z_t 中估计出的信号 $\tilde{S}_{wt}$ 由下式表示：

$$\tilde{S}_{wt} = \sum_{k=1}^{T} H(t,k) Z_k, \quad t = 1,\cdots,T \tag{3-19}$$

权值 $H(t,k)$ 的选择，应使估计误差和数据集 $\{Z_t, t=1,\cdots,T\}$ 正交，即有

$$E\{[S_{wt} - \sum_{k=1}^{T} H(t,k) Z_k] Z_l^T\} = 0$$
$$t = 1,\cdots,T, l = 1,\cdots,T \tag{3-20}$$

或者，等价为

$$E[S_{wt} Z_l^T] = \sum_{k=1}^{T} H(t,k) E[Z_k Z_l^T]$$
$$t,l = 1,\cdots,T \tag{3-21}$$

因为 Z_t 为白色的，故有

$$E[Z_k Z_l^T] = D_z(k)\delta(k-l) \tag{3-22}$$

其中，$D_z(k)$ 为 Z_k 的方差，$\delta(\quad)$ 为冲击函数，且有

$$D_z(k) = D_{sw}(k) + D_n(k) \tag{3-23}$$

其中，$D_{sw}(k)$ 为 S_{wk} 的方差，$D_n(k)$ 为 n_k 的方差，又因为(3-18)式，因此

$$E[S_{wt} Z_l^T] = D_{sw}(l)\delta(l-t) \tag{3-24}$$

那么，(3-21)式变为

$$D_{sw}(l)\delta(l-t) = \sum_{k=1}^{T} H(t,k) D_z(k)\delta(k-l) \tag{3-25}$$

即有

$$H(t,l) = D_{sw}(l)\delta(l-t) D_z^{-1}(t) \tag{3-26}$$

因此，(3-19)式变为

$$\tilde{S}_{wt} = D_{sw}(t)(D_{sw}(t) + D_n(t))^{-1} Z_t$$
$$t = 1,\cdots,T \tag{3-27}$$

其中，$D_{sw}(t)$ 为 S_{wt} 的方差，$D_n(t)$ 为 n_t 的方差。$\tilde{S}_{wt}$ 即为最小均方误差估计值。

(4)语音增强算法

对于(3-13)式所示问题，将 $S_t, t=1,\cdots,T$ 作为线性预测 HMM 输出的观察值序列，意味着用线性预测 HMM 表示语音 S_t 的先验知识，那么，首先，要得到一个线性预测 HMM，即训练出一个模型，记为 λ，它表示了无噪声语音 $S_t, t=1,\cdots,T$。λ 的观察值概密函数可以表示如下：

$$b_j(\boldsymbol{X}) = (2\pi)^{-K/2}\exp\left[-\frac{1}{2}\delta(\boldsymbol{X};\boldsymbol{a}_j)\right] \tag{3-28a}$$

其中
$$\delta(\boldsymbol{X};\boldsymbol{a}_j) = R_x(0)R_a(0) + 2\sum_{i=1}^{p} R_x(i)R_a(i) \tag{3-28b}$$

这里，$\boldsymbol{X}$ 为归一化语音帧，$R_x(i)$为 $\boldsymbol{X}$ 的自相关函数，$R_a(i)$为 $\boldsymbol{a}_j$ 的自相关函数，且 $\boldsymbol{a}_j$ 为描述 $b_j(\boldsymbol{X})$的 LPC 系数，$\boldsymbol{a}_j, j=1,\cdots,N$，可用训练算法求出。

另一方面，(3-28)式也可用更为原始的方式表示：

$$b_j(\boldsymbol{X}) = (2\pi)^{-K/2}|\boldsymbol{C}_j|^{-1/2}\exp\left[-\frac{1}{2}\boldsymbol{X}^T\boldsymbol{C}_j^{-1}\boldsymbol{X}\right] \tag{3-29}$$

其中，$\boldsymbol{C}_j$ 为一个协方差矩阵，它描述了 $b_j(\boldsymbol{X})$，但此时，$\boldsymbol{X}$ 为没有归一化处理的语音帧。当然，$\boldsymbol{a}_j$ 和 $\boldsymbol{C}_j$ 存在着等价关系，后面将会看到，$\boldsymbol{C}_j$ 能够通过 $\boldsymbol{a}_j$ 求出来。

对高斯白噪声 $n_t, t=1,\cdots,T$，显然有

$$f(\boldsymbol{n}_t) = (2\pi)^{-K/2}|\boldsymbol{D}_n(t)|^{-1/2}\exp\left[-\frac{1}{2}\boldsymbol{n}_t^T\boldsymbol{D}_n^{-1}(t)\boldsymbol{n}_t\right] \tag{3-30}$$

其中，$\boldsymbol{D}_n(t)$为 $\boldsymbol{n}_t$ 之协方差矩阵，且只有对角线元素非零，即

$$\boldsymbol{D}_n(t) = (D_{ij})_{K\times K},\ D_{ii} = \sigma_n^2,\ i=1,\cdots,K,\ \text{其它 } D_{ij}=0.$$

有了上述预备知识之后，就可以给出在高斯白噪声条件下语音增强的一个新方法：

a. 选初值 $\tilde{\boldsymbol{S}}_t = \boldsymbol{Y}_t$，即近似认为 $\boldsymbol{Y}_t$ 也是由 λ 产生，因此，由 Viterbi 算法，可以求出一个相应于 $\tilde{\boldsymbol{S}}_t$ 的 Markov 状态序列 $Q=q_1, q_2,\cdots,q_T$。

b. 在已知 Q 的条件下，根据前面介绍的白化方法，可以把有色序列 $\tilde{\boldsymbol{S}}_t, t=1,\cdots,T$，作为白色序列等价处理，记为 $\tilde{\boldsymbol{S}}_{wt}$，且有

$$f(\tilde{S}_{wt}) = (2\pi)^{K/2}|D_{nw}(t)|^{-1/2}\exp\left[-\frac{1}{2}\tilde{S}_{wt}^{T}D_{nw}^{-1}(t)\tilde{S}_{wt}\right] \tag{3-31}$$

其中，$D_{nw}(t)$为$\tilde{S}_{wt}$的方差矩阵，且有

$$D_{nw}(t) = C_j|_{q_t=\theta_j} \tag{3-32}$$

C_j为(3-29)式所示之值。

c. 对于我们要解决的问题，此时变为从Y_t中估计出新的$\overline{S_{wt}}$，且

$$Y_t = \tilde{S}_{wt} + n_t, \qquad t = 1,\cdots,T \tag{3-33}$$

由最小均方误差估值原理结论(3-27)式，有

$$\begin{aligned}\overline{S_{wt}} &= D_{nw}(t)(D_{nw}(t) + D_n(t))^{-1}Y_t \\ &= C_j|_{q_t=\theta_j}\left[C_j|_{q_t=\theta_j} + D_n(t)\right]^{-1}Y_t \\ &\qquad t = 1,\cdots,T\end{aligned} \tag{3-34}$$

d. 此时$\overline{S_{wt}}$就是增强处理之后的语音。如果$\overline{S_{wt}}$效果还不满意，可以令$Y_t=\overline{S_{wt}}$转到a步，再一次增强。

(5)增强算法的实现

已提出的增强方法，如(3-34)式所示，要实现它，必须考虑下面几个问题：

a. 高斯白噪声产生和$D_n(t)$之计算

用计算机可以产生(0,1)均匀分布随机数序列，记为$r_1, r_2, \cdots$，用下式变换，可以得到服从$N(0,\sigma_n^2)$的高斯白噪声序列[15]

$$u_k = \sigma_n\sqrt{-2\ln r_k}\cos(2\pi r_{k+1}) \tag{3-35a}$$

$$u_{k+1} = \sigma_n\sqrt{-2\ln r_k}\sin(2\pi r_{k+1}) \tag{3-35b}$$

这是因为，由(3-35)式可以得到

$$r_k = \exp\left[-\frac{1}{2\sigma_n^2}(u_k^2 + u_{k+1}^2)\right] \tag{3-36a}$$

$$r_{k+1} = \frac{1}{2\pi}\left[\operatorname{arctg}\frac{u_{k-1}}{u_k} + c\right] \tag{3-36b}$$

那么，(u_k, u_{k-1})的概密函数为

$$g(u_k, u_{k-1}) = f(r_k, r_{k-1})|J| \tag{3-37}$$

其中，$|J|$为下面行列式值的绝对值

$$\begin{vmatrix} \frac{\partial r_k}{\partial u_k} & \frac{\partial r_k}{\partial u_{k-1}} \\ \frac{\partial r_{k-1}}{\partial u_k} & \frac{\partial r_{k-1}}{\partial u_{k-1}} \end{vmatrix}$$

易推出：

$$g(u_k, u_{k-1}) = \frac{1}{2\pi\sigma_u}\exp\left[-\frac{1}{2\sigma_u^2}u_k^2\right]\frac{1}{2\pi\sigma_u}\exp\left[-\frac{1}{2\sigma_u^2}u_{k-1}^2\right] \tag{3-38}$$

那么，$D_u(t)$为 $K\times K$ 阶对角线矩阵，且 $D_{ii}=\sigma_u^2$。

b. C_j 的计算

这是一个较复杂的问题。C_j 由(3-29)式定义，它是描述 X 没有归一化时的 $b_j(X)$的参数。而当 X 归一化时，描述 $b_j(X)$的参数为 LPC 系数 a_j。通常在线性预测 HMM 训练算法中，只求出 a_j，因此，下面推导从 a_j 计算出 C_j 的公式。

从(3-28b)式右边部分可得出

$$\begin{aligned}\delta(X; a_j) &= R_x(0)R_a(0) + 2\sum_{i=1}^{p} R_x(i)R_a(i) \\ &= R_a(0)\sum_{i=1}^{K} x_i^2 + 2\sum_{i=1}^{p} R_a(i)\sum_{l=1}^{K-i} x_l x_{l+i} \\ &\triangleq X^T C_j^* X\end{aligned} \tag{3-39}$$

其中

$$X^T = (x_1 x_2 \cdots x_K)$$

$$C_j^* = \begin{bmatrix} R_a(0) & R_a(1) & \cdots & R_a(p) & \cdots & 0 \\ R_a(1) & R_a(0) & \cdots & & & \vdots \\ \vdots & & & & & R_a(p) \\ R_a(p) & & & & & \vdots \\ \vdots & & & & & R_a(1) \\ 0 & \cdots & R_a(p) & \cdots & R_a(1) & R_a(0) \end{bmatrix} \tag{3-40}$$

比较(3-28a)式和(3-29)式，设 X 没有归一化时，描写 $b_j(X)$的平均预测残差能量为 σ_j^2，那么，显然有

$$C_j^{-1} = C_j^* / \sigma_j^2 \tag{3-41}$$

因此，有

$$C_j = \sigma_j^2 C_j^{*-1},\ j = 1, \cdots, N \tag{3-42}$$

其中 C_j^* 如(3-40)式所示。$R_a(i)$为 a_j 的自相关函数。(3-42)式中 σ_j^2 可以这样计算出来：线性预测 HMM 训练算法是采用自相关方法从自相关函数 $R_j(i)$求出 a_j 的。又因为用 LPC 理论中自相关方法求 a_j 的同时，可以求出预测残差能量 E_j，而 $\sigma_j^2 = E_j/K$，K 为帧长。但注意，这时处理的语音应是没有归一化的。至此，我们得到了从 a_j 计算出 C_j 的方法。实际编程时，利用 C_j^* 矩阵特点，可以找到很有效的求逆子程序[18]，而且(3-34)式中矩阵求逆也可类似处理。

(6)实验结果

我们利用(3-34)式所示的增强算法做了大量语音增强实验[237]，这里给出一个典型例子。

无噪语音为一短语“Good Morning”，波形如图 3.10(a)所示。波形幅度为整型数，区间为[−2 048，2 048]，信号平均能量为

$$\sigma_s^2 = 144\ 178.1 \tag{3-43}$$

对这段语音每帧作 LPC 分析($p=10$)，可以求出每帧的预测残差能量，再用样点数平均，记为 σ_s^{*2}，有

$$\sigma_s^{*2} = 2\ 319.319 \tag{3-44}$$

对应于图 2.1 模型的 σ，因为 $A(Z)$表示信号时变的特征，因此，σ 在某种意义上表示了信号的强弱。

无噪语音信号的先验知识，即其相应线性预测 HMM 训练问题，可用经典算法求取，但是，求自相关函数 $\overline{R}_j(i)$时，语音信号应是没有归一化的，而且在用 LPC 理论中自相关法求出描述 $b_j(X)$

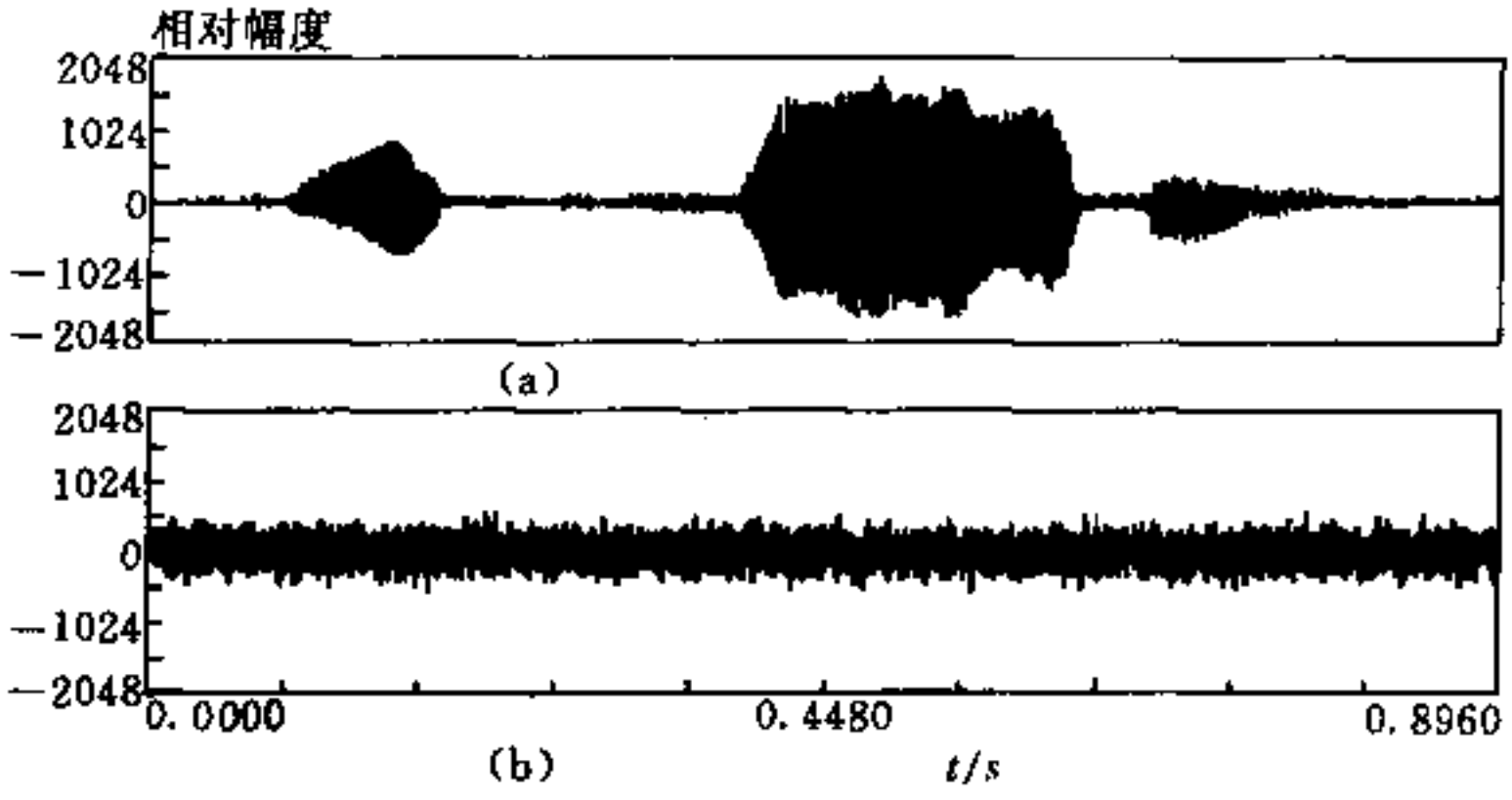


图 3.10　实验用信号和噪声波形

(a) 无噪语音"Good Morning"波形，$\sigma_s^2 = 144\ 178.1$；

(b) 高斯白噪声波形，$\sigma_n^2 = 23\ 193.19$

的参数 a_j 时，可同时求其平均预测残差能量，亦即 σ_j^2，然后，利用(3-42)式，求出模型参数另一形式 C_j，$j=1,\cdots,6$。

实验中使用的高斯白噪声序列，由(3-35)式描述。当方差 $\sigma_n^2 = 23\ 193.19$ 时，其波形如图 3.10(b)所示。实验中，对无噪语音叠加不同 σ_n^2 值的噪声，再用上节给出的增强方法处理，得到增强后的语音，从而可以验证方法的正确性。

首先考虑噪声较弱的情形，$\sigma_n^2 = 2\ 319.319$，那么，依照通常信噪比定义，去噪前的信噪比为：

$$\lambda_1 = 10\lg\frac{\sigma_s^2}{\sigma_n^2} = 10\lg\frac{144\ 178.1}{2\ 319.319} \approx 17.94\text{dB} \qquad (3\text{-}45)$$

叠加上噪声的语音波形如图 3.11(a)所示。去噪增强一次后的语音波形如图 3.11(b)所示。通过 D/A 转换、喇叭之后，听上去有明显改善。如果近似认为图 3.11(b)所示波形前三帧为噪声，信号保持不变，那么，去噪增强之后的信噪比为

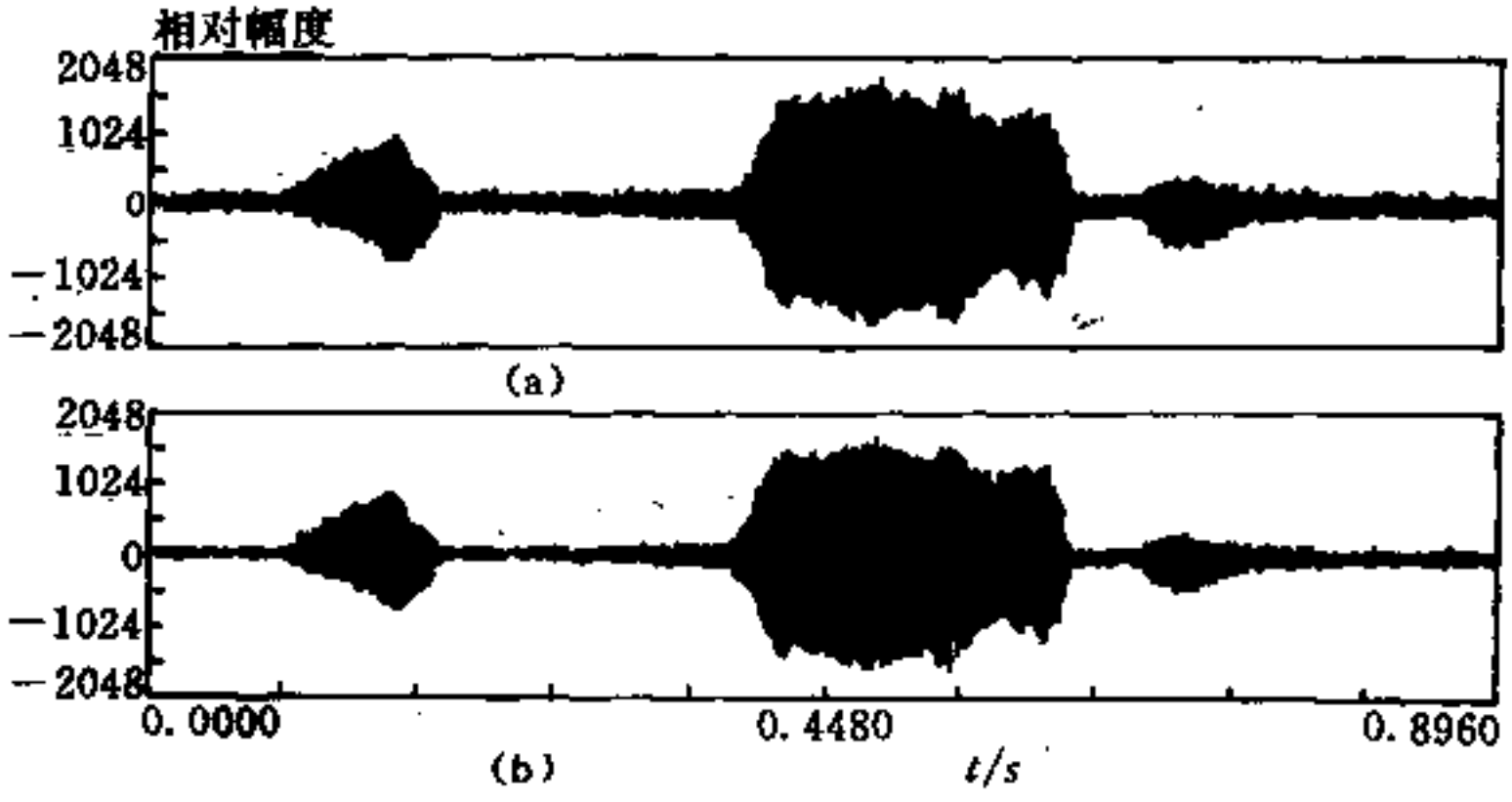


图 3.11　弱噪声时语音增强实验的波形

(a)有噪语音波形，$S/N=17.94$dB；

(b)增强后语音波形，$S/N=23.25$dB

$$\lambda_{s1}=10\lg\frac{144\ 178.1}{682.353}\approx 23.25\text{dB} \tag{3-46}$$

由此可见，信噪比改善了 5.31dB。

再考虑噪声较强的情形，$\sigma_n^2=23\ 193.19$，此时，去噪前的信噪比为

$$\lambda_{s2}=10\lg\frac{144\ 178.1}{23\ 193.19}\approx 7.94\text{dB} \tag{3-47}$$

叠加有噪声的波形，如图 3.12(a)所示。去噪增强迭代一次处理之后，波形如图 3.12(b)所示。将语音通过 D/A 转换、喇叭之后，听上去有很大的改善。类似的方法，可求出增强处理之后的信噪比为

$$\lambda_{s2}=10\lg\frac{144\ 178.1}{3\ 260.697}\approx 16.46\text{dB} \tag{3-48}$$

即信噪比改善了 8.52dB。

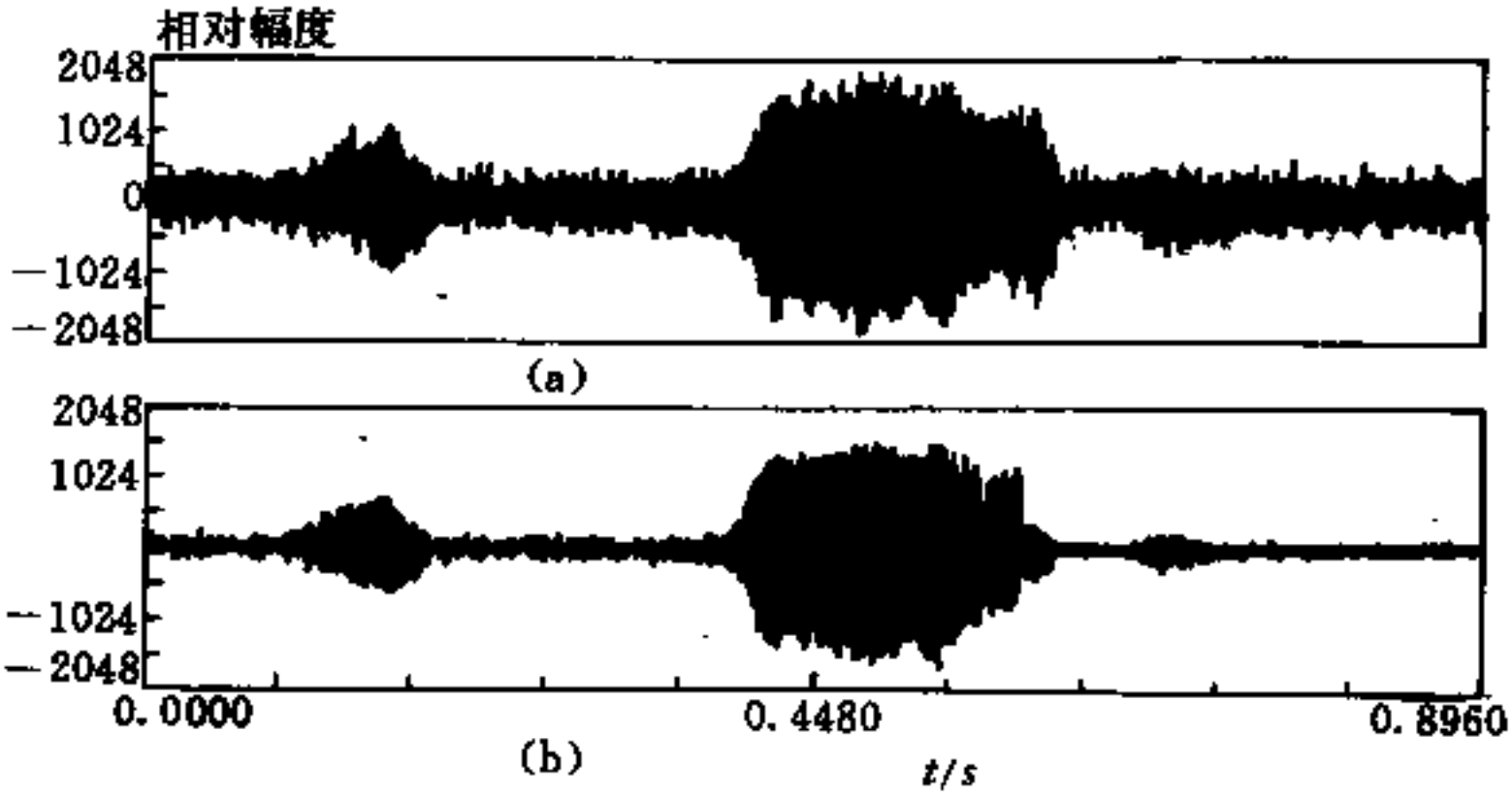


图 3.12　强噪声时语音增强实验的波形

(a) 有噪语音波形，$S/N=7.94$dB；

(b)增强后语音波形，$S/N=16.46$dB

从实验可知，这里给出的算法有很好的语音增强效果。但是，从本方法关键公式(3-34)式来看，由于涉及到矩阵求逆，因此，本方法在目前的计算机上处理时间较长。但是，本方法各帧处理是独立的和类似的，因此，一个有噪语音段每帧之增强处理可以同时并行处理，而经典去噪增强方法很多则不具备这一特点，在 VLSI 技术发展很快的今天，本方法的这个特点对硬件实现是很有益处的。

3.2.2　噪声环境下的语音处理

一般的语音处理的算法，都没有特别考虑到噪声的影响，因此，在噪声环境下，以这些算法为基础的系统的性能会急剧下降。对抗噪声的影响一般有三个途径：第一个途径就是首先对有噪声的语音进行增强处理，再送入一般的语音信号处理系统之中，就像

3.2.1 所讨论的语音增强方法那样。第二个途径是加强语音处理算法自身的自适应性和 Robust 性，使它们对有噪声和无噪声的环境都能适应。最后一个途径就是针对噪声特性，在语音处理算法中作些改进以消除噪声的影响。这里讨论后两种途径涉及的一些问题。

加强语音处理算法自身的抗噪能力，不仅可能，因为人类自身不必做复杂的适应就能在有噪声的环境中识别出语音，而且这种方法很有实际意义，因为不必知道噪声的特性。这方面的有代表性的工作是 Juang 等人从谱距离量度出发针对语音识别所做的工作[111,157]，其基本思想为：

设 c 为测试谱矢量(含有未知噪声)，η 为无噪声谱矢量，与一般欧氏距离不同，这里加了均衡因子 θ，那么，距离为

$$d(c,\eta) = (c - \theta\eta)^T(c - \theta\eta) \tag{3-49}$$

其中

$$\theta = \frac{c^T\eta}{|\eta|^2} \tag{3-50}$$

或者写成

$$\hat{d}(c,\eta) = |c|^2(1 - \cos^2\beta) \tag{3-51}$$

其中

$$\cos\beta = \frac{c^T\eta}{|c||\eta|} \tag{3-52}$$

根据这种距离量度可以构造出一种具有一阶均衡的 HMM。事实上，本书也几次提到 HMM 和距离量度的关系，参见(2-12)式。所谓一阶均衡 HMM，就是其观察值概密函数包含有均衡因子 θ，即有如下形式：

$$f(c) = k\exp\left[-\frac{1}{2}(c - \theta\eta)^T W(c - \theta\eta)\right] \tag{3-53}$$

这里，k 为归一化因子。类似(3-50)式，可推出均衡因子 θ 的表达式。

由 Baum-Welch 算法的重估公式思想，可以导出这种 HMM 相应的训练、识别算法。孤立词识别实验表明，这种方法能很好地对抗噪声的影响，相当于 SNR 提高 15～20dB。

改进语音处理算法，以便消除噪声的影响，也是一个很好的选择。例如，在识别过程中采用一个由高斯型 HMM 构成的滤波器[34]；对背景语音和待识别语音分别建立不同形式的 HMM 的方法[223]；克服汽车内噪声的方法[155]等等。这里给出一种噪声环境下估计线性预测 HMM 参数的不同于经典 HMM 训练算法的方法[246]。

这里讨论的有噪语音，可用(3-13)式表示，亦即有

$$\boldsymbol{Y}_t = \boldsymbol{S}_t + \boldsymbol{n}_t, \quad t = 1, \cdots, T \tag{3-54}$$

其中，$\boldsymbol{Y}_t$ 为有噪语音，$\boldsymbol{S}_t$ 为无噪语音，$\boldsymbol{n}_t$ 为高斯白噪声，且记 $\boldsymbol{Y}_t = (y_{t1}, \cdots, y_{tK})^T$，$\boldsymbol{S}_t = (s_{t1}, \cdots, s_{tK})^T$，$\boldsymbol{n}_t = (n_{t1}, \cdots, n_{tK})^T$，$K$ 为每帧长度。

基于 HMM 的所有语音处理，都是分帧进行的。亦即在一帧之内，视语音为平稳的，那么，有

$$\begin{aligned} R_y(\tau) &= R_s(\tau) + R_n(\tau) \\ &= \begin{cases} R_s(0) + \sigma_n^2, & \tau = 0 \\ R_s(\tau), & \tau \neq 0 \end{cases} \end{aligned} \tag{3-55}$$

由此可见，噪声对每帧相关函数的影响，表现在 $\tau=0$ 时增加了一个 σ_n^2 值，那么，在求取有噪语音的线性预测 HMM 参数时，设法在每帧自相关函数 $\tau=0$ 时减去 σ_n^2 值，即可获取消除了噪声影响的参数。

线性预测 HMM 的参数，由两部分组成：描述 Markov 链的参数以及描述各状态对应之观察值概密函数 $b_j(\boldsymbol{X})$的参数 a_j 或等价的自相关函数 $R_j(i)$，$j=1, \cdots, N$。那么，对有噪语音的线性预测 HMM 参数估计问题，可这样解决：

描述 Markov 链的参数仍由经典算法求取，因为这些参数对整个线性预测 HMM 影响不是太大，这样处理是可以接受的。而 $R_j(i)$可用下面对(2-36)式修正的公式求取：

$$\bar{R}_j(i)=\begin{cases}\dfrac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)[R_t^{(l)}(0)-\sigma_n^2]}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)}, & i=0\\ \dfrac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)R_t^{(l)}(i)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{*(l)}(j)\beta_t^{*(l)}(j)}, & i\neq 0\end{cases} \tag{3-56}$$

实验中选取的有噪语音，如图 3.11(a)所示。这是图 3.10(a)所示无噪语音“Good Morning”叠加上 $\sigma_n^2=2\ 319.319$ 的高斯白噪声而成。实验是这样进行的：无噪语音，记为 O，用经典方法求出一个线性预测 HMM，记为 λ，有噪语音，记为 O_n，用经典方法也求出一个线性预测 HMM，记为 λ_n，同时，再用经典方法和上面的(3-56)式，求出一个线性预测 HMM，记为 λ_r，可以计算出：

$$\lg P(O/\lambda)=-15.671\ 03 \tag{3-57a}$$

$$\lg P(O/\lambda_n)=-32.352\ 87 \tag{3-57b}$$

$$\lg P(O/\lambda_r)=-25.824\ 82 \tag{3-57c}$$

上面这些数值，表示一种相对的概率密度函数值，单个数值本身没什么意义，只有在相互比较时，才具有一种类似概率的意义，表示用模型表征相应语音的可靠程度。为方便计，下面仍称其为“概率值”。根据 HMM 距离量度思想[112]，我们来看这三个模型的区别所在：

由(3-57)式可知，由无噪声语音 O 的真实模型 λ 产生 O 的概率对数值为 $-15.671\ 03$，而由有噪声语音 O_n 训练的模型 λ_n 产生 O 的概率对数值为 $-32.352\ 87$，说明了由 λ_n 描述 O 产生了不少失真，但经过去噪处理算法(3-56)求取的模型 λ_r 产生 O 的概率对数值为 $-25.824\ 84$，说明了由 λ_r 描述 O 比用 λ_n 描述 O 精确得多，表明了这里提出的有噪语音线性预测 HMM 参数估计方法有明显的抑制噪声影响的作用。

现在具体看一下噪声被抑制的程度。对图 3.11(b)所示的用去噪增强方法处理过的语音，记为 O_w，用经典方法可求出一个模型，记为 λ_w，易求出：

$$\lg P(O/\lambda_w) = -22.036\ 49 \tag{3-58}$$

由此可见，$\lg P(O/\lambda_r)$和 $\lg P(O/\lambda_w)$数值上接近，这说明了用这里的方法得到的线性预测 HMM，即 $\hat{\lambda}_r$，在对噪声抑制程度上，接近于先将有噪语音增强处理一次，再用普通方法求其线性预测 HMM 参数。这再一次证实了这里提出的有噪语音线性预测 HMM 参数估计方法是很有效的。实际应用修正公式时，σ_n^2 可从无语音段统计出来。

应该指出，对于噪声环境下的 HMM 参数估计问题，直接对训练算法加以修正，消除噪声的影响，比起先将有噪语音增强，再用经典方法训练要合理得多，不仅因为前者更省时和方便，而且因为我们从有噪语音中最终获取的是 HMM 参数而不是无噪声的语音。

§3.3 语音压缩

3.3.1 语音特征参数数据的压缩实验

语音特征参数，在语音识别和编码中有广泛应用[4,196,256]。特征参数，不仅指一般的时域特征、频域特征、LPC 系数等，还包括基音周期和共振峰[249,260]，LSP 系数[262]等等。基于语音特征参数的语音分析-综合系统，是低比特率语音编码的核心。目前成功的低比特率语音编码方案有多脉冲预测编码(MP-LPC)[21]，自适应预测编码(APC)[22,49]，码激励预测编码(CELP)[210]等等，它们的共同基础是根据语音产生模型，如图 3.13 所示，而得到的 LPC 分析-综合系统，亦称为 LPC 声码器，如图 3.14 所示。这里讨论根据线性预测 HMM 的特性而进行的对图 3.14 中语音特征参数数据

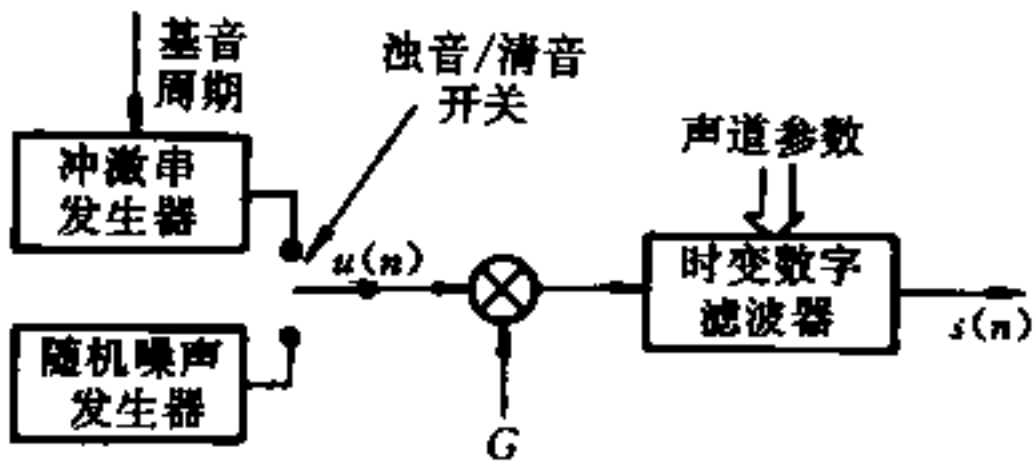


图 3.13　语音产生模型示意图

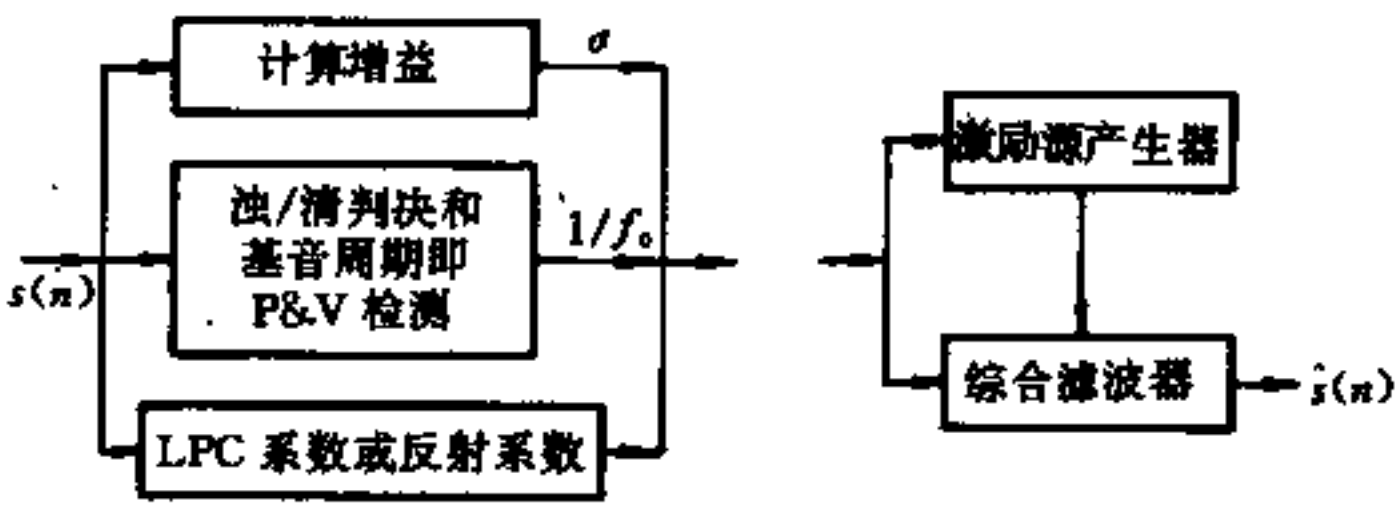


图 3.14　LPC 分析-综合系统(声码器)框图

(LPC 或反射系数)进行压缩的实验[263]。

(1)基本思想[252]

对一段语音 $O=O_1,O_2,\cdots,O_T$,若用每帧的 LPC 系数描述其特征,需要 T 组不同的 LPC 系数,这里探讨利用线性预测 HMM 特性,只需要更少的 LPC 系数就可以表征 O 的方法。

下面分析 Viterbi 得分的计算。所谓 Viterbi 得分,就是用 Viterbi 算法计算出的 $P(O/\lambda)$值。由于线性预测 HMM 的观察值概密函数可用(2-9)式表示,因常数大小没有影响,因此,简化(2-9)式为下式:

$$b_j(\boldsymbol{X}) = \exp[-D(\boldsymbol{X};\boldsymbol{a}_j)] \tag{3-59a}$$

$$\lg b_j(\boldsymbol{X}) = -D(\boldsymbol{X};\boldsymbol{a}_j) \tag{3-59b}$$

其中,$D(\boldsymbol{X};\boldsymbol{a}_j)$表示语音帧 $\boldsymbol{X}$ 的 LPC 系数 $\boldsymbol{a}_z$ 和描述 $b_j(\boldsymbol{X})$的参数 $\boldsymbol{a}_j$ 之间的 Itakura 距离。若线性预测 HMM 选用图 1.5(d)所示 Markov 链,那么,Viterbi 算法可以重新简写如下:

a. 初始化：　$\delta_1(1)=\lg b_1(\boldsymbol{O}_1)=-D(\boldsymbol{O}_1;\boldsymbol{a}_1)$　(3-60a)

b. 递归：　$\delta_t(j)=\max\begin{bmatrix}\delta_{t-1}(j-1)+\lg a_{j-1j}-D(\boldsymbol{O}_t;\boldsymbol{a}_j)\\ \delta_{t-1}(j)+\lg a_{jj}-D(\boldsymbol{O}_t;\boldsymbol{a}_j)\end{bmatrix}$　(3-60b)

c. 终结：　$P=\lg P(O/\lambda)=\delta_T(N)$　(3-60c)

比较 DTW 典型实现算法[194,209]可以得到一个结论：

如果将状态转移概率 a_{jj} 的影响，解释为 DTW 路径的加权，那么，Viterbi 得分值 $\lg P(O/\lambda)$ 等价于由各帧 LPC 系数表征的 T 帧语音 O 的模板（记为 P_0）和一个 N 帧模板 $a_1,a_2,\cdots,a_N$（记为 P_*）之间的 DTW 距离（负值）。

这样，当用经典方法训练线性预测 HMM 时，从 O 得到 λ 的过程可以解释为：

a. 选择初始模型 λ_0，此时，若记描述 λ_0 的 N 个观察值概密函数的 LPC 系数为 $a_1^{(0)},\cdots,a_N^{(0)}$，那么，$P(O/\lambda_0)$ 的对数值的负值大小，表示了将 T 帧 LPC 系数表征的语音 O 压缩成 N 帧 LPC 系数 $a_1^{(0)},\cdots,a_N^{(0)}$ 表示的语音（记为 M_0）时，O 和 M_0 的 DTW 距离，记为 d_0。

b. 用重估（reestimation）公式，得到新的模型 λ_1，当然，也得到了一组新的参数 $a_1^{(1)},\cdots,a_N^{(1)}$，此时，$P(O/\lambda_1)$ 的对数值负值大小，表示了将 O 压缩为 M_1 时，O 和 M_1 的 DTW 距离，记为 d_1，因为 $P(O/\lambda_1)>P(O/\lambda_0)$，因此 $d_1<d_0$，或者是说，此时得到的压缩语音 M_1 比 M_0 更准确地表示了原来语音 O。

c. 当一直持续上述过程，最后由 O 得到了 λ，此时，$P(O/\lambda)$ 达到最大值，描述 λ 的观察值概密函数的 N 组 LPC 系数，记为 $a_1,\cdots,a_N$，设其表征了某段语音 M，那么，将 T 帧语音 O 压缩成 N 帧 LPC 系数表示的 M，得到了最佳的结果，因为 O 和 M 的 DTW 距离最小。

因此，可以得到结论：线性预测 HMM 用经典方法训练的过程，是一个语音（特征）压缩的过程，训练中重估公式每次迭代，既

是为了得到一个更好的表达训练数据的模型，也是为了得到更好的压缩后的语音(特征)。通常的研究，只注意到模型，并未考虑这个压缩后的语音(特征)。

虽然确定了线性预测 HMM 的压缩作用，但将其用于实际语音压缩，还面临一个困难：上述对 T 帧语音 O 压缩后的语音只有 N 帧，具体说来，是只有 N 帧 LPC 系数表示的一个"虚拟"的语音。下面采用一个简单处理方法，得到了用 N 帧不同 LPC 系数描述的，仍然是 T 帧长的语音特征序列，从而使线性预测 HMM 用于实际语音压缩成为可能。

当用 O 训练出 λ，得到一组最佳参数 $a_1,\cdots,a_N$ 之后，用 Viterbi 算法，可以得到一个 Markov 状态序列 $Q=q_1,q_2,\cdots,q_T$，显然，$q_t\in(\theta_1,\cdots,\theta_N)$，那么，据此可构造一个特征序列：

$$F=F_1,F_2,\cdots,F_T,\qquad \text{其中},F_t=a_j|q_t=\theta_j$$

即当 t 时刻的状态 q_t 为 θ_j 时，令 F_t 为 θ_j 对应之 $b_j(X)$的参数 a_j。因此，用 F 描述 O 时，虽然 F 和 O 一样也是 T 帧，但 F 只包含有 N 组不同 LPC 系数。比直接用 T 帧 LPC 系数描述 O 就需要少得多的特征参数。又由于 Viterbi 算法得到的 Markov 状态序列 Q 是一个最佳结果，因此，从这个意义讲，由此构造的 F 特征序列也是最佳的。

这样，可以得到一个对 LPC 分析-综合系统的修正方案，如图 3.15 所示。这个方案比 LPC 声码器所需分析数据中语音特征参数要少 T/N 倍，因而可以实现比它更低的数码率或更高的压缩比，下面用实验对其进一步考查。

(2)实验系统组成与实现

实验系统，既要对我们提出的新的语音分析-综合方案(图 3.15)的可行性给出证明，又要将其与原来的 LPC 分析-综合系统(图 3.14)的性能进行比较，因此，实验系统的组成如图 3.16 所示。

$s(n)$为一个语音段，一般不超过 1 秒。对实际连续语音，可以

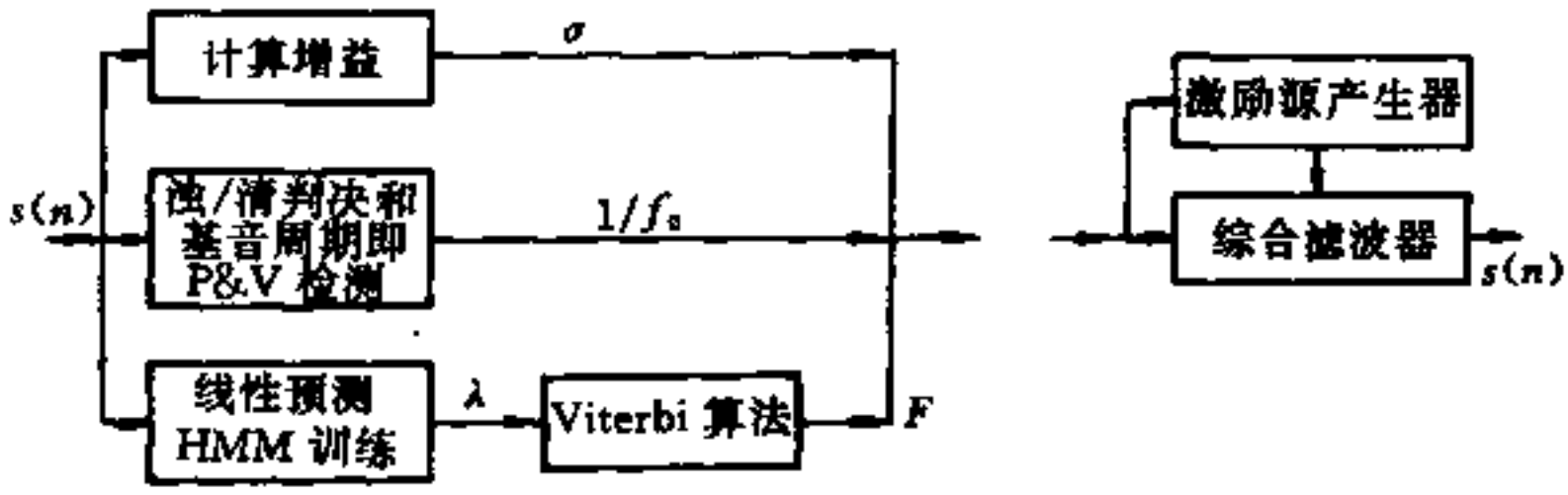


图 3.15　LPC+线性预测 HMM 分析-综合系统框图

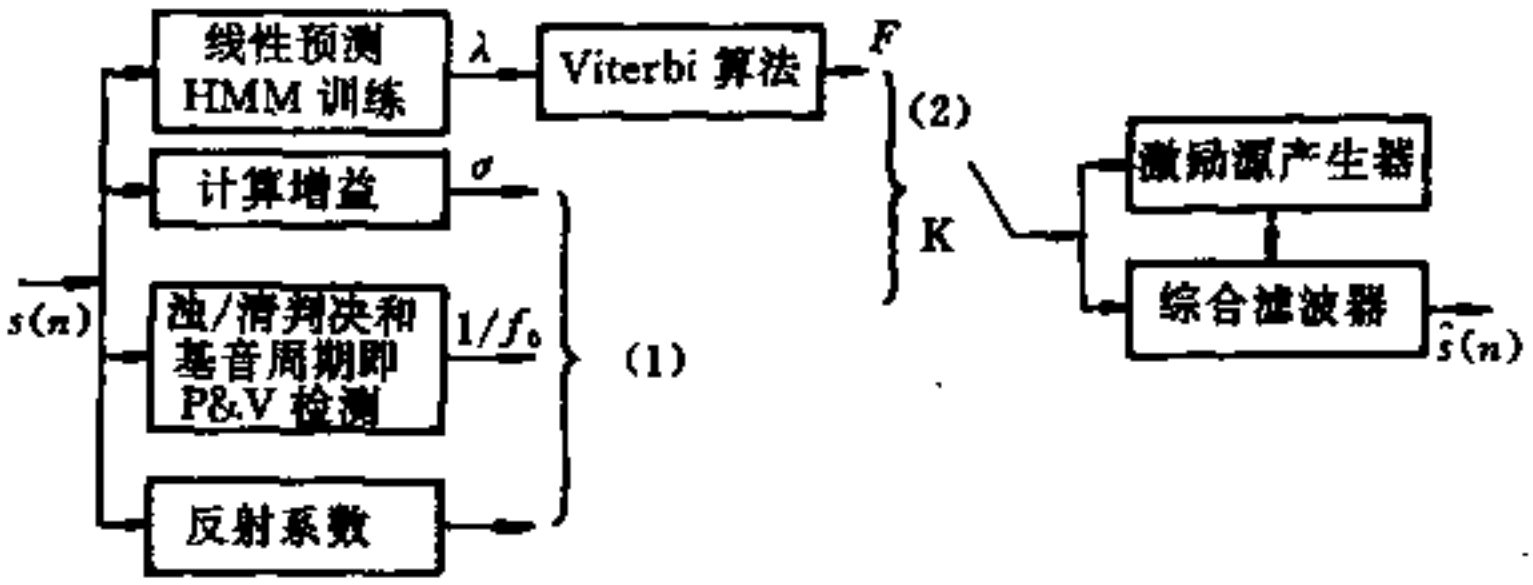


图 3.16　实验系统的组成

注：(1) LPC 分析-综合系统，

(2) LPC+线性预测 HMM 分析-综合系统

分段处理。这里只讨论一个语音段的情况，记为 $O=O_1,O_2,\cdots,O_T$，共有 T 帧。语音特征参数选用了与 LPC 系数等价的各帧的反射系数，这样做的理由是：可以利用我们已有的世界标准信号处理软件包 ILS(Interactive Laboratory System)，使实验系统的实现更方便些，而且，一般实际的 LPC 声码器也是采用反射系数。那么，图中构造的 F 序列，就由相应的 N 组不同的反射系数组成，这一点是容易实现的，因为线性预测 HMM 训练算法，描述 $b_j(X)$的参数 a_j 是根据其自相关函数 $R_j(i)$计算出来的。事实上，用 LPC 理论的自相关法从 $R_j(i)$求取 a_j 的同时，反射系数 ρ_j 作为中间结果之一也自然求出了。因此，当用 Viterbi 算法求出状态序列时，即得到 $Q=q_1,q_2,\cdots,q_T$ 之后，可以构造 F 序列 $F=F_1,F_2,\cdots,F_T$ 如

下：

$$F_i = \rho_j|_{\sigma_i=\sigma_j} \tag{3-61}$$

如图 3.16 所示实验系统的大部分，是直接借助 ILS 软件包来实现的[219]。其中，实现语音 LPC 分析部分的工作过程大致为：语音经过预处理之后，一小段语音（大约 2 个普通帧）放入 P&V 缓存帧中用来作 P&V 检测。再从这一小段语音中抽取一普通帧长的语音放入分析缓存帧中。之所以这样做，是为了使 P&V 检测结果更为准确。当连续两个普通帧长的语音都是浊音或清音时，P&V 缓存帧大小就变为分析缓存帧、即普通帧大小，并且相对分析帧作左右移动，当浊音出现时，P&V 缓存帧时间上延迟，即移到右边，否则，时间上超前，移到左边，放入分析帧的语音被预加重、加汉明窗，再用自相关法求其反射系数，同时，LPC 系数、残差增益也被求出。因为高质量语音合成的关键是 P&V 检测结果的准确与否，因此，ILS 软件包采用了一个基于倒谱处理的极复杂的 P&V 检测方法——统计浊/清鉴别器。

实验系统中综合部分的实现框图如图 3.17 所示。这样，如图 3.16 所示实验系统的实现问题就解决了。当开关 K 指向(1)时，实验系统就是 LPC 分析-综合系统，它的分析部分和综合部分的实

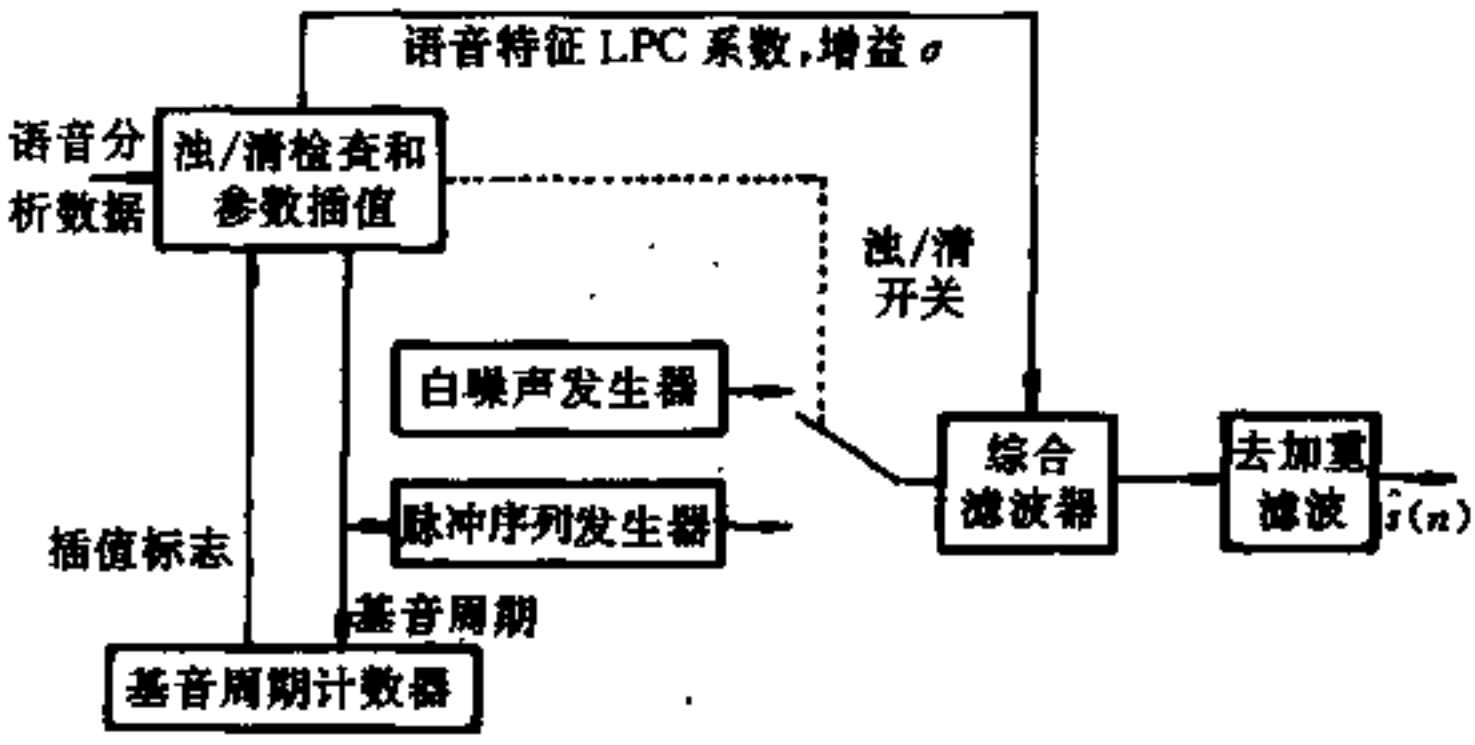


图 3.17　实验系统综合部分实现框图

现完全由 ILS 软件包完成。当开关指向(2)时，实验系统变为 LPC＋线性预测 HMM 分析-综合系统。分析部分的实现，可用 F 序列和 ILS 软件包产生的增益、基音周期和清/浊音判决结果一起作为语音分析数据。综合部分仍由图 3.17 所示框图来实现。

(3)实验结果

在图 3.16 所示实验系统上，我们进行了大量语音特征参数数据压缩实验[237]。下面给出一组典型的关于词和短语的语音分析-综合实验结果(线性预测 HMM 状态数为 6，即 $N=6$)。

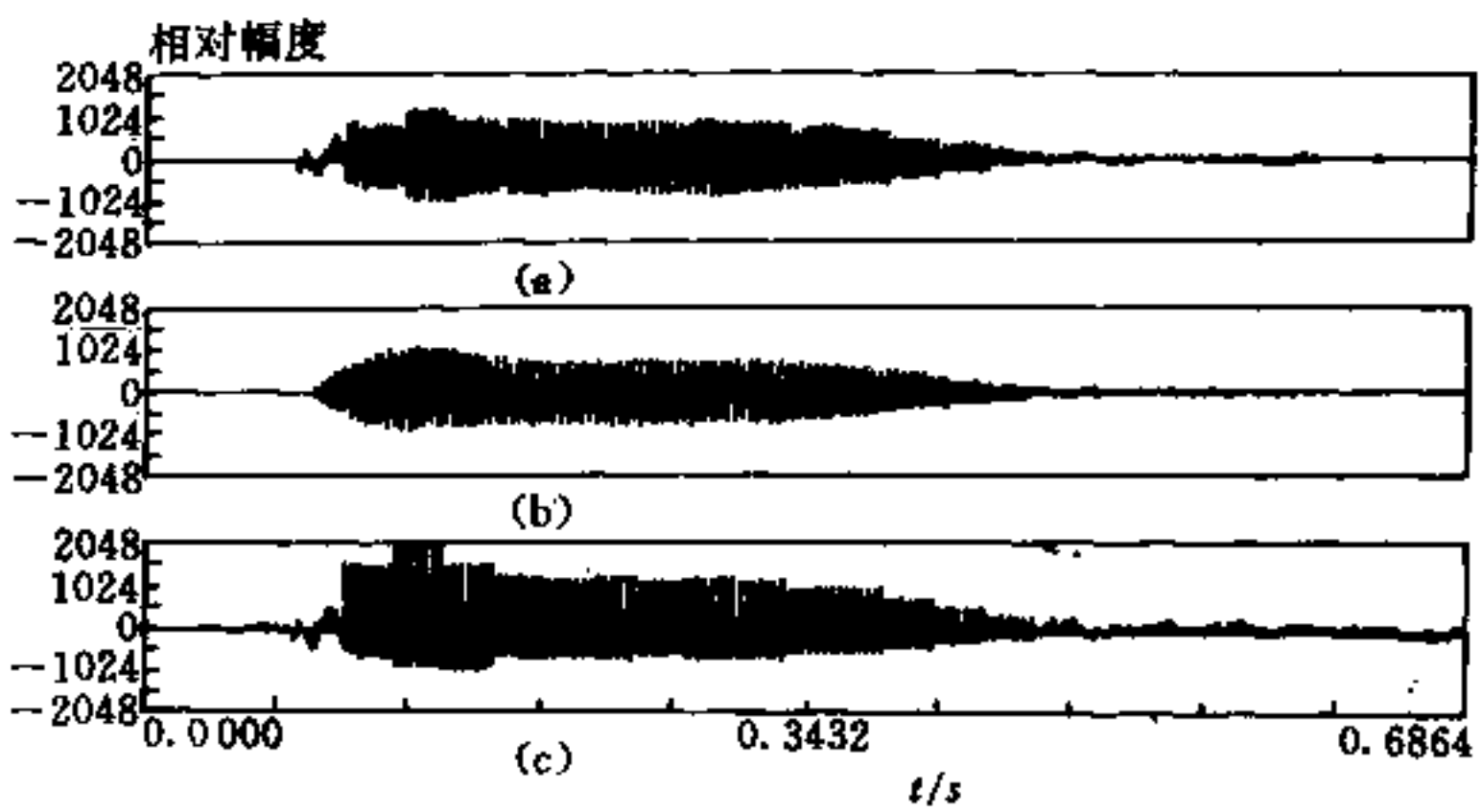


图 3.18　语音“8”实验结果

(a)LPC 分析-综合系统综合出“8”的波形；

(b)语音“8”原始波形；

(c)LPC＋线性预测 HMM 分析-综合系统综合出“8”的波形

图 3.18 表示孤立词语音“8”的实验结果波形。图 3.19 表示短语“Good Morning”的实验结果波形。实验时，对每个波形表示的语音都通过 D/A、喇叭，仔细地用耳朵辨听它们的差异，结论是：用 LPC 分析-综合系统和 LPC＋线性预测 HMM 分析-综合系统重建的语音，从波形上看有些差别，但人耳感觉不出它们彼此的区

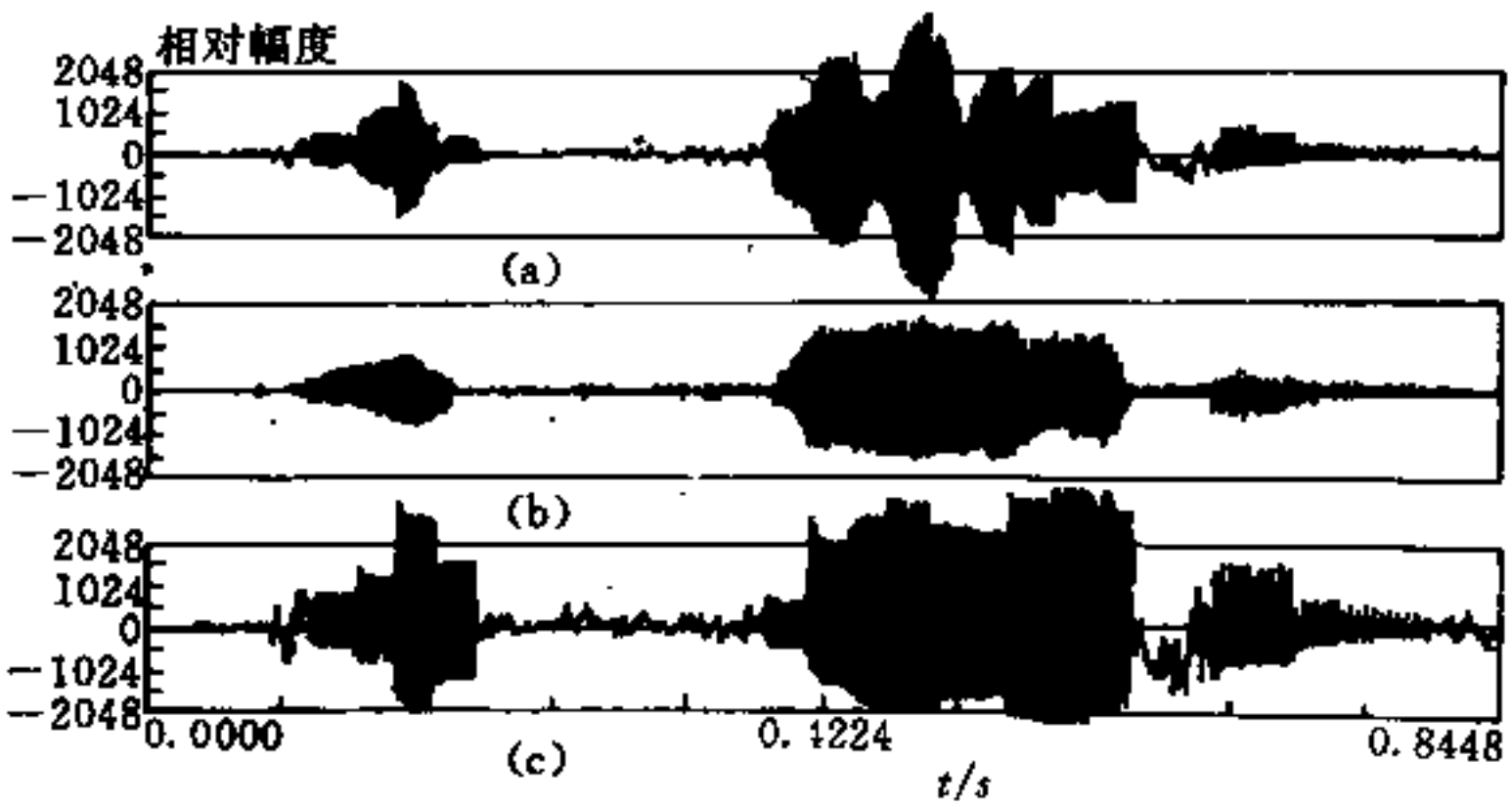


图 3.19　语音"Good Morning"实验结果

(a)LPC 分析-综合系统综合出"Good Morning"的波形；

(b)语音"Good Morning"原始波形；

(c)LPC＋线性预测 HMM 分析-综合系统综合出"Good Morning"的波形

别。注意到 LPC+线性预测 HMM 系统综合语音时所用分析数据中的语音特征序列中仅含有 N 个($N=6$)不同的反射系数特征，而 LPC 分析-综合系统的相应特征序列却包含 T 个($T\geqslant 26$)不同的反射系数特征，因此，可以得到结论：这里利用线性预测 HMM 特性实现的对所描述语音的特征参数进行压缩的方法是有效的和可行的。它可以在增加压缩比的情况下，几乎不降低语音质量。但是，有必要指出的是，F 序列，即只含 N 个不同特征参数的序列的求取时间，远远大于求取语音每帧特征参数，即求取一个含有 T 个不同特征参数的序列所花的时间。这就是线性预测 HMM 压缩方法提高了压缩比所付出的代价，换句话说，在语音质量几乎不降低的情况下，LPC+线性预测 HMM 语音分析-综合系统以增加终端处理时间为代价，换取了压缩比的提高或者说数码率的降低。下

面近似地作一定量分析。

根据文献[179,198]的分析，一个典型的 LPC 声码器的数码率可以确定如下：每帧语音的 $p=10$ 阶反射系数比特数分配为：$K_1 \sim K_4$，每个系数 5bit，$K_5 \sim K_8$，每个系数 4bit，K_9 为 3bit，K_{10}为 2bit，共计 41bit 传输反射系数。另外，7bit 传输基音周期和清/浊音判决，5bit 传输增益，1bit 用于同步，这样，每帧语音分析数据总计需要 54bit。在我们的实验中，每帧为 25.6ms，这样，LPC 声码器典型数码率为：2.1kb/s。对于图 3.18(b)的长 26 帧的语音波形，LPC＋线性预测 HMM 语音分析-综合系统的数码率可以确定如下：传输 6 组不同的反射系数所需比特数为：6×41＝246bit，传输 26 帧基音周期、清/浊音判决、增益所需比特数为：26×(7＋5)＝312bit，考虑到同步要求提高，设为：26×3＝78bit，那么，数码率为：(246＋312＋78)bit/0.6864s＝926.6bit/s。对图 3.19(b)所示语音波形，LPC＋线性预测 HMM 语音分析-综合系统数码率为：859.4bit/s。

综上所述，可以得到结论：如果 LPC 声码器的数码率为典型值 2.1kb/s，引入线性预测 HMM 之后建立的 LPC＋线性预测 HMM 语音分析-综合系统数码率可低至接近 1kb/s，即大约为 LPC 声码器数码率的一半。两个系统重建的语音质量几乎没差别。这充分说明了，利用线性预测 HMM 特性，对语音特征参数进行压缩，可以极为有效地降低传输时的数码率。而且，对语音质量并无明显影响，但要付出处理时间延长的代价。

3.3.2 经典矢量量化方法的改进

矢量量化(Vector Quantization，VQ)，作为一种高效的信源编码技术，在语音处理中获得了广泛的应用[83,156]，在 Buzo 等人成功地构造出第一个实用矢量量化器之后[147,148]，人们对矢量量化的实现方法进行了大量研究。80 年代中期以来，随着神经网络的研究热潮的兴起，用 Kohonen 提出的自组织特征映射(Self-

Organizing Feature Map，SOFM)来实现矢量量化，受到了广泛的重视[122,150]。

SOFM 结构是一个两层网络，每一个输入节点都通过一个具有可变权值的连接与每一个输出节点相连。SOFM 的矢量量化器是通过修正连接输入节点与输出节点之间的权值来实现的。输出节点按二维网格方式排列，随着输入矢量依次输入网络，节点之间的连接权不断地被修正，当网络接收到足够的输入矢量之后，各连接权就自动地形成各个相应的矢量中心，并使各矢量中心的点密度近似于输入矢量的概密函数，而且，拓扑位置上相近的节点对物理量相似的输入是敏感的。每个输出节点都有一个邻域，随着训练时间增大，邻域逐渐缩小，从而产生出所要的码本。

这里利用 HMM 特性，对经典 SOFM 矢量量化器进行了改进，改进分码本的产生和量化两方面，由此给出了一种实现矢量量化的新方法——混合 SOFM/HMM 方法[257]。这个方法不同于从信息论出发，结合 HMM 思想，对 SOFM 所作的改进[204]。

(1)方法

基本思想是，根据 SOFM 方法产生的码本，构造相应的 HMM，从而可以使用 HMM 的各种算法对码本进行改进处理，达到更好的码本训练和量化的效果。

设经典 SOFM 矢量量化方法产生的码本中的码矢为 $C_1, C_2, \cdots, C_M$。这实质上为 M 个(帧)语音倒谱系数特征，训练集记为 $O^{(l)}, l=1,\cdots,L$，其中 $O^{(l)}$ 表示第 l 个句子的各帧倒谱系数，即 $O^{(l)}=\boldsymbol{O}_1^{(l)},\cdots,\boldsymbol{O}_{T_l}^{(l)}$，其中 T_l 为第 l 个句子的帧数，$\boldsymbol{O}_t^{(l)}$ 为第 l 个句子的第 t 帧语音的倒谱系数。由已知的码本可以构造一个与 HMM 相应的 M 个观察值概密函数[247]，记为

$$b_j(\boldsymbol{X}) = k\exp[-D(\boldsymbol{X};\boldsymbol{C}_j)] \qquad (3\text{-}62)$$

$$j = 1,\cdots,M$$

其中，k 为常数，$\boldsymbol{C}_j$ 为第 j 个码矢，它是描述这个 HMM 第 j 个状态对应的观察值概密函数 $b_j(\boldsymbol{X})$ 的参数，$D(\boldsymbol{X};\boldsymbol{C}_j)$ 表示语音帧 $\boldsymbol{X}$

的倒谱系数 O_x 和 C_j 之间的欧氏距离，即有

$$D(X;C_j)=(O_x-C_j)^T(O_x-C_j)$$

$$=\sum_{m=1}^{p}(O_{xm}-C_{jm})^2 \tag{3-63}$$

这里，p 为 LPC 分析的阶数。倒谱系数由相应的 LPC 系数转换而成。O_{xm}和 C_{jm}分别为 O_x 和 C_j 的第 m 个元素值。

描述 HMM 的参数，除了 M 个观察值概密函数的参数之外，还有初始状态概率矢量 $\pi=(\pi_1,\cdots,\pi_M)$以及状态转移概率矩阵 $A=\{a_{ij}\}_{M\times M}$。根据 HMM 原理，由已知的训练集，每迭代一次，就可改进一次 HMM 参数集。当然，同时也改进了描述 M 个观察值概密函数的 M 个倒谱系数(对应码本中 M 个码字)。π 和 A 的初值可以选为等概率值。

$\bar{\pi}_i$ 和 $\bar{a}_{ij}$的重估迭代公式如(1-60)和(1-61)式所示。$\bar{C}_{jm}$的迭代重估公式为[237,247]

$$\bar{C}_{jm}=\frac{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{(l)}(j)\beta_t^{(l)}(j)O_{lm}^{(l)}/P(O^{(l)}/\lambda)}{\sum_{l=1}^{L}\sum_{t=1}^{T_l}\alpha_t^{(l)}(j)\beta_t^{(l)}(j)/P(O^{(l)}/\lambda)} \tag{3-64}$$

其中，λ 为本次迭代前的 HMM 参数集，α 和 β 为著名的前向和后向变量，$O_{lm}^{(l)}$为第 l 个句子中第 t 帧倒谱系数 $O_t^{(l)}$ 的第 m 个元素。实际处理时，为了防止下溢(underflow)，必须用比例因子(Scaling)进一步处理(3-64)式。迭代一次结束后，$\bar{\pi}_i$、$\bar{a}_{ij}$和 $\bar{C}_{jm}$又组成一个改进后的 HMM 参数集 $\bar{\lambda}$。根据 HMM 训练原理，$\bar{\lambda}$ 比 λ 更好地表征了训练数据集。因此，新的码本 $\bar{C}_{jm}$对训练数据总的量化误差会减少，码本的熵也会提高(码本的熵是由各个码字在所有训练数据集中出现的概率计算出来的，也是衡量码本质量的一个指标)。而且，这种改进可再次进行，直至某个收敛点为止。

经典 SOFM 量化的过程为：对一个待量化的序列 $O=O_1,O_2,\cdots,O_T$，对应一个句子的 T 帧倒谱系数，对于任一帧 O_t，与码本中

M 个码矢比较，选取距离最近者为 O_i 的量化结果。引入 HMM 思想，根据码本构造一个 HMM 之后，由经典的 Viterbi 算法，对于给定的序列 $O=O_1,O_2,\cdots,O_T$，可以求出一个状态序列 $Q=q_1,q_2,\cdots,q_T$，由于这里一个状态与码本中一个码矢相对应，因而求出了序列 O 的量化结果，由于 Viterbi 算法给出的状态序列是最大似然意义上的一个最佳结果，因此，这样得到的量化结果是相对序列 O 的一个全局的最佳结果，比起经典量化方法中分别求出 O 中各帧 O_i 的单个最佳结果组合 O 的量化结果要好。必须指出，在量化的过程中，如果 π 和 A 总是保持等概率的初值，而且，Viterbi 算法也使用对数形式，那么，Viterbi 算法量化的结果就退化为经典 SOFM 量化的结果。这里因为 π 和 A 为等概率值，意味着码字间没有关系，因此，Viterbi 算法得到的全局最优就退化成经典 SOFM 量化的局部最优。但若 π 和 A 由重估公式改进之后，新的全局优化的量化结果应更好一些。

(2)实验

实验数据是我们所做的一个用于连续语音识别的语音库的一部分(参见 3.1.2)，语音通过一个 TMS320C25 信号处理板，以 10kHz 采样(12bit 量化)送入计算机，做预加重处理，帧长 200 点(20ms)，帧移步长 100 点，加汉明窗，用自相关法求取 12 阶 LPC 系数，转化成 12 阶倒谱系数。为了矢量量化的方便，对倒谱系数做了双线性变换，以压缩其动态范围，矢量量化码本大小为 256。实验数据共有 554 个汉语句子的大约 16 万帧倒谱系数。

首先，用经典 SOFM 方法产生码本，用码本对所有训练数据做量化，求出总的平均每帧的量化误差和码本的熵值。然后，用混合 SOFM/HMM 方法改进旧码本产生新码本，再用 Viterbi 算法做量化，也可求出总的平均每帧的量化误差以及码本的熵值。实验结果如表 3.5 所示。

表 3.5　对实验数据矢量量化的对比实验结果

方法	平均每帧量化误差	码本的熵
经典 SOFM	0.087 4	6.263 6
混合 SOFM/HMM	0.063 3	6.503 4

上述初步实验表明，相对经典 SOFM 矢量量化方法而言，新的 SOFM/HMM 方法，对实验数据的平均每帧量化误差明显减少，码本熵也明显提高。因此，可以说新的矢量量化方法的性能优于经典方法。但是，无论是码本的训练还是量化的过程，新方法都更复杂、更费时，需要进一步改进。

第四章　HMM 其它问题讨论

本章分三节讨论与 HMM 相关的三方面问题，旨在使对 HMM 的了解更广更深入一些。首先，在第一节中，讨论 HMM 与神经网络(Neural Networks，NN)的多方面的密切联系。与 HMM 一样，NN 在 80 年代中期也重新获得广泛的重视，在很多领域，包括语音处理，都有令人鼓舞的成功应用[115,150,151]。在不少权威国际会议上的大会报告中，知名学者们都以 HMM 和 NN 为中心介绍语音处理(识别)的最新进展[57,158]。事实上，两者的确存在紧密的联系[255]。这里主要涉及到一种著名的 NN——多层感知机(Multilayer Perceptrons，MLP)，讨论将包括 HMM 和 MLP 算法上的统一描述以及兼有两者之长的混合 HMM/MLP 方法在语音处理中的应用。然后，第二节讨论 HMM 算法的 VLSI 设计，这是 HMM 算法的专用芯片实现的基础。这里将给出 HMM 算法的多处理器实现和 Systolic 结构。最后，第三节讨论了有关 HMM 语音处理系统软硬件实现的一些问题，给出了 HMM 三大基本算法之一的 Viterbi 算法的 C 语言程序，并以 TMS320C25 芯片为核心，设计实现了一个有一定通用性的语音处理硬件系统。

§4.1　HMM 与神经网络(NN)

4.1.1　HMM 与多层感知机(MLP)的统一描述

感知机(Perceptron)，是 50 年代由 Rosenblatt 提出的一个单层网络，输入信息通过一个阶梯型函数作用后，直接在输出节点给

出信息。原始的感知机算法只有一个输出节点，相当于现在所说的神经网络中单个神经元。由于感知机提出了自组织、自学习的思想，对能够解决的问题，有一个收敛的算法，并从数学上给出了严格的证明，因而，它对神经网络的发展起了很大的推动作用[119,150,151]。但不久，人们发现它亦具有很大的局限性，尤其是Minsky 和 Papert 指出了感知机只能解决一阶谓词逻辑问题[161]。此后，对感知机的研究就冷了下来。1985 年，Rumelhart、McClelland 和他们的同事们组成的 PDP（Parallel Distributed Processing）小组，提出了著名的 BP（Back-Propagation）算法，给出了多层感知机（MLP）的学习方法[206]，从而使感知机这一思想得以复活。由于 MLP 能解决高阶谓词逻辑问题，尤其是其强大的模式分类能力，它在很多领域（包括语音处理领域）中都有非常成功的应用，成为最为知名的一种神经网络[42,43,45,82,212,222]。

所谓 MLP，就是一种多层网络，不仅有输入节点、输出节点，而且还有一层或多层隐节点。对于输入值，要先向前传播到隐节点，经作用函数运算后，再把隐节点的输出信息传播到输出节点，最后给出输出值。网络形式如图 4.1 所示。由于 MLP 网络由 BP 算法训练，因此，MLP 也常被称之为 BP 网络。BP 算法的学习过程，由正向传播和反向传播组成。在正向传播过程中，输入信息从输入层经隐节点层逐层处理，并传向输出层，每一层的神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出，则转入反向传播，将误差信号沿原来的连接通路返回，通过修改各层神经元权值，使误差信号最小。在文献[150,151]中对 BP 算法有清晰简明的描述。

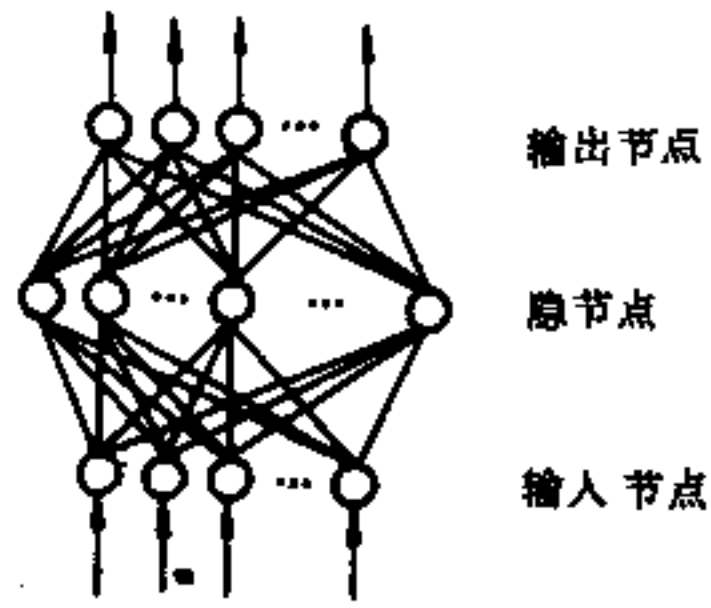


图 4.1　MLP 网络

Hwang、Vlontzos 和 Kung 等人根据 MLP 的一般网络形式

——L 层前馈连接主义网络(L-layer feedforward connectionist network),给出了 HMM 和 MLP 模型算法上的统一描述[100,101]。这种网络如图 4.2 所示。事实上,相当多的神经网络都可以归结为

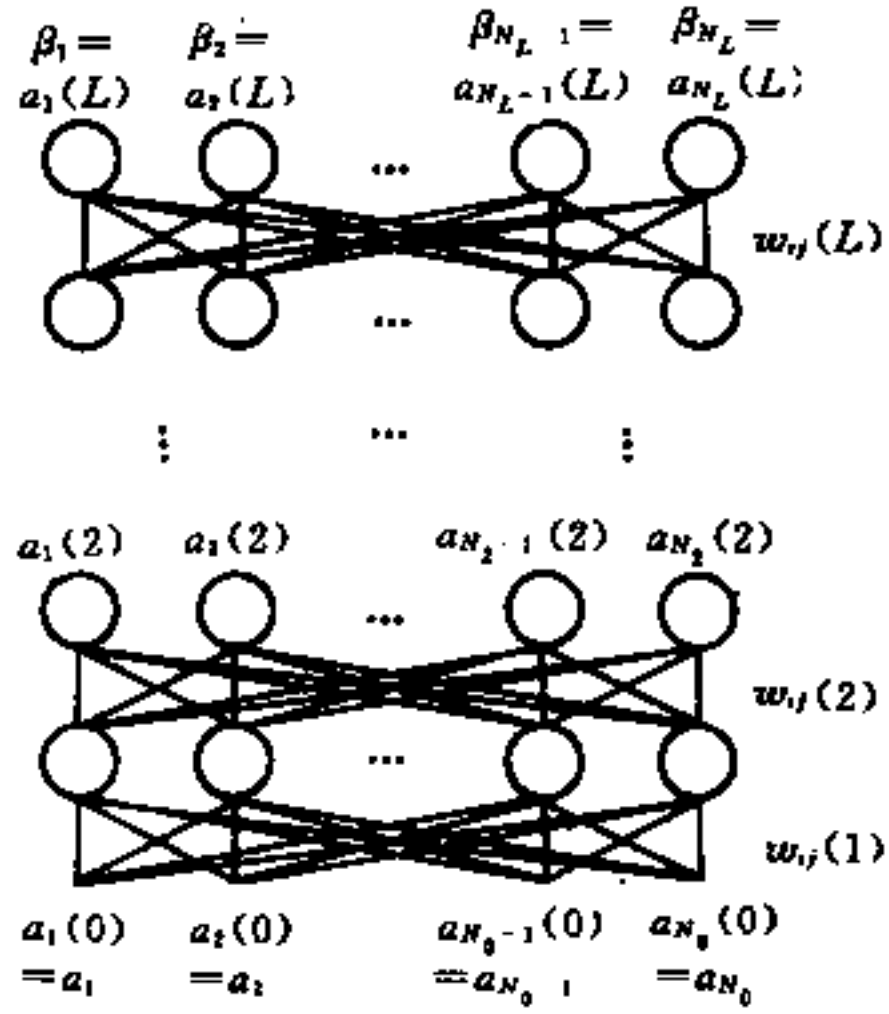


图 4.2 L 层前馈连接主义网络

这种网络结构。它的一种特殊情形(主要标志为各层权值一样,即 $w_{ij}(l) = w_{ij}, \forall l$),称之为回归齐次 BP 网络(recurrent homogeneous back propagation network),在回忆阶段(retrieving phase),对应于输入(测试模式)$\{a_i, i=1,\cdots,N_0\}$,L 层网络的系统动力学通过所有层迭代在输出层产生对应响应$\{\beta_i, i=1,\cdots,N_L\}$:

$$u_i(l+1) = \sum_{j=1}^{N} w_{ij} a_j(l) \tag{4-1a}$$

$$a_j(l+1) = f_i(u_i(l+1) + \theta_i(l+1)) = f_i(l+1) \tag{4-1b}$$

这里 $1 \leqslant i \leqslant N_{l-1}$, $0 \leqslant l \leqslant L-1$, f_i 为非减可微函数,如 sigmoid 函数,为简单起见,把外加输入$\{\theta_i(l+1)\}$处理为特殊的可修正的突触权值$\{w_{i0}(l+1)\}$,且 $a_0(l)=1$。

而对于 HMM 的前向算法,如(1-21)式所示,也可写成如下形

式：

$$u_i(l+1)=\sum_{j=1}^{N} w_{ij}a_j(l) \tag{4-2a}$$

$$a_i(l+1)=f_i(\theta(l+1))u_i(l+1) \tag{4-2b}$$

这里，$0\leqslant l\leqslant L-1$，$1\leqslant i\leqslant N$。$w_{ij}$表示 HMM 中状态转移概率 a_{ji}；$a_j(l)$就是著名的前向变量$\alpha_l(j)$.如(1-19)式定义，而且 $f_i(\theta(l+1))$就是观察值输出概率 $b_i(\theta(l+1))$，其中外加输入 $\theta(l+1)$为时刻 $l+1$ 的输入观察值，类似于 HMM 算法中常见的 O_{l-1}（l 对应于时间变量）。比较(4-1)式和(4-2)式，可见在回忆阶段，*HMM* 和回归齐次 *BP* 网络的描述是一致的。在学习阶段，回归齐次 *BP* 网络的学习过程为：给定一输入训练模式对：$\{\alpha_i, i=1,\cdots,N_0\}$和$\{t_j, j=1,\cdots,N_L\}$，目的是使对所有层迭代选择一个$\{w_{ij}\}$集合，使平方误差 E 最小，这里

$$E=\frac{1}{2}\sum_{i=1}^{N_L}(t_i-a_i(L))^2=\frac{1}{2}\sum_{i=1}^{N_L}(t_i-\beta_i)^2 \tag{4-3}$$

根据梯度下降法，有

$$\begin{aligned} w_{ij} &\Leftarrow w_{ij}-\eta\frac{\partial E}{\partial w_{ij}}=w_{ij}-\eta\sum_{l=1}^{L}\frac{\partial E}{\partial w_{ij}(l)} \\ &=w_{ij}-\eta\sum_{l=1}^{L}\delta_i(l)\frac{\partial a_i(l)}{\partial u_i(l)}a_j(l-1) \\ &=w_{ij}-\eta\triangle w_{ij} \end{aligned} \tag{4-4}$$

其中，反向传播校正信号 $\delta_i(l)$为

$$\delta_i(l)=\frac{\partial E}{\partial a_i(l)}=\sum_{j=1}^{N_{l+1}}\delta_j(l+1)\frac{\partial a_j(l+1)}{\partial u_j(l+1)}w_{ji}(l+1) \tag{4-5}$$

且

$$\delta_i(L)=-(t_i-\beta_i) \tag{4-6}$$

由于 HMM 和回归齐次 BP 网络的小差别，如(4-1)式和(4-2)式所示，而且，根据 HMM 训练的特点，梯度下降法的结果为

$$\begin{aligned} w_{ij} &\Leftarrow w_{ij}\cdot\eta\sum_{l=1}^{L}\delta_i(l)f_i(\theta(l))a_j(l-1) \\ &=w_{ij}\cdot\eta\triangle w_{ij} \end{aligned} \tag{4-7}$$

而反向传播校正信号 $\delta_i(l)$ 为

$$\delta_i(l) = \sum_{j=1}^{N} \delta_j(l+1) f_j(\theta(l+1)) w_{ji} \tag{4-8}$$

尤其是，由于约束条件 $\sum_{i=1}^{N} w_{ij} = 1$，可推导出：

$$\eta = \frac{1}{\sum_{l=0}^{L-1} \delta_j(l) a_j(l)} \tag{4-9}$$

于是 $$w_{ij} \Leftarrow w_{ij} \frac{1}{\sum_{l=1}^{L-1} \delta_j(l) a_j(l)} \cdot \sum_{l=1}^{L} \delta_i(l) f_i(\theta(l)) a_j(l-1) \tag{4-10}$$

显然，HMM 中状态转移概率的重估公式，如(1-41)式所示，亦可写成与(4-10)式一样的形式，注意到 $\delta_j(l)$ 事实上就是 HMM 中的后向变量 $\beta_l(j)$，如(1-24)式所示。Niles 和 Silverman 也给出了同一结论[174]。HMM 中的其它参数，也可用类似的方式由对应的回归齐次 BP 网络中的梯度下降法求出。这样，从算法的形式描述上实现了两者的统一。只不过处理梯度增量时，HMM 用的是乘，而 BP 网络用的是减，即相应算子的具体内容不同。

4.1.2 混合 HMM/MLP 方法

HMM 在语音处理各个领域都获得了巨大的成功，它处理语音信号中包含的统计信息和序列信息(随时间而变化的信息)的强大能力得到了公认。但是，一般认为，HMM 也有不少缺陷：由于训练准则和算法的限制，它对模式的辨识能力较差；模型的拓扑结构和观察值概密函数形式的先验选择往往和实际有出入；认为状态序列由一个一阶 Markov 链产生也不一定妥当等等。与 HMM 相比，神经网络，特别是 MLP，由于训练算法的特点(基于模式辨别的训练方式)而具有极强的模式辨识分类能力，而且，对输入的统计特性等不必作先验假定等等。但 MLP 处理信号的序列信息能力较差。虽然，改进 MLP，使之能够处理序列输入，不失为一种好

的选择，例如，将序列输入变为好像固定的模式，在孤立语音单元识别中，每个单元和 MLP 一个特定输出单元相联系，而待识别的模式被看作一个整体进入网络[43,181]；或者，将 MLP 改成回归(recurrent)网络(连接加上时延以及反馈回路)，也能处理序列输入[41,222,224]。但是，在解决语音处理(识别)的问题时，综合两类方法的优点，提出兼有两者之长的混合 HMM/MLP 方法，是一个更有吸引力的方案，不少学者对此作了卓有成效的努力[39]。

其中一个有代表性的工作，就是用 MLP 来计算 HMM 系统中状态对应的观察值输出概率[40,41,165]，基本思路为：MLP 每个输出单元和 HMM 中每个状态相对应，这样，对一个特定输入矢量 x_n，通过训练 MLP 就可产生概率 $p(\theta_k/x_n)$，θ_k 为 HMM 的 N 个状态中的一个。这是因为，对输入训练集$\{x_1, x_2, \cdots, x_T\}$，在某一时刻输入 x_n，并且这个输入和 HMM 的状态 θ_k 相对应，那么，MLP 参数的训练是最小化下式：

$$E = \frac{1}{2}\sum_{n=1}^{T}\sum_{k=1}^{N}[g_k(x_n) - d_k(x_n)]^2 \tag{4-11}$$

其中，$g_k(x_n)$表示 x_n 为输入时，MLP 与状态 θ_k 相联系的输出单元 k 的输出值，$d_k(x_n)$就是相应的目标值。可以证明[41]，只要 MLP 参数足够多，而且训练过程不在局部最小处停留，那么，MLP 的最佳输出值就是所要的概率值

$$g_k^{opt}(x_n) = p(\theta_k/x_n) \tag{4-12}$$

这些后验概率会产生一个最佳的分类。而且，根据 Bayes 准则，这个概率可以转化成 HMM 中状态对应的观察值输出概率

$$p(x_n/\theta_k) = \frac{p(\theta_k/x_n)p(x_n)}{p(\theta_k)} \tag{4-13}$$

显然，这个混合 HMM/MLP 估计出的 HMM 状态对应观察值输出概率具有更好的辨别分类的性质，而且，对训练数据的统计分布也不必作假定。由于 MLP 可以考虑输入的上下文相关信息，这样，混合 HMM/MLP 就没有它们各自的那些不足。

另一类混合 HMM/MLP 方法，是针对线性预测 HMM(参见 2.1.2)而提出的，即用 MLP 作为一个非线性的预测器[141]，基本思想为：对每一状态 θ_k，对应一个特殊的自回归过程 F_k，训练时，当预测阶数为 p 时，F_k 的参数确定由最小化下式完成：

$$E = \sum_{k=1}^{N} \sum_{x_n \in \theta_k} \| \hat{x}_n - x_n \|^2 \tag{4-14}$$

其中，$x_n \in \theta_k$ 表示 x_n 在状态 θ_k 上观察到，而

$$\hat{x}_n = F_k(X_{n-p}^{n-1}), \text{且 } X_{n-p}^{n-1} = \{x_{n-p}, \cdots, x_{n-1}\}$$

可以推导出，如果预测误差 $e_n = \hat{x}_n - x_n$ 为零均值单位方差的高斯噪声，那么，最小化(4-14)式的 E 就等价于下式的估计：

$$\prod_{k=1}^{N} \prod_{x_n \in \theta_k} p(x_n / \theta_k, X_{n-p}^{n-1})$$

$$= \prod_{k=1}^{N} \prod_{x_n \in \theta_k} (2\pi)^{-d/2} \exp\left[-\frac{1}{2} \| x_n - F_k(X_{n-p}^{n-1}) \|^2 \right] \tag{4-15}$$

其中，$x_n \in R^d$，(4-15)式显然是标准线性预测 HMM 观察值概密函数的一般化。标准线性预测 HMM 中，$F_k(X_{n-p}^{n-1})$是一个线性函数，即

$$\hat{x}_n = \sum_{i=1}^{p} a_{ki} \cdot x_{n-i} \tag{4-16}$$

但这里，MLP 在输入 X_{n-p}^{n-1}时训练，最小化 e_n，使 $F_k(X_{n-p}^{n-1})$成为一个非线性预测器，这样，(4-15)式的估计性能会更好。在这个方案中，HMM 每个状态对应于一个特定的 MLP(标准线性预测 HMM 中每个状态对应一组 LPC 系数)。

除此之外，还有不少新颖的混合 HMM/MLP 方法在语音处理中得到成功应用的报道[36,77,79,140,161,166,167,174]。因此，在解决语音处理(识别)问题时，同时综合采用 HMM 和 NN 两类方法，是值得重视的。事实上，在我们的基于离散 HMM 大词汇量连续语音识别系统中，也采用了一种有名的神经网络——Kohonen 的自组织特征映射(SOFM)来完成语音预处理中的矢量量化任务(参见

3.1.2)。

§4.2 HMM 算法的 VLSI 设计

4.2.1 多处理器实现

HMM 算法中，基本的计算就是前向变量 α（如(1-19)式定义）和后向变量 β（如(1-24)式定义）的迭代运算，如(1-21)式和(1-26)式所示，对于观察值序列 $O=O_1,O_2,\cdots,O_T$，重写如下：

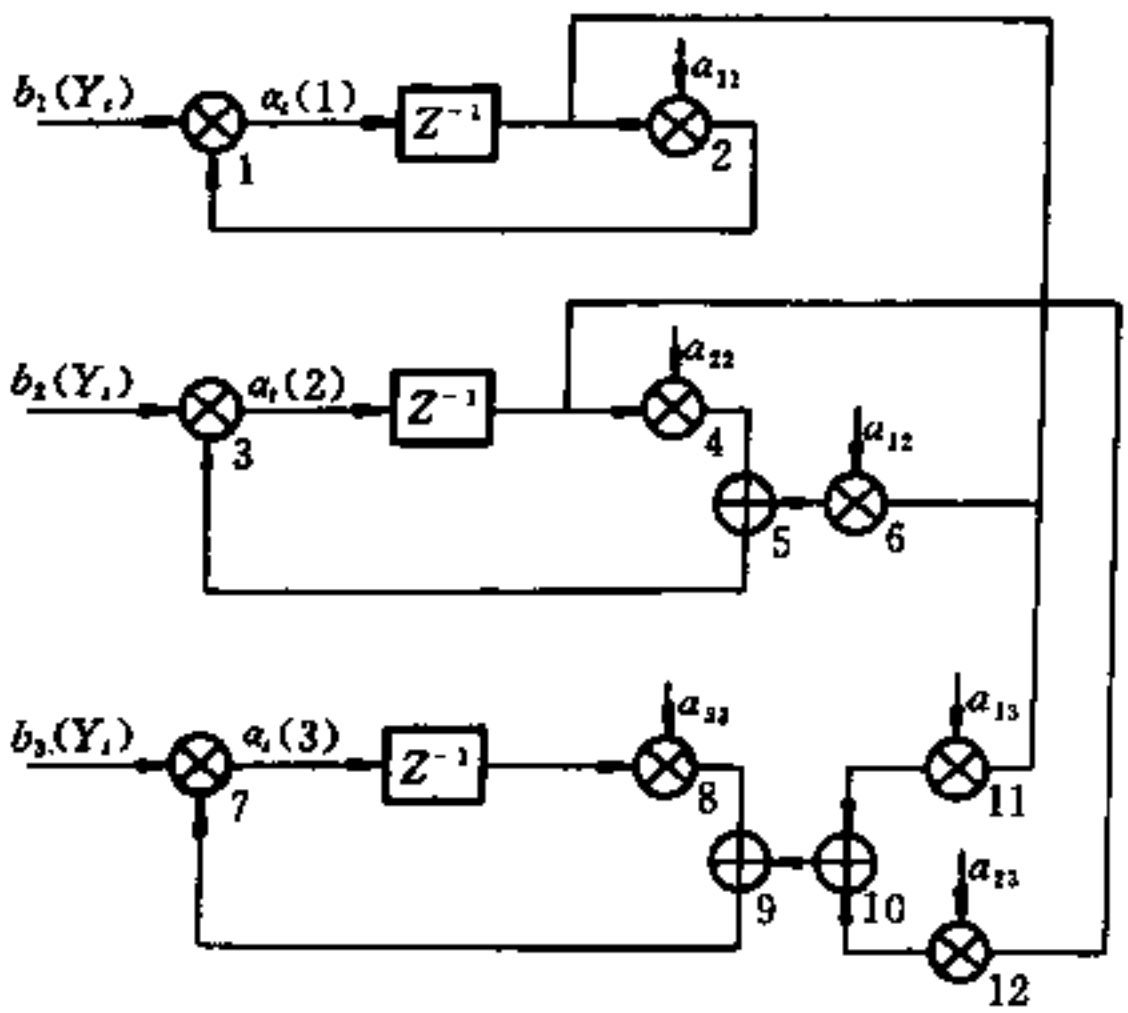
$$\alpha_t(i)=\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji}b_i(O_t)$$

$$i=1,\cdots,N,t=1,\cdots,T \tag{4-17}$$

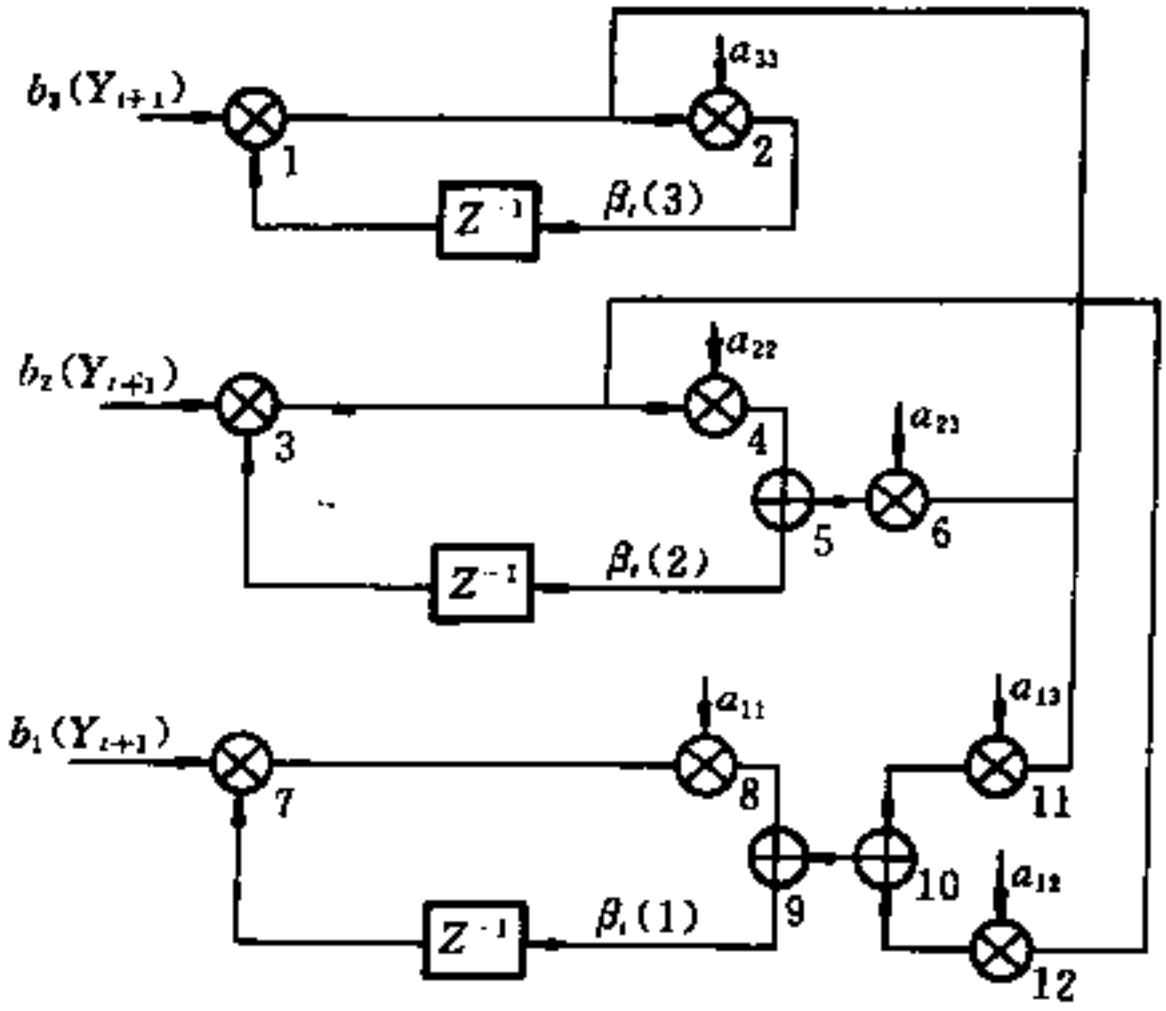
$$\beta_t(j)=\sum_{i=1}^{N}a_{ji}b_i(O_{t+1})\beta_{t+1}(i)$$

$$j=1,\cdots,N,t=1,\cdots,T \tag{4-18}$$

如果 HMM 为左-右形式，即 $a_{ij}=0(j<i)$，那么，当状态数 $N=3$ 时，α 和 β 计算的流程图如图 4.3 所示[182]。对于流程图中标号 1 到 12 的运算的时间关系进行分析，可以得出 α 和 β 计算的时序关系，如图 4.4 所示。这个时序图就是用多处理器来实现算法的基础。经过分析可知[182]，用图 4.6 所示的环形多处理器结构可以实现 α 和 β 的计算，如图 4.5 所示。顺时针方向计算 α，反时针方向计算 β。这个双向环形结构含有 $2N$ 个处理器，工作效率很高，使处理器之间的通信最小化了。



(a)



(b)

图 4.3　3 状态左-右形式 HMM 的 α 和 β 计算流程图

(a) α 计算流程图；　(b) β 计算流程图

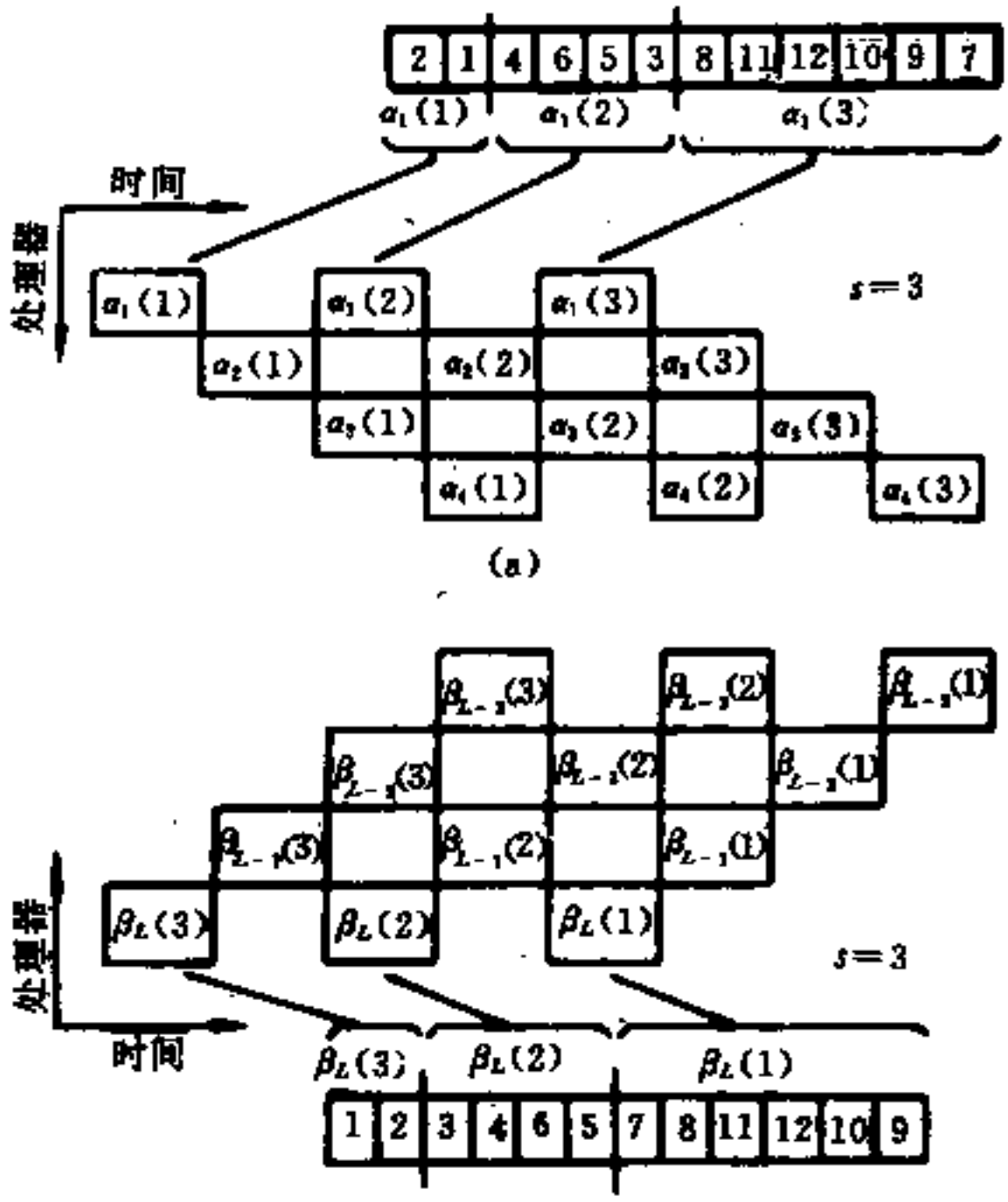


图 4.4 α 和 β 计算的时序关系

(a)α 时序；(b)β 时序

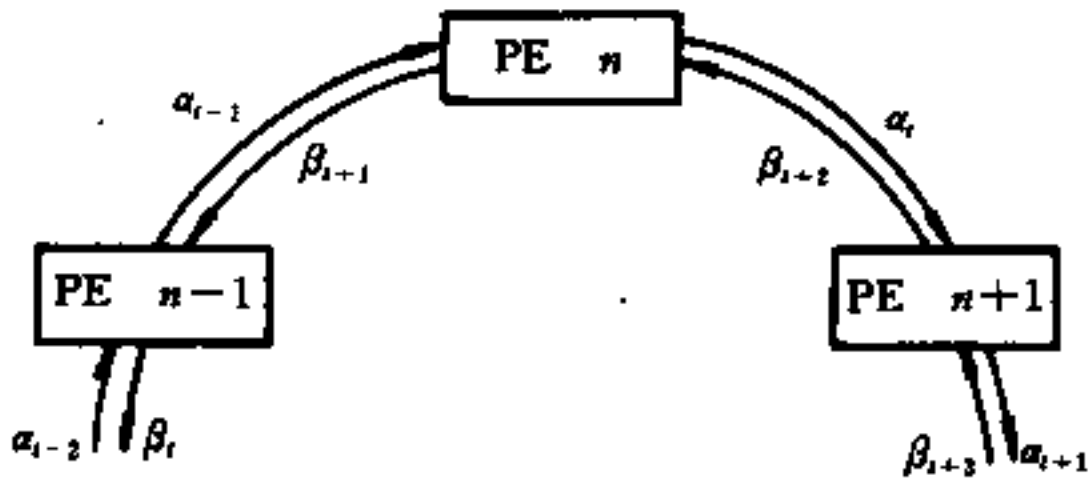


图 4.5 双向环形多处理器 α 和 β 的数据流

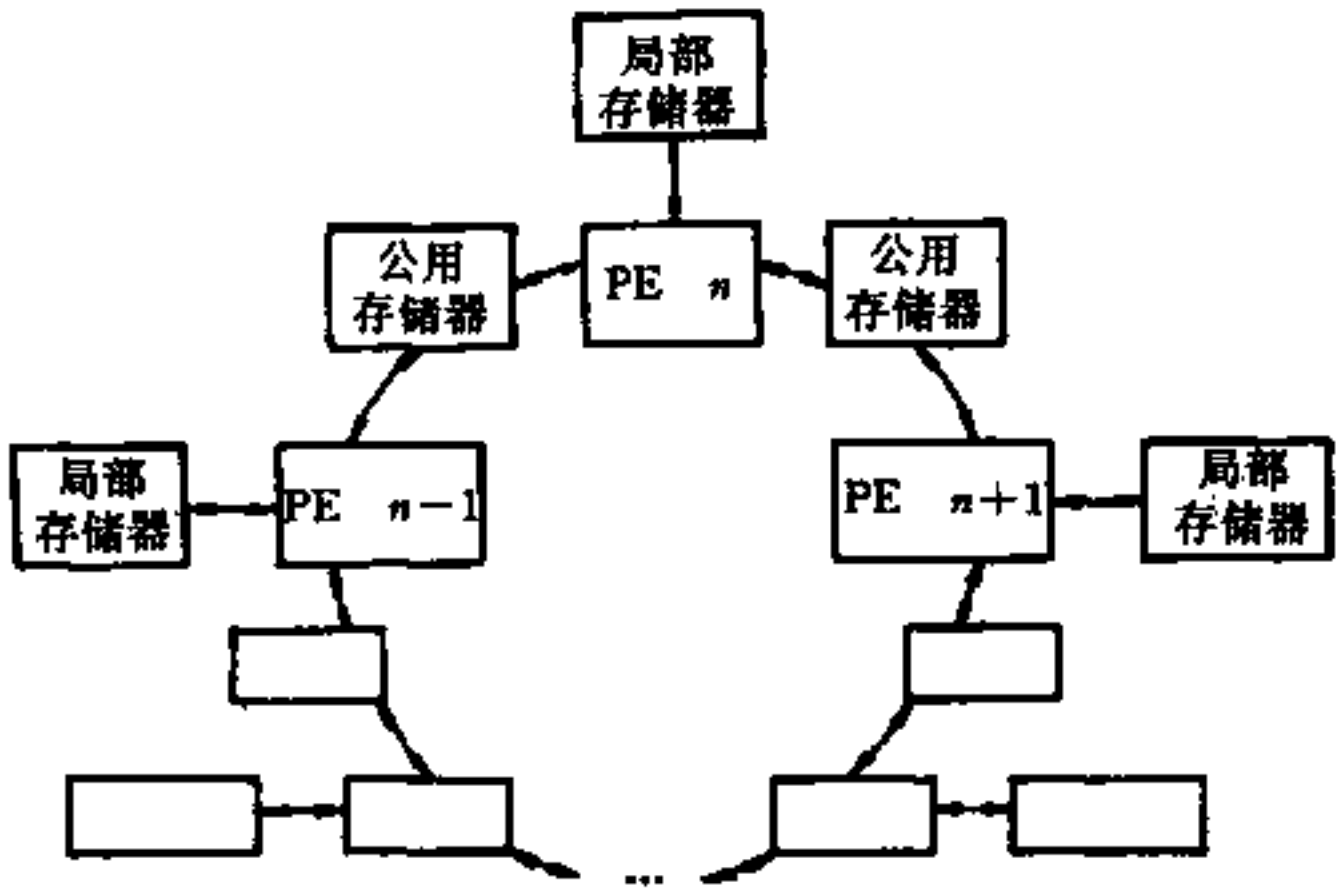


图 4.6 环形多处理器结构

4.2.2 Systolic 结构

算法的结构，是采用专用阵列处理器技术实现算法的基础。根据 VLSI 设计原理对结构的要求，现在公认最有影响力的是 1978 年 H. T. Kung 等提出来的 Systolic 结构[124,125]以及 1982 年 S. Y. Kung 等提出来的 Wavefront 结构[127]。这两种结构，前者是同步工作方式，后者为异步工作方法。这里只讨论一些 HMM 算法的 Systolic 结构。

所谓 Systolic 结构，按照 H. T. Kung 的定义，它是一个由处理器或称处理单元(Processing Element，PE)组成的一个网络，数据有节奏地通过网络，计算出需要的结果。因生理学家用 Systole 表示心脏和动脉有节奏地收缩使血液在体内循环，上述网络与此功能类似，因此得名。

过去十多年里，人们对信号处理中的大量经典算法，如滤波、卷积、变换、矩阵运算等等都导出了常常是不止一种的 Systolic 结构，而且，对系统地从算法到 Systolic 结构的方法，也进行了大量

研究，提出了不少方法和步骤，其中，S. Y. Kung 的三步法是较为成熟的一种[126]：

算法 ① Dependence Graph(DG) ② 信号流图(SFG) ③ Systolic 结构。

根据三步法，从 HMM 前向算法入手，将运算化为一串矩阵-矢量相乘，利用非线性分配思想，可以导出一种 Systolic 结构[253]。

但是，关于 HMM 算法的 Systolic 结构，目前公认的最为成功的一种，是由 S. Y. Kung 等人提出的环形结构[101,128,129]。由于他们同时也提出了 HMM 和回归齐次 BP 网络的统一描述(参见 4.1.1)，这样，这里的 Systolic 结构，不仅适用于 HMM 算法，对回归齐次 BP 网络的训练和识别(或称回忆)，也是适用的。

在回忆阶段(Retrieving phase)，由(4-1)式和(4-2)式所示，所有的计算都可以归结为一个连贯的矩阵-矢量相乘(Matrix-Vector Multiplication，MVM)问题：

$$\boldsymbol{U}(l+1)=\boldsymbol{W}\boldsymbol{A}(l) \tag{4-19a}$$

$$\boldsymbol{A}(l+1)=f[\boldsymbol{U}(l+1),\boldsymbol{\theta}(l+1)] \tag{4-19b}$$

这里，$\boldsymbol{U}(l)=[u_1(l),\ u_2(l),\ \cdots,\ u_N(l)]^T$，$\boldsymbol{A}(l)=[a_1(l),a_2(l),\cdots,a_N(l)]^T$，$\boldsymbol{\theta}(l)=[\theta_1(l),\theta_2(l),\cdots,\theta_N(l)]^T$，$\boldsymbol{W}=\{w_{ij}\}$，且 $f[\boldsymbol{X}]$ 算子表示对矢量 $\boldsymbol{X}$ 的每一个元素作非线性函数 f_i 运算。由于连贯 MVM 问题的一个特征是每一次迭代的输出将被用作下次迭代的输入，经过仔细地安排处理单元 PE 和时序，可导出一个环形 Systolic 结构，如图 4.7 所示。在第 $l+1$ 次迭代，每一处理单元 PE，或者说，第 i 个 PE，可被看作一个网络单元，对应的输入权值 $(w_{i1},w_{i2},\cdots,w_{iN})$ 存储于第 i 个 PE 的存储器中，这样，第 $l+1$ 次迭代可描述如下：

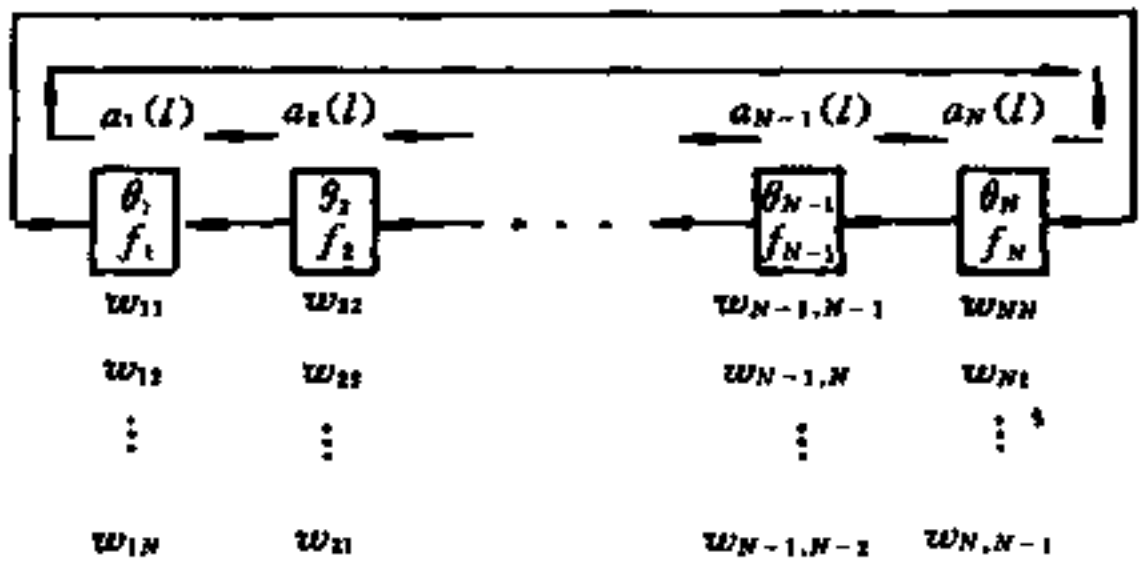
a. 第 i 个 PE 产生的每一个单元的激励 $a_i(l)$ 都乘以 w_{ii}，乘积加到累加器 $u_i(l+1)$ 中，其初始值为 0(HMM 时)。MAC(Multiply and Accumulation)之后，$a_i(l)$ 顺时针穿过环形阵列并访问一次其它 PE(在 N 个时钟单元里)，如图 4.7(a)。

b. 当 $a_j(l)$到达第 i 个 PE 时，它将与 w_{ij}相乘，乘积累加到 $u_i(l+1)$，如图 4.7(b)。

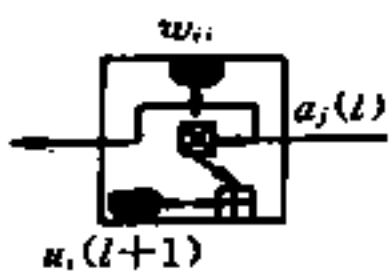
c. N 个时钟单元之后，累加器 $u_i(l+1)$得到了必要的乘积。

d. $a_i(l)$回到第 i 个 PE，还需一个时钟，处理器准备好完成非线性激活操作 f_i，以便为下一次迭代产生 $a_i(l+1)$，如图 4.7(c)。

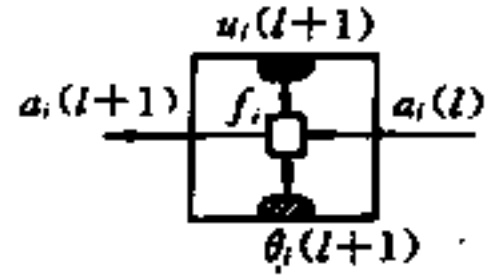
上述过程是全流水方式进行的，它将随着 l 的增加不断递归直到 L 次迭代全部完成。



(a)



(b)



(c)

图 4.7　连贯 MVM 问题的环形 Systolic 结构

(a)$a_j(l)$将以流水方式访问每个 PE；

(b)$a_j(l)$乘 w_{ij}，乘积累加到 $u_i(l+1)$；

(c)处理器完成非线性激励操作 f_i 并产生 $a_i(l+1)$

在学习阶段，由(4-4)、(4-5)式和(4-7)、(4-8)式所示，所有的计算分为两类：连贯的外积更新(Outer-Product Updating，OPU)问题和连贯的矢量-矩阵相乘(Vector-Matrix Multiplication，VMM)问题，这两类问题都能用环形 Systolic 结构来有效实现。

连贯的 OPU 问题，就是(4-4)和(4-7)式中$\triangle w_{ij}$的计算，可以

归纳为

$$\triangle w_{ij} \Leftarrow \triangle w_{ij} + g_i(l+1) \cdot h_j(l+1) \tag{4-20}$$

或

$$\triangle \boldsymbol{W} \Leftarrow \triangle \boldsymbol{W} + \boldsymbol{g}(l+1)\boldsymbol{h}^T(l+1) \tag{4-21}$$

这里，$\boldsymbol{g}(l+1)=[g_1(l+1), g_2(l+1), \cdots, g_N(l+1)]^T$，$\boldsymbol{h}(l+1)=[h_1(l+1), h_2(l+1), \cdots, h_N(l+1)]^T$，特别是，对于 HMM，$g_i(l+1)=\delta_i(l+1)f_i(\theta(l+1))$，$h_j(l+1)=a_j(l)$，如(4-7)式所示，连贯 OPU 问题的环形 Systolic 结构，如图 4.8 所示，工作过程大致为：

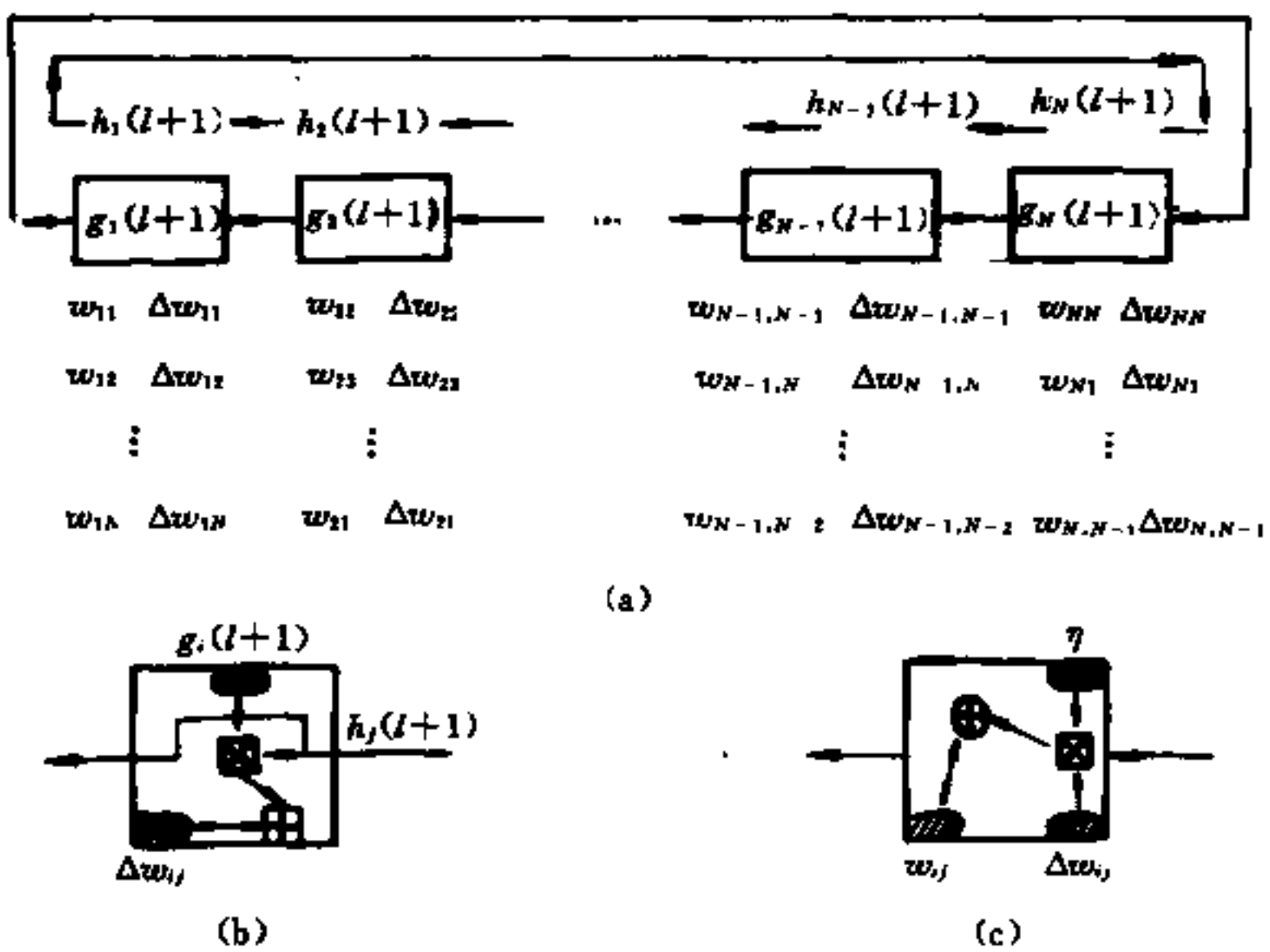


图 4.8　连贯 OPU 问题的 Systolic 结构

(a)$h_j(l+1)$以流水方式访问每个 PE；

(b)$h_j(l+1)$与 $g_i(l+1)$相乘，乘积累加到$\triangle w_{ij}$；

(c)环形结构在每个 PE 中更新 w_{ij}

a. 计算 $g_i(l+1)$值并存储在第 i 个 PE，第 j 个 PE 产生的 $h_j(l+1)$值在 N 个时钟内向左流过所有其它 PE，如图 4.8(a)所示。

b. 当 $h_j(l+1)$到达第 i 个 PE 时，将被乘以 $g_i(l+1)$，乘积累

加到$\triangle w_{ij}$(在第L次迭代的初值为0)，如图4.8(b)所示。

c. N个时钟之后，第$l+1$次迭代的$N\times N$个权值累加器都增长了。环形阵列现在准备为下一次的第l次迭代做连贯OPU累加。

当$\{\triangle w_{ij}\}$做完L层累加之后，环形阵列准备好在每个PE局部更新w_{ij}，这也需N个时钟：

$$w_{ij}\Leftarrow w_{ij}\oplus\eta\triangle w_{ij},\forall\ i,j \tag{4-22}$$

这里，算子$\oplus$在HMM中表示乘，在BP网中表示减，如图4.8(c)。η的计算，如(4-9)式所示，可在每个PE内部对所有l次迭代累加$\delta_j(l)a_j(l)$，累加完毕之后，再做一次除即可。

连贯VMM问题，就是(4-5)和(4-8)式中$\delta_i(l)$的计算，可以归纳为

$$\delta_i(l)=\sum_{j=1}^{N}g_j(l+1)w_{ji} \tag{4-23}$$

或
$$d^T(l)=g^T(l+1)W \tag{4-24}$$

这里，$d(l)=[\delta_1(l),\delta_2(l),\cdots,\delta_N(l)]^T$，第$l+1$次迭代的连贯VMM运算应当和第$l+1$次迭代的连贯OPU运算同时进行。其环形Systolic结构如图4.9所示。工作过程可以描述如下：

a. 在第$l+1$次迭代的第j个PE的信号$g_j(l+1)$和w_{ji}相乘，见图4.9(a)。

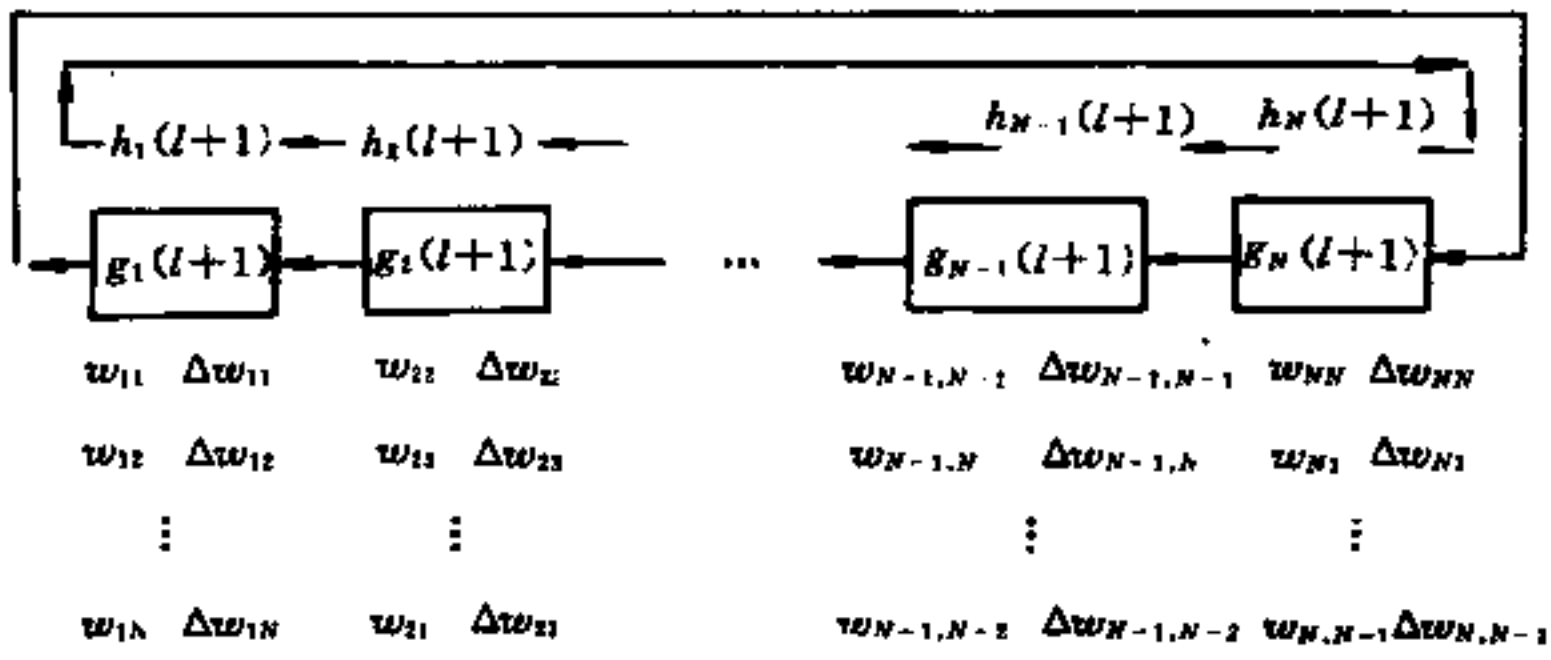
b. 乘积累加至新到的累加器$\delta_i(l)$(第i个PE的初值为0)中，与$h_i(l+1)$(如图4.8)一起向左穿过整个环形阵列，如图4.9(b)。

c. N次这样累加之后，累加器$\delta_i(l)$在累加到所有乘积之后，又回到第i个PE，即

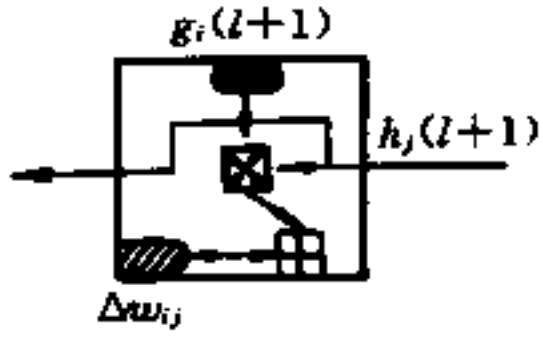
$$\delta_i(l)=\sum_{j=1}^{N}g_j(l+1)w_{ji} \tag{4-25}$$

它将被用来产生$g_i(l)$，如图4.9(c)。

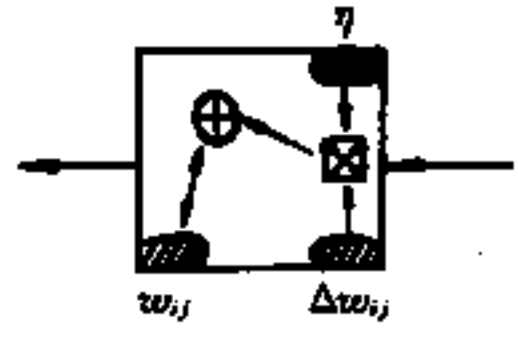
上述Systolic结构的讨论，虽是针对标准HMM的，但结果不难推广到HMM的各种修正形式，例如具有高阶Markov链的HMM[10,74]。



(a)



(b)



(c)

图 4.9 连贯 VMM 问题的 Systolic 结构
(a) $\delta_j(l)$以流水方式访问每个 PE；
(b)$g_i(l+1)$乘 w_{ij}，乘积累加到 $\delta_j(l+1)$；
(c)$\delta_i(l)$准备产生 $g_i(l)$

§4.3 关于 HMM 语音处理系统软硬件实现

4.3.1 HMM 算法 C 语言编程举例

C 语言是目前最为流行的计算机算法语言。因此，HMM 算法的 C 语言编程，是 HMM 语音处理系统软件实现的基础，对于从事 HMM 在语音处理中的应用研究是十分重要的。限于篇幅，这里只给出了 HMM 三大基本算法之一的 Viterbi 算法（如(1-29)式～(1-35)式所示）的 C 语言程序。参照这个例子，HMM 其它算法

的 C 语言编程用类似的方式是不难实现的。这个程序虽是针对离散 HMM 编写的，但很容易推广到其它类型的 HMM。

首先，给出了一个头文件，介绍了一些数据结构和变量的定义，它们对不同类型的 HMM 的各种算法 C 语言编程都是有参考价值的。接着，给出了 Viterbi 算法的 C 语言程序。最后，单独给出了一个计算 j 状态的观察值概密函数 $b_j(O_t)$ 的一个 C 语言函数。它对于将这里的针对离散 HMM 的程序移植到连续 HMM 或其它类型的 HMM 算法的 C 语言编程是有利的。

(1)HMM 算法头文件

```
/************File Name: hmm .h********
    the header file used for all the HMM functions
*****************************************/
#define LOG ZERO -2000000
                                /* the log value of zero */
#define MAX STATE 6
                                /* the number of the HMM states */
#define MAX PARAMETER PER STATE 256
                        /* the number of the parameters per state to describe
                        the observation density function of that state; for
                        example, for the discrete HMM, it is the number of
                        possible observation vectors or the size of VQ
                        codebook; for the Gaussian HMM, it is the number of
                        the mean and covariance parameters; for the linear
                        predictive HMM, it is the number of LPC coefficients
                        or the LPC analysis order+1 */
typedef struct
  {
  float init prob [MAX STATE];
  float trans prob [MAX STATE][MAX STATE];
  float observ param [MAX STATE]\
      [ MAX PARAMETER PER STATE];
  } HMM;
                             /* HMM parameters */
```

```
typedef struct
  {
  int frame_data; /* one frame data of the observation sequence, here the
                     VQ output of one frame for the discrete HMM,for other
                     types of HMM it may be modified as "float frame_data
                     [FRAME_LENGTH]" */
  }DATA;

#define log_delta(i,t)\
  log_delta_sequence[(long)(t) * MAX_STATE+(i)]
#define log_bjx(i,t) log_bjx_sequence[(long)(t) * MAX_STATE+(i)]
#define state_path(i,t)\
  state_path_sequence[(long)(t) * MAX_STATE+(i)]
            /* used in the Viterbi algorithm */
```

(2)Viterbi 算法 C 语言程序

```
/* * * * * * * * * * File Name: Viterbi.c * * * * * * * * * * *
        The Viterbi algorithm, one of the three basic algorithms for hidden
        Markov models (HMM) in speech processing, is used to obtain the
        optimum state sequence and the corresponding probability, i.e.,
        the Viterbi score, when the observation sequence and the HMM
        parameters are known.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
/* * * * * * * * * * Function Description * * * * * * * * * *
FUNCTION:
        float Viterbi(DATA *observ_sequence, int *state_sequence, int
        sequ_leng)
ARGUMENTS:
        first one: the known observation sequence for speech signal;
        second one: the output state sequence, to be computed;
        third one: the length of the observation sequence, or of the state
        sequence
RETURNS:
        the corresponding probability of producing the observation
        sequence, namely, the Viterbi score, when the HMM
        parameters are given
NOTES:
        1) the HMM parameter is the extern variable;
        2) this function is designed for all kinds of HMM;
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
```

```
#include <stdio.h>
#include <string.h>
#include <process.h>
#include <stdlib.h>
#include <math.h>
#include <alloc.h>
#include  "hmm .h"
/* ----extern variables-------- */
  extern HMM model;
/* ----function prototypes--- */
float Viterbi(DATA *observ_sequence, int *state_sequence,\
    int sequ_leng);
float bjx(DATA x, int j);
                    /* the observation density function of state j for
                    the frame data x, return the log value */
/* ---------------------------------- */
float Viterbi(DATA *observ sequence, int *state_sequence,\
    int sequ_leng)
{
float huge * log_delta sequence;
float huge * log bjx sequence;
int huge * state_path sequence;
float log trans prob[MAX STATE][MAX STATE];
float log_init prob[MAX STATE];
int i, j, k, t, temp_state;
float log_prob, temp_max, temp;
long max size;
/* -------Step 1: preparation---- */
max_size=(long)MAX STATE*sequ_leng;
if ((log_delta_sequence=(float huge *) farcalloc(max size,\
        sizeof (float)))==NULL)
        {
        printf( strerror("\n? ERR Not Enough Memory Space\
        ...log delta"));
        exit(1);
        }
if (log_bjx_sequence=(float huge *)farcalloc(max_size,\
       sizeof(float)))==NULL)
       {
```

```
        printf( strerror("\n? ERR Not Enough Memory Space\
        ...log bjx"));
        exit(1);
        }
if ((state path sequence=(int huge *) farcalloc (max size,\
          sizeof(int)))==NULL)
        {
          printf ( strerror("\n? ERR Not Enough Memory Space\
          ...state path"));
          exit(1);
          }
for(i=0; i<MAX STATE; i++)
{
if (model. init prob[i]==0) log init prob[i]=LOG ZERO;
else log init prob[i]=log(model. init prob[i]);
  for(k=0; k<MAX STATE;k++)
    if(model. trans prob [i][k]==0)\
      log trans prob[i][k]=LOG ZERO;
      else log trans prob[i][k]=log(model. trans prob[i][k]);
}
for(t=0;t<sequ leng; t++)
{
  for(i=0; i<MAX STATE;i++)
    log bjx(i,t)=bjx( *observ sequence,i);
  observ sequence++;
}
/*-------Step 2: initialization-------*/
for(i=0; i<MAX STATE;i++)
  {
  log delta(i,0)=log init prob[i]+log bjx(i,0);
  state path(i,0)=0;
  }
/*----------Step 3: recursion----------*/
for(j=0; j<MAX STATE;j++)
{
  for(t=1; t<sequ leng;t++)
  {
  temp max=log delta(0,t-1)+log trans prob[0][j];
  temp state=0;
```

```
    for(i=1;i<MAX STATE;i++)
      {
      temp=log_delta(i,t-1)+log trans prob[i][j];
      if(temp>temp max)
                {
                temp_max=temp;
                temp state=i;
                }
      }
      log delta(j,t)=temp max+log bjx(j,t);
      state path(j,t)=temp state;
    }
}
/*--------Step 4: termination----------*/
temp max=log_delta(0,sequ leng-1);
temp state=0;
for(i=1;i<MAX STATE;i++)
  if(log_delta(i,sequ leng-1)>temp max)
    {
    temp max=log delta(i,sequ leng-1);
    temp state=i;
    }
log prob=temp max;
/*--------Step 5: traceback------*/
state sequence[sequ leng-1]=temp state;
for(t=sequ leng-2;t>=0;t--)
  state sequence[t]=state path(state sequence[t+1],t+1);

farfree(log delta sequence);
farfree(log bjx_sequence);
farfree(state_path_sequence);

return log prob;
}
```

(3)计算 $b_j(O_t)$ 的函数

```
    ********File Name: bjx.c**********/
    **********Function Description**********
FUNCTION:
      float bjx(DATA x, int j);
```

```
        This function is used to compute the oservation density function of
        the state j for the frame data x;
ARGUMENTS:
                first one: the corresponding speech data of one frame;
                second one: the state;
RETURNS:
                the log value of the function for state j and the frame data x;
NOTES:
                1) the HMM parameter is the extern variable;
                2) for different types of HMM, the function bjx() should be
                   modified; the "DATA" structure defined in "hmm_.h" is also
                   changed; here only the bjx() for discrete HMM is presented;
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
#include <stdio.h>
#include <string.h>
#include <process.h>
#include <stdlib.h>
#include <math.h>
#include "hmm_.h"
/*------extern variables-------*/
  extern HMM model;
/*------function prototypes---*/
float bjx(DATA x, int j);
     /* the observation density function of state j for the frame data x */
float bjx (DATA x, int j)
{
float bjx_discrete_value;

if(model. observ_param[j][x. frame_data]==0)\
bjx_discrete_value=LOG_ZERO;
else bjx_discrete_value=log(model. observ_param[j][x. frame_data]);

return bjx_discrete_value;
}
/*-----------------------------*/
```

4.3.2 基于 TMS320C25 芯片的硬件系统设计

HMM 语音处理系统，是典型的数字信号处理系统。因此，采用目前很流行的数字信号处理芯片 TMS320C25 为核心来设计硬件系统是一个比较理想的选择，能兼顾系统的实时性和通用性。这是因为 TMS320C25 使用了哈佛结构、多级流水线、专用硬件乘法器、特殊的数字信号处理指令（例如灵活的位操作指令、强大的数据块传送指令等等）以及快速的指令周期（100ns）[17,18]。事实上，已有一些成功的系统使用了这种芯片[233]。这里给出一个有一定通用性的脱机的语音存储系统设计。这个系统也能改变为微机的一个扩展卡，为微机提供语音接口和语音信号的快速处理，此外，这个系统对设计 HMM 语音处理其它硬件系统（比如识别系统）也很有参考价值。

所设计的语音存储系统由 TMS320C25 的最小系统，加上 A/D、D/A 等部分组成。系统能在 A/D 和 D/A 两种状态下工作，在 A/D 状态，系统存储语音：模拟话音从话筒出来，经滤波、放大、A/D 数字化后，自动存储到容量为两帧语音的缓存之中，当存满一帧时，C25 对其作压缩处理，再将压缩后数据存入数据存储器，与此同时，A/D 产生的数字化数据继续放入缓存的另一帧，这个过程一直进行下去，直到数据存储器存满或收到系统中止工作的信号。另一方面，在 D/A 状态，过程正好反过来，系统恢复语音：C25 从数据存储器中取一帧已压缩的数据，作解压缩处理之后，放入缓存的一帧之中，系统再将这一帧语音经 D/A、滤波、放大，从喇叭中输出。与此同时，C25 继续处理数据存储器中下一帧数据，解压缩后，放入缓存另一帧内，等待 D/A 处理，这个过程一直持续到数据存储器内的数据全部取出或得到系统中止工作的信号。在这两种状态中，都用一个数字钟（秒，十秒，分三位）来显示语音存入或放出的长度（时间）。

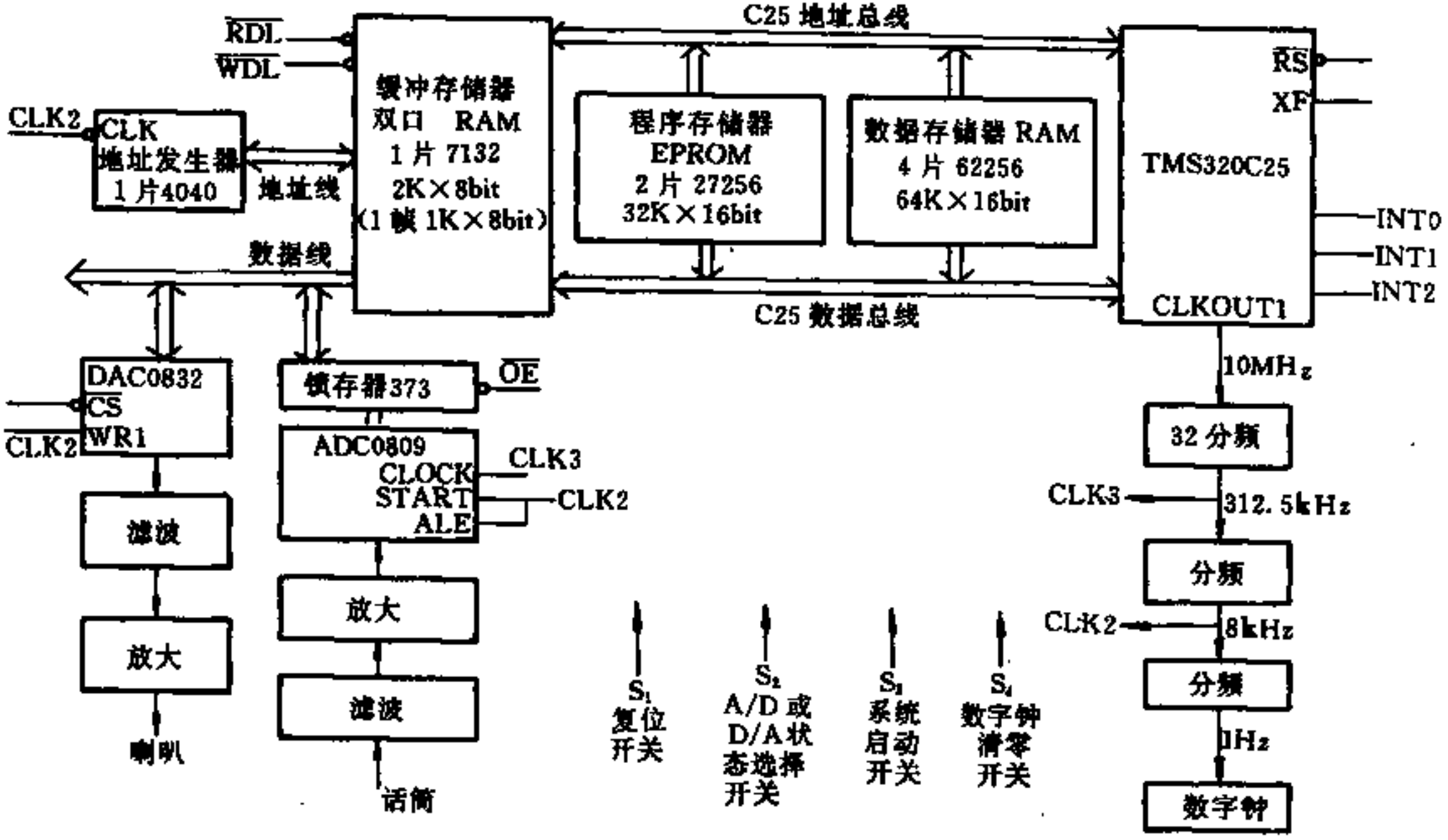


图 4.10 基于 TMS320C25 的语音存储系统主要部分框图

系统主要部分的框图如图 4.10 所示。其设计和实现的过程可以大致描述如下：

TMS320C25 有 16 位地址和数据线，对数据存储器和程序存储器的寻址均为 64K×16bit。系统中的数据存储器由 4 片 62256（每片 RAM 为 32K×8bit）组成，两片为一组，成为 16 位。类似，程序存储器由 2 片 27256（每片 EPROM 为 32K×8bit）组成。这样，系统有 64K×16bit 数据存储器，32K×16bit 程序存储器。由于缓冲存储器同时与 D/A、A/D 和 C25 相连，必须具有两套地址和数据线，即是双口 RAM，这里使用 1 片 IDT7132，2K×8bit，分成上下两帧，每帧 1K×8bit。由于 C25 程序存储器寻址范围尚有 32K×16bit 的剩余部分没有用到，7132 的一组地址和数据线就挂在这里与 C25 相连。7132 的另一组数据线通过锁存器 373 和 A/D 部分（1 片 ADC0809）相连，也与 D/A 部分（1 片 DAC0832）相连。系统的各时钟信号由 C25 的 CLKOUT1 脚产生的 10MHz 经分频得到，包括 ADC0809 用到的 312.5kHz 时钟，A/D、D/A 采样脉冲 8kHz，地址发生器 4040 计数时也用到的 8kHz 脉冲，数字钟用到的 1Hz 脉冲。

系统的控制，由 S_1、S_2、S_3 和 S_4 四个开关来完成。S_4 开关专门用于数字钟清零。S_1 为系统复位开关，用于中止系统正在进行的工作，它通过 C25 $\overline{RS}$脚使 C25 停止工作，它同时锁住 8kHz 时钟，即使 A/D、D/A 和数字钟也停止工作，当然，7132 的地址发生器也停止了对采样脉冲的计数，即停止了地址的累加。此时，数字钟显示的时间就是存入或放出的语音长度。S_2 开关选择系统工作的状态，即系统存入或放出语音，它是通过控制锁存器 373 的$\overline{OE}$脚和 DAC0832 的$\overline{CS}$脚来实现 7132 的数据线是和 A/D 部分还是和 D/A 部分连通，同时，S_2 也控制 7132 的$\overline{RDL}$和$\overline{WDL}$脚，实现 7132 是读还是写，即数据是出还是进 7132。S_3 开关用于开启系统的工作，即开启送往 A/D、D/A、4040 等处的 8kHz 时钟脉冲，数字钟也开始开作，同时，与开关 S_2 一起产生中断 INT0 和 INT1，通知

C25 可以进行语音存入或放出的工作。7132 放满一帧语音之后，应通知C25 对这一帧语音进行处理，这是由 7132 最高位地址 A10 的变化(4040 计数累加至 1K)，产生 C25 中断 INT2 来实现的。当然，系统启动工作时 4040 必须首先清零，以保证 C25 从 7132 的低 1K 数据帧开始处理。从 7132 取出一帧的过程类似。此外，当数据存储器 64K×16bit 全部放满或全部取出来，系统应停止工作，此时 C25 的 XF 脚产生一个控制信号，锁住 8kHz 时钟脉冲，即停止了 A/D、D/A 和数字钟等的工作。

结合系统的上述设计，C25 的程序编写思路为：系统开机后，执行主程序，然后进入等待循环，当收到 INT0 中断时，设置 A/D 允许标志；当收到 INT1 中断时，设置 D/A 允许标志；当收到 INT2 中断时，若有 A/D 允许标志，则将 7132 中一帧语音数据进行压缩处理，并将压缩结果放入数据存储器；若有 D/A 允许标志，则将数据存储器中一帧已压缩数据取出，解压缩处理后，将结果放入 7132 之中；处理完后，返回等待循环，直到收到下一次 INT2 中断。

根据系统的总的设计构想，我们实现了一个初步可演示的基于 TMS320C25 的语音存储系统[239]。

附录　Baum-Welch 算法中重估公式的推导

Baum-Welch 算法是估计 HMM 参数的经典方法，其著名的重估(reestimation)公式在本书中多次用到，是 HMM 在语音处理中应用的重要基础，在这个附录中将给出 Baum-Welch 算法中重估公式的推导。首先，给出重估公式成立的理论基础。然后，分别给出离散 HMM 和连续 HMM 具体的重估公式。

一、重估公式的理论基础[144]

引理：设 $u_i, i=1,\cdots,S$，为正实数，$v_i, i=1,\cdots,S$，为非负实数，即 $\sum_i v_i \geqslant 0$，那么，由对数函数的凹特性，有如下结论：

$$\ln\frac{\sum_i v_i}{\sum_i u_i} = \ln\left[\sum_i \left(\frac{u_i}{\sum_k u_k}\cdot\frac{v_i}{u_i}\right]\right.$$

$$\geqslant \sum_i \frac{u_i}{\sum_k u_k}\ln\frac{v_i}{u_i}$$

$$= \frac{1}{\sum_k u_k}\left[\sum_i (u_i \ln v_i - u_i \ln u_i)\right] \qquad (1)$$

此处所有求和均是从 1 到 S。

定义辅助函数

$$Q(\lambda,\bar{\lambda}) = \frac{1}{P(O/\lambda)}\sum_{\text{所有}S} P(O,S/\lambda)\ln P(O,S/\bar{\lambda}) \qquad (2)$$

其中，λ 为原来的模型 $\lambda=(\pi,A,B)$，$\bar{\lambda}$ 为新求取的模型 $\bar{\lambda}=(\bar{\pi},\bar{A},\bar{B})$，$O$ 为训练用观察值序列，$O=O_1,O_2,\cdots,O_T$，S 为某个状态序列 $S=q_1,q_2,\cdots,q_T$，那么，由引理易推出下面的定理：

如果 $Q(\lambda,\bar{\lambda})\geqslant Q(\lambda,\lambda)$，那么 $P(O/\bar{\lambda})\geqslant P(O/\lambda)$。

该定理构成了重估公式的理论基础：对辅助函数 $Q(\lambda,\bar{\lambda})$，只要能找到 $\bar{\lambda}$，使 $Q(\lambda,\bar{\lambda})$达到最大值，那么，就能保证 $Q(\lambda,\bar{\lambda})\geqslant Q(\lambda,$

λ)，从而使 $P(O/\bar{\lambda})\geqslant P(O/\lambda)$，这样，新得到的模型 $\bar{\lambda}$ 在表示训练序列 O 方面就比原来的模型 λ 要好。一直重复这个过程，直到某个收敛点，就可以得到根据训练序列 O 估计出的结果模型，而使 $Q(\lambda,\bar{\lambda})$最大而求取 $\bar{\lambda}$ 参数的公式就称之为重估公式。不同的 λ，其参数 π，A 和 B 就不同，重估公式的具体形式也不同，在给出几种典型 HMM 重估公式之前，先给出一个后面会用到的结论。

如果 $c_i>0, i=1,\cdots,N$，在 $\sum_{i=1}^{N}x_i=1$ 的约束条件下，函数

$$F(x)=\sum_{i=1}^{N}c_i\ln x_i \tag{3}$$

的唯一最大值点为

$$x_i=\frac{c_i}{\sum_{k=1}^{N}c_k} \tag{4}$$

二、离散 HMM 的重估公式[114]

因为 $$\ln P(O,S/\bar{\lambda})=\ln\bar{\pi}_{q_1}+\sum_{t=1}^{T-1}\ln\bar{a}_{q_tq_{t-1}}+\sum_{t=1}^{T}\ln\bar{b}_{q_t}(O_t) \tag{5}$$

所以有

$$Q(\lambda,\bar{\lambda})=\sum_{i=1}^{N}\sum_{j=1}^{N}c_{ij}\ln\bar{a}_{ij}+\sum_{j=1}^{N}\sum_{k=1}^{M}d_{jk}\ln\bar{b}_{jk}+\sum_{i=1}^{N}e_i\ln\bar{\pi}_i \tag{6}$$

其中

$$c_{ij}=\sum_{t=1}^{T-1}\xi_t(i,j) \tag{7}$$

$$d_{jk}=\sum_{\substack{t=1\\ O_t=V_k}}^{T}\xi_t(j) \tag{8}$$

$$e_i=\xi_1(i) \tag{9}$$

而 $\xi_t(i,j)$如(1-38)式定义，$\xi_t(i)$由(1-39)式定义。求 $Q(\lambda,\bar{\lambda})$的最大值，根据(3)和(4)式的结论，易得到离散 HMM 的重估公式，如(1-40)、(1-41)、(1-42)式所示。

三、高斯型连续 HMM 的重估公式[94,108,110]

这种 HMM 的观察值概密函数如(2-3)式所示，即

$$b_j(\boldsymbol{X}) = \sum_{k=1}^{K} c_{jk} b_{jk}(\boldsymbol{X}) = \sum_{k=1}^{K} c_{jk} N(\boldsymbol{X}, \mu_{jk}, \Sigma_{jk}) \qquad (10)$$

其中，$N(\boldsymbol{X}, \mu_{jk}, \Sigma_{jk})$为高斯概密函数，均值矢量为 μ_{jk}，方差矩阵为 Σ_{jk}，由于 π，A 的估计与离散 HMM 相同，因此，这里需要导出 c_{jk}，μ_{jk}和 Σ_{jk}的重估公式。此时

$$Q(\lambda, \bar{\lambda}) = \sum_{\text{所有}S}\sum_{\text{所有}K} \frac{P(O, S, K/\lambda)}{P(O/\lambda)} \ln P(O, S, K/\bar{\lambda}) \qquad (11)$$

其中 K 为某个分支，且

$$P(O, S, K/\lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t k_t}(O_t) c_{q_t k_t} \qquad (12)$$

$$\ln P(O, S, K/\bar{\lambda}) = \ln \bar{\pi}_{q_1} + \sum_{t=1}^{T-1} \ln \bar{a}_{q_t q_{t-1}} + \sum_{t=1}^{T} \ln \bar{b}_{q_t k_t}(O_t) + \sum_{t=1}^{T} \ln \bar{c}_{q_t k_t} \qquad (13)$$

因此

$$Q(\lambda, \bar{\lambda}) = Q_\pi(\lambda, \bar{\pi}) + \sum_{i=1}^{N} Q_{a_i}(\lambda, \bar{a}_{ij}) + \sum_{j=1}^{N}\sum_{k=1}^{K} Q_{b_j}(\lambda, \bar{b}_{jk}) + \sum_{j=1}^{N} Q_{c_j}(\lambda, \bar{c}_{jk}) \qquad (14)$$

其中
$$Q_\pi(\lambda, \bar{\pi}) = \sum_{i=1}^{N}\sum_{\text{所有}K} P(q_1 = \theta_i, K/O, \lambda) \ln \bar{\pi}_i \qquad (15)$$

$$Q_{a_i}(\lambda, \bar{a}_{ij}) = \sum_{j=1}^{N}\sum_{t=1}^{T-1}\sum_{\text{所有}K} P(q_t = \theta_i, q_{t+1} = \theta_j, K/O, \lambda) \ln \bar{a}_{ij} \qquad (16)$$

$$Q_{b_j}(\lambda, \bar{b_{jk}}) = \sum_{t=1}^{T} P(q_t = \theta_j, k_t = k/O, \lambda) \ln \bar{b}_{jk}(O_t) \qquad (17)$$

$$Q_{c_j}(\lambda, \bar{c_{jk}}) = \sum_{k=1}^{K}\sum_{t=1}^{T} P(q_t = \theta_j, k_t = k/O, \lambda) \ln \bar{c}_{jk} \qquad (18)$$

分别最大化 $Q_{c_j}(\lambda, \bar{c}_{jk})$和 $Q_{b_j}(\lambda, \bar{b}_{jk})$，考虑到$\bar{b}_{jk}(O_t)$为高斯概密函数，易得到(2-5)，(2-6)和(2-7)式所示重估公式。

四、线性预测型 HMM 的重估公式[109,113,187,237]

这种 HMM 的观察值概密函数如(2-31)式所示，即

$$b_j(\boldsymbol{X}) = (2\pi)^{K/2}\exp\left[-\frac{1}{2}\delta(\boldsymbol{X};\boldsymbol{a}_j)\right] \tag{19}$$

其中，K 为归一化语音帧 $\boldsymbol{X}$ 的长度，$\boldsymbol{a}_j$ 为描述 $b_j(\boldsymbol{X})$的一组 LPC 系数，记 $R_a(i)$和 $R_x(i)$分别为 $\boldsymbol{a}_j^T$ 和 $\boldsymbol{X}^T$ 的自相关函数，则有

$$\delta(\boldsymbol{X};\boldsymbol{a}_j) = R_a(0)R_x(0) + 2\sum_{i=1}^{p}R_a(i)R_x(i) \tag{20}$$

这里需要导出参数 $\boldsymbol{a}_j$ 对应的自相关函数 R_j 的重估公式，类似高斯型连续 HMM 重估公式的推导，有

$$Q(\lambda,\bar{b}_j) = \sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)\ln\bar{b}_j(O_t) \tag{21}$$

最大化 $Q(\lambda,\bar{b}_j)$等价于最小化下式：

$$\sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)\delta(\boldsymbol{O}_t;\bar{\boldsymbol{a}}_j)$$

$$= \sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)\left[R_a(0)R_t(0) + 2\sum_{i=1}^{p}R_a(i)R_t(i)\right] \tag{22}$$

其中，$R_t(i)$为 $\boldsymbol{O}_t$ 的自相关函数，$R_a(i)$为 $\bar{\boldsymbol{a}}_j$ 的自相关函数。上式还可进一步化为

$$R_a(0)\left[\sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)R_t(0)\right]$$

$$+ 2\sum_{i=1}^{p}R_a(i)\left[\sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)R_t(i)\right]$$

$$\triangleq\delta(\boldsymbol{O}_j^*;\bar{\boldsymbol{a}}_j) \tag{23}$$

其中，$\boldsymbol{O}_j^*$ 为某段语音，其自相关函数为

$$\bar{R}_j(i) = \sum_{t=1}^{T}P(O,q_t = \theta_j/\lambda)R_t(i) \tag{24}$$

那么，由 LPC 分析理论，最小化 $\delta(\boldsymbol{O}_j^*;\bar{\boldsymbol{a}}_j)$等价于用自相关法根据 $\bar{R}_j(i)$求其 LPC 系数 $\bar{\boldsymbol{a}}_j$。也就是说，$\bar{\boldsymbol{a}}_j$ 对应的自相关函数 $\bar{R}_j(i)$的重估公式如(24)式所示。

参 考 文 献

1 高雨青，黄泰翼，陈永彬. 隐 Markov 模型参数估计的一种新方法. 自动化学报，1991，17(1)：56～62

2 高雨青. 汉语全音节语音识别理论和系统研究：〔博士学位论文〕. 南京：东南大学，1989

3 中国电子，仪器仪表学会信号处理学会数字信号处理程序库编译组. 数字信号处理程序库. 北京：清华大学出版社，1983

4 姚天任. 数字语音处理. 武汉：华中理工大学出版社，1992

5 齐士钤，张家录. 汉语普通话辅音音长分析. 声学学报，1982，7(1)：8～13

6 王作英，郭进，孙甲松等. 赛德九一九语音识别与理解系统. 计算机智能接口与智能应用'93 会议论文集，黑龙江镜泊湖，1993：247～258

7 王作英. 非齐次语音识别 HMM 模型和 THED 语音识别与理解系统. 电信科学，1993，9(4)：31～35

8 战普明，陆大金，王作英. 用 KL 鉴别信息作为语音识别距离测度的研究. 电子学报，1993，21(1)：1～7

9 战普明，陆大金，王作英. 基于 KL 鉴别信息的语音识别粗分类. 电子学报，1993，21(4)：1～6

10 黄载禄，冯昭志，万发贯. 高阶神经网络与二阶隐马尔柯夫模型的统一的数学模型. 华中理工大学学报，1993，21(6)：1～5

11 徐波. 基于 HMM 的汉语语音全音节识别及大词汇识别研究：〔硕士学位论文〕. 北京：中科院自动化所，1992

12 王还. 汉语词汇统计与分析. 北京语言学院语言教学研究所. 北京：外语教学与研究出版社，1985

13 陈永彬，王仁华. 语言信号处理. 合肥：中国科大出版社，1990

14 马大猷. 语言信息和语言通信. 北京：知识出版社，1987

15 中科院计算中心概率统计组. 概率统计计算. 北京：科学出版社，1979

16 刘德贵，费景高，于泳江等. FORTRAN 算法汇编(第一分册). 北京：国防工业出版社，1980

17 汪亚南，王新扬，李晓峰. TMS320 系列高速单片计算机原理与应用. 成

都:电子科技大学出版社,1992

18 中科院声学所语言与通信研究室.第二代数字信号处理器 TMS32020、TMS320C25 用户指南.1991

19 Asai K,Hayamizu S, Handa K. Dividing the Distributions of HMM and Linear Interpolation in Speech Recognition. Proc. ICASSP'92, 1992,I, 29～32

20 Askar M, Derin H. A Recursive Algorithm for the Bayes Solution of the Smoothing Problem. IEEE Trans. AC, 1981, 26(2),558～561

21 Atal B S. Predictive Coding of Speech at Low Bit Rates. IEEE Trans. COM, 1982,30 (4), 600～614

22 Atal B S, Schroeder M R. Adaptive Predictive Coding of Speech Signals. BSTJ, 1970, 49(10):1973～1986

23 Bahl L R, Brown P F, De Souza P V, et al. Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. Proc. ICASSP'86,1986:49～52

24 Bahl L R, Brown P F, De Souza P V, et al. A Fast Algorithm for Deleted Interpolation. Proc. Eurospeech'91, 1991:1209～1212

25 Bahl L R, Brown P F, De Souza P V, et al. A New Algorithm for the Estimation of Hidden Markov Model Parameters. Proc. ICASSP'88, 1988:493～496

26 Bahl L R, Jelinek F. Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition. IEEE Trans. IT,1975,21(2):404～411

27 Bahl L R, Jelinek F, Mercer R L. A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Trans. PAMI,1983,5(1):179～190

28 Baker J K. The DRAGON System —— An Overview. IEEE Trans. ASSP,1975,23(1):24～29

29 Barrett R F, Holdsworth D A. Frequency Tracking Using Hidden Markov Models with Amplitude and Phase Information. IEEE Trans. SP, 1993,41(10):2965～2976

30 Baum L E. An Inequality and Associated Maximization Technique in

Statistical Estimation for Probabilistic Functions of Markov Process. Inequalities,1972,3,1～8

31 Baum L E, Egon J A. An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to a Model for Ecology. Bull. Amer. Meteorol. Soc.,1967,73:360～363

32 Baum L E, Petrie T. Statistical Inference for Probabilistic Function of Finite State Markov Chains. Ann. Math. Stat., 1966,37:1554～1563

33 Baum L E, Petrie T, Soules G, et al. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Processes. Ann. Math. Stat., 1970,41,164～171

34 Beattie V L, Young S J. Noisy Speech Recognition Using Hidden Markov Model State-Based Filtering. Proc. ICASSP'91, 1991:917～920

35 Bellegarda J, Nahamoo D. Tied Mixture Continuous Parameter Models for Large Vocabulary Isolated Speech Recognition. Proc. ICASSP'89, 1989:13～16

36 Bengio Y, Cardin R, De Mori R, et al. A Hybrid Coder for Hidden Markov Models Using a Recurrent Neural Network. Proc. ICASSP'90,1990:537～540

37 Boll S F. Suppression of Acoustic Noisein Speech Using Spectral Subreaction. IEEE Trans. ASSP,1979,27(1):113～120

38 Bossemeyer R W, Wilpon J G, Lee C H, et al. Automatic Speech Recognition of Small Vocabularies with in the Context of Unconstraint Input. JASA,1988,84(Supp. 1)

39 Bourlard H. Neural Nets and Hidden Markov Models: Review and Generalizations. Proc. Eurospeech'91,1991:363～369

40 Bourlard H, Morgan N. A Continuous Speech Recognition System Embedding MLP into HMM. in Advances in Neural Information Processing System (2), Edted by Touretzky D, San Mateo, C A: Morgan Kaufmann,1990,186～193

41 Bourlard H, Wellekens C J. Links between Markov Models and Multilayer Perceptron. IEEE Trans. PAMI,1990,12(12),1167～1178

42 Bourlard H, Wellekens C J. Multilayer Perceptrons and Automatic Speech Recognition. Proc. 1st ICNN, San Diego, CA, USA, 1987, 407～416

43 Bourlard H, Wellekens C J. Speech Pattern Discrimination and Multilayer Perceptrons. Computer Speech and Language, 1989, 3, 1～19

44 Brown P F. The Acoustic-Modeling Problem in Automatic Speech Recognition. Ph.D Dissertation, CMU, USA, 1987

45 Burr D J. Experiments with a Connectionist Text Reader. Proc. 1st ICNN, San Diego, CA, USA, 1987, (IV): 717～724

46 Bush M A, Kopec G E. Network-Based Connected Digit Recognition. IEEE Trans. ASSP, 1987, 35(5): 1401～1413

47 Buzo A, Gray A H, Gray R M, et al. Speech Coding Based upon Vector Quantization. IEEE Trans. ASSP, 1980, 28(5): 562～574

48 Casacuberta F, Vidal E, Mas B, et al. Learning the Structure of HMM's through Grammatical Inference Techniques. Proc. ICASSP'90, 1990, 717～720

49 Chen J H, Gersho A. Real-Time Vector APC Speech Coding at 4800 BPS with Adaptive Post filtering. Proc. ICASSP'87, 1987, 2185～2188

50 Chen X X, Cai C N, Go P, et al. A Hidden Markov Model Applied to Chinese Four-Tone Recognition. Proc. ICASSP'87, 1987, 797～800

51 Chou W, Juang B H, Lee C H. Segmental GPD Training of HMM Based Speech Recognizer. Proc. ICASSP'92, 1992, I: 473～476

52 Chow Y L. Maximum Mutual Information Estimation of HMM Parameters for Continuous Speech Recognition Using the N-Best Algorithm. Proc. ICASSP'90, 1990, 701～704

53 Chow Y L, et al. BYBLOS: the BBN Continuous Speech Recognition System. Proc. ICASSP'87, 1987: 89～92

54 Cohen M, Murveit H, Bernstein J, et al. The DECIPHER Speech Recognition System. Proc. ICASSP'90, 1990, 77～80

55 Cohen P R, Feigenbaum E A. The Handbook of Artificial Intelligence, Vol. I, Chapter V, Understanding Spoken Language, London:

Pitman,1981

56 De Mori R. Computer Models of Speech Using Fuzzy Algorithms. Plenum, USA,1983

57 De Mori R. Planning, Neural Networks and Markov Models for Automatic Speech Recognition. Proc. 9th ICPR, 1988:395~402

58 Deng L, Kenny P, Lennig M, et al. Modeling Acoustic Transitions in Speech by State-Interpolation Hidden Markov Models. IEEE Trans. SP,1992,40(2):265~271

59 Deng L, Lennig M, Seitz F, et al. Large Vocabulary Word Recognition Using Context Dependent Allophonic Hidden Markov Models. Computer Speech and Language, 1990, 4: 345~357

60 Derin H. The Use of Gibbs Distribution in Image Processing. in Communications and Networks: A Survey of Recent Advances, Vol. I, Edted by Black T and Poor V, NY, Springer,1985:266~298

61 Derin H, Elliott H. Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields. IEEE Trans. PAMI, 1987,9(1): 39~55

62 Devijver P A, Dekesel M M. Learning the Parameters of a Hidden Markov Random Field Image Model: a Simple Example. in Pattern Recognition Theory and Applications, Edited by Devijver P A and Kitter J, Springer-Verlag Berl in Heidelberg, 1987:141~163

63 Ephraim Y. Gain-Adapted Hidden Markov Models for Recognition of Clean and Noisy Speech. IEEE Trans. SP, 1992,40(6):1303~1315

64 Ephraim Y. A Bayesian Estimation Approach for Speech Enhancement Using Hidden Markov Models. IEEE Trans. SP,1992,40(4):725~735

65 Ephraim Y, Dembo A, Rabiner L R. A Minimum Discrimination Information Approach for Hidden Markov Modeling. Proc. ICASSP'87,1987:25~28

66 Ephraim Y, Malah D. Speech Enhancement Using a Minimum Mean Square error Short Time Spectral Amplitude Estimator. IEEE Trans. ASSP,1984,32(6):1109~1121

67 Ephraim Y, Malah D, Juang B H. Speech Enhanecement Based upon

Hidden Markov Modeling. Proc. ICASSP'89,1989:353～356

68 Ephraim Y, Malah D, Juang B H. On the Application of Hidden Markov Models for Enhancing Noisy Speech. Proc. ICASSP'88,1988: 533～536

69 Ephraim Y, Malah D, Juang B H. On the Application of Hidden Markov Models for Enhancing Noisy Speech. IEEE Trans. ASSP, 1989,37(12):1846～1856

70 Ephraim Y, Rabiner L R. On the Relations between Modeling Approaches for Information Source. Proc. ICASSP'88,1988:24～27

71 Franco H, Serral heiro A. Training HMMs Using Minimum Recognition Error Approach. Proc. ICASSP'91,1991,357～360

72 Farges E P, Clements M A. Hidden Markov Models Applied to Very Low Bit Rate Speech Coding. Proc. ICASSP'86,1986:433～436

73 Farges E P, Clements M A. An Analysis-Synthesis Hidden Markov Model of Speech. Proc. ICASSP'88,1988:323～326

74 Feng Z Z, Huang Z L, Wan F G. Systolic Neural Network Architecture for Second Order HMM's. Proc. 11th ICPR,1992

75 Fissore L, Laface P, Micca G. Comparison of Discrete and Continuous HMMs in a CSR Task over the Telephone. Proc. ICASSP'91,1991:253～256

76 Frangoulis E, Sgardoni V. A Novel Speaker Adaptation Approach for Continuous Densities HMMs. Proc. ICASSP'91,1991:861～864

77 Franzini M, Lee K F, Weibel A. Connectionist Viterbi Training: a New Hybrid Method for Continuous Speech Recognition. Proc. ICASSP'90,1990,425～428

78 Gao Y Q, Chen Y B, Wu B X. Dynamic Adaptation of HMM for Robust Speech Recognition. Proc. ISCAS'89,1989:1336～1339

79 Gao Y Q, Huang T Y, Chen D W. HMM-Based Warping in Neural Networks. Proc. ICASSP'90, 1990:501～504

80 Geman S, Geman D. Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images. IEEE Trans. PAMI, 1984,6(5), 712～741

81 Giachin E, Rosenberg A E. Word Juncture Modeling Using Phonological Rules for HMM-Based Continuous Speech Recognition. Proc. ICASSP'90,1990:737~740

82 Gorman R P, Sejnowski T J. Learned Classification of Sonar Target Using a Massively Parallel Network. IEEE Trans. ASSP, 1988, 36 (7):1135~1140

83 Gray R M. Vector Quantization. IEEE ASSP Magazine,1984,1(2):4~29

84 Gu H Y, Tseng C Y, Lee L S. Isolated Utterance Speech Recognition Using Hidden Markov Models with Bounded State Durations. IEEE Trans. SP,1991,39(8):1743~1751

85 Guedon Y, Cocozza-Thivent C. Use of the Derin's Algorithm in Hidden Markov Models for Automatic Speech Recognition. Proc. ICASSP'89, 1989:282~285

86 Gupta V N, Lennig M, Mermelstein P. Integration of Acoustic Information in a Large Vocabulary Word Recognizer. Proc. ICASSP'87,1987:697~700

87 Hanazawa T, Kita K, Nakamura S, et al. ATR HMM-LR Continuous Speech Recognition System. Proc. ICASSP'90,1990:53~56

88 Haton J P, Carbonnel N, Fohr D, et al. Interaction between Stochastic Modeling and Knowledge-Based Techniques in Acoustic-Phonetic Decoding of Speech. Proc. ICASSP'87,1987:868~871

89 Hattori H, Nakamura S, Shikano K. Supplementation of HMM for Articulatory Variation in Speaker Adaptation. Proc. ICASSP'90,1990: 153~156

90 He Y. Extended Viterbi Algorithm for Second Order Hidden Markov Process. Proc. 9th ICPR,1988:718~720

91 Higging A L, Wohlford R E. Keyword Recognition Using Template Concatenation. Proc. ICASSP'85,1985:1233~1236

92 Huang X D. Semi-Continuous Hidden Markov Models for Speech Recognition. Ph. D Thesis, Dept. Electrical Engineering, University of Edinburgh,1989

93 Huang X D. Phoneme Classification Using Semicontinuous Hidden Markov Models. IEEE Trans. SP, 1992,40(5):1062～1067

94 Huang X D, Ariki Y, Jack M A. Hidden Markov Models for Speech Recognition. Edinburgh University Press,1990

95 Huang X D, Jack M A. Unified Modeling of Vector Quantization and Hidden Markov Model Using Semi-Continuous Hidden Markov Models. Proc. ICASSP'89,1989:639～642

96 Huang X D, Jack M A. Semi-Continuous Hidden Markov Models for Speech Signals. Computer Speech and Language, 1989,3,239～251

97 Huang X D, Jack M A. Hidden Markov Modeling of Speech Based on a Semi-Continuous Model. IEE Electronics Letters, 1988,24(1):6～7

98 Huang X D, Jack M A, Ariki Y. Parameter Reestimation of Semi-Continuous Hidden Markov Models with Feedback to Vector Quantization Codebook. IEE Electronics Letters,1988,24(22):1375～1377

99 Huang E F, Soong F K. A Probabilistic Acoustic Map Based Discriminative HMM Training. Proc. ICASSP'90,1990,693～696

100 Hwang J N, Kung S Y. A Unifying Viewpoint of Multilayer Perceptrons and Hidden Markov Models. Proc. ISCAS'89,1989:770～773

101 Hwang J N, Vlontzos J A, Kung S Y. A Systolic Neural Network Architecture for Hidden Markov Models. IEEE Trans. ASSP, 1989, 37(12):1967～1979

102 Hwang M Y, Hon H W, Lee K F. Modeling Between-Word Coarticulation in Continuous Speech Recognition. Proc. Eurospeech'89,1989:5～8

103 Itakura F. Minimum Predictive Residual Principle Applied to Speech Recognition. IEEE Trans. ASSP,1975,23(1),67～72

104 Itou K, Hayamizu S, Tanaka H. Continuous Speech Recognition by Context-Dependent Phonetic HMM and an Efficient Algorithm for Finding N-Best Sentence Hypotheses. Proc. ICASSP'92,1992,I:21～24

105 Jelinek F. Continuous Speech Recognition by Statistical Methods. Proc. of the IEEE,1976,64(4):532～536

106 Jelinek F, Bahl L R, Mercer R L. Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech. IEEE Trans. IT, 1975,21(2):250～256

107 Jelinek F, Mercer R L. Interpolated Estimation of Markov Source Parameters from Sparse Data. in Pattern Recognition in Practice, Edited by Gelsema E S and Kanal L N, North Holl and Publishing Co., Amsterdam,1980,381～402

108 Juang B H. Maximum Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains. AT & T Bell Lab. Tech. J., 1985,64(6):1235～1249

109 Juang B H. On the Hidden Markov Model and Dynamic Time Warping for Speech Recognition——A Unified View. AT & T Bell Lab. Tech. J., 1984,63(9):1213～1243

110 Juang B H, Levinson S E, Sondhi M M. Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains. IEEE Trans. IT,1986,32(2):307～309

111 Juang B H, Paliwal K K. Hidden Markov Models with First-Order Equalization for Noisy Speech Recognition. IEEE Trans. SP,1992,40 (9):2136～2143

112 Juang B H, Rabiner L R. A Probabilitic Distance Measure for Hidden Markov Models. AT & T Bell Lab. Tech. J., 1985,64(2):391～408

113 Juang B H, Rabiner L R. Mixture Autoregressive Hidden Markov Models for Speech Signals. IEEE Trans. ASSP, 1985,33(6):1404～1413

114 Juang B H, Rabiner L R. The Segmental k-Means Algorithm for Estimating Parameters of Hidden Markov Models. IEEE Trans. ASSP, 1990,38(9):1639～1641

115 Kawahara T, Doshita S. HMM Basedon Pair-Wise Bayes Classifiers. Proc. ICASSP'92, 1992,I:365～368

116 Kenny P, Lennig M, Mermelstein P. A Linear Predictive HMM for

Vector-Valued Observations with Applications to Speech Recognition. IEEE Trans. ASSP,1990,38(2):220～225

117 Kita K, Kawabata T, Saito H. HMM Continuous Speech Recognition Using Predictive LR Parsing. Proc. ICASSP'89, 1989:703～706

118 Klatt D H. Overview of the ARPA Speech Understanding Project,in Trends in Speech Recognition. Edited by Lea W A. Prentice-HallInc., Englewood Cliffs,USA,1980:249～271

119 Kohnen T. An Introduction to Neural Computing. Neural Networks, 1988,1(1):3～16

120 Koo J M, Lee H S, Un C K. An Improved VQ Codebook Design Algorithm for HMM. Proc. ICASSP'92,1992,I:357～360

121 Kriouile A, Mari J F, Haton J P. Some Improvements in Speech Recognition Algorithm Based on HMM. Proc. ICASSP'90,1990:545～548

122 Krishnamurthy A K, Ahalt S C, Melton D E, et al. Neural Networks for Vector Quantization of Speech and Images. IEEE J. Select. Areas Commun.,1990,8(8):1449～1457

123 Kundu A, He Y,Bahl P. Recognition of Handwritten Word: First and Second Order Hidden Markov Model Based Approach. Pattern Recognition,1989,22(3):283～297

124 Kung H T. Why Systolic Architectures. IEEE Computer,1982,15 (1):37～46

125 Kung H T, Leiserson C E. Systolic Arrays (for VLSI). in Sparse Matrix Symposium. SIAM,1978:256～282

126 Kung S Y. VLSI Array Processors, Englewood Cliffs, N J: Prentice-Hall,1987

127 Kung S Y, Arun K S, Gal-Ezer R J, et al. Wavefront Array Processors:Language, Architecture, and Applications. IEEE Trans. C,1982,31(11):1054～1066

128 Kung S Y, Hwang J N, Paralell Architectures for Artificial Neural Nets. Proc. ICNN'88,1988,2:165～172

129 Kung S Y, Hwang J N. A Unified Systolic Architecture for Artificial

Neural Networks. J. Paralell Distributed Comput., Special Issue on Neural Networks, 1989, 6(2): 358～387

130 Lee C H, Lin C H, Juang B H. A Study on Speaker Adaptation of Continuous Density HMM Parameters. Proc. ICASSP'90, 1990: 145～148

131 Lee C H, Lin C H, Juang B H. A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. IEEE Trans. SP, 1991, 39(4): 806～814

132 Lee K F. Large Vocabulary Speaker Independent Continuous Speech Recognition: the SPHINX System. Ph. D Dissertation, Computer Science Dept., CMU, USA, 1988

133 Lee K F. Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition. IEEE Trans. ASSP, 1990, 38(4): 599～609

134 Lee K F, Hon H W. Large Vocabulary Speaker Independent Continuous Speech Recognition Using HMM. Proc. ICASSP'88, 1988, 123～126

135 Lee K F, Hon H W. Speaker-Independent Phone Recognition Using Hidden Markov Models. IEEE Trans. ASSP, 1989, 37(11): 1641～1648

136 Lee K F, Hon H W, Hwang M Y. Speech Recognition Using Hidden Markov Models: a CMU Perspective. Speech Communication, 1990, 9(5/6): 497～508

137 Lee K F, Hon H W, Hwang M Y, et al. Recent Progress and Future Outlook of the SPHINX Speech Recognition System. Computer Speech and Languang, 1990, 4: 57～69

138 Lee K F, Hon H W, Reddy R. An Overview of the SPHINX Speech Recognition System. IEEE Trans. ASSP, 1 990, 38(1): 35～45

139 Lee L S, Tseng C Y, Gu H Y, et al. A Real-Time Mandarin Dictation Machine for Chinese Language with Unlimited Texts and Very Large Vocabulary. Proc. ICASSP'90, 1990: 65～68

140 Leuven K U. TDNN Labeling for a HMM Recognizer. Proc.

ICASSP'90,1990,421～423

141 Levin E. Word Recognition Using Hidden Control Neural Network Architecture. Proc. ICASSP'90,1990,433～436

142 Levinson S E. Continuously Variable Duration Hidden Markov Models for Speech Recognition. Computer Speech and Language,1986,1:29～46

143 Levinson S E, Ljolje A, Miller L G. Large Vocabulary Speech Recognition Using a Hidden Markov Model for Acoustic/Phonetic Classification. Proc. ICASSP'88,1988,505～508

144 Levinson S E, Rabiner L R, Sondhi M M. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. BSTJ, 1983,62(4):1035～1074

145 Lim J S, Oppenheim A V. All-Pole Modeling of Degraded Speech. IEEE Trans. ASSP, 1978,26(3):197～210

146 Lim J S, Oppenheim A V. Enhancement and Bandwidth Compression of Noisy Speech. Proc. of the IEEE, 1979, 67(12):1586～1604

147 Lim J S, Oppenheim A V, Braida L D. Evaluation of an Adaptive Comb Filtering Method for Enhancing Speech Degraded by White Noise Addition. IEEE Trans. ASSP, 26(4):354～358

148 Linde Y, Buzo A, Gray R M. An Algorithm for Vector Quantizer Design. IEEE Trans. COM,1980,28(1):84～95

149 Liporace L A. Maximum Likelihood Estimation for Multivariate Observation of Markov Sources. IEEE Trans. IT,1982,28(5):729～734

150 Lippmann R P. An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, 1987,4(2):4～22

151 Lippmann R P. Neural Nets for Computing. Proc. ICASSP'88,1988:1～6

152 Ljolje A, Ephraim Y, Rabiner L R. Estimation of Hidden Markov Model Parameters by Minimizing Empirical Error Rate. Proc. ICASSP'90,1990:709～712

153 Ljolje A, Fallside F. Synthesis of Natural Sounding Pitch Contours in Isolated Utterances Using Hidden Markov Models. IEEE Trans. ASSP,1986,34(5):1074～1079

154 Ljolje A, Levinson S E. Development of an Acoustic/Phonetic Hidden Markov Model for Continuous Speech Recognition. IEEE Trans. SP, 1991,39(1):29～39

155 Lockwood P, Boudy J, Blanchet M. Non-Linear Spectral Subtraction (NSS) and Hidden Markov Models for Robust Speech Recognition in Car Noise Environments. Proc. ICASSP'92,1992,I:265～268

156 Makhoul J, Roucos S. Vector Quantization in Speech Recognition. Proc. of the IEEE, 1985,73(11):1551～1588

157 Mansour D, Juang B H. A Family of Distortion Measure Based upon Projection Operation for Robust Speech Recognition. IEEE Trans. ASSP, 1989, 37(11):1659～1671

158 Mariani J. Recent Advances in Speech Recognition. Proc. ICASSP'89,1989:429～440

159 Markel J D, Gray A H, Jr. Linear Predictive of Speech. Springer-Verlag, NY, USA,1976. (有中译本)

160 Martin E A, Lippmann R P, Paul D B. Dynamic Adaptation of Hidden Markov Models for Robust Isolated Word Speech Recognition. Proc. ICASSP'88,1988:52～54

161 Mathan L, Miclet L. Rejection of Extraneous Input in Speech Recognition Applications, Using Multi-layer Perceptrons and the Trace of HMMs. Proc. ICASSP'91,1991:93～96

162 Merhav N, Ephraim Y. Hidden Markov Modeling Using the Most Likely State Sequence. Proc. ICASSP'91,1991:469～472

163 Merialdo B. Phonetic Recognition Using Hidden Markov Models and Maximum Mutual Information Training. Proc. ICASSP'88,1988:111～114

164 Minsky M, Papert S. Perceptron. Cambridge, M A, USA, MIT Press,1969

165 Morgan N, Bourlard H. Continuous Speech Recognition Using

Multilayer Perceptrons with Hidden Markov Models. *Proc.* ICASSP'90,1990,413～416

166 Morgan N, Bourlard H, Wooters C, et al. Phonetic Context in Hybrid HMM/MLP Continuous Speech Recognition. Proc. Eurospeech'91, 1991:109～111

167 Morgan N, Hermansky H, Bourlard H, et al. Continuous Speech Recognition Using PLP Analysis with Multilayer Perceptrons. Proc. ICASSP'91,1991,49～52

168 Murveit H, Weintraub M. 1000-Word Speaker Independent Continuous Speech Recognition Using Hidden Markov Models. Proc. ICASSP'88,1988,115～118

169 Myers C S, Rabiner L R, Rosenberg A E. An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Word Recognition. Proc. ICASSP'80, 1980, 173～177

170 Nadas A, et al. On a Model-Robust Training Method for Speech Recognition. IEEE Trans. ASSP, 1988,36(9),1432～1436

171 Nakamura S, Shikano K. Speaker Adaptation Applied to HMM and Neural Networks. Proc. ICASSP'89,1989,89～92

172 Ney H, Paeseler A. Phoneme-Based Continuous Speech Recognition Results for Different Language Models in the 1 000-Word SPICOS System. Speech Communication,1988,7(4):367～374

173 Ney H, Noll A. Phonetic Modeling Using Continuous Mixture Densities. Proc. ICASSP'88, 1988,437～440

174 Niles L T, Silverman H F. Combining Hidden Markov Models and Neural Network Classifiers. Proc. ICASSP'90, 1990,417～420

175 Nishimura M, Sugawara K. Speaker Adaptation Method for HMM-Based Speech Recognition. Proc. ICASSP'88,1988,207～210

176 Nitta T, Iwasaki J, Masai Y, et al. Representing Dynamic Features of Phonetic Segment in an Orthogonalized Codebook of HMM Based Speech Recognition System. Proc. ICASSP'92,1992,I:385～388

177 Normandin Y, Morgan S D. An Improved MMIE Training Algorithm for Speaker-Independent, Small Vocabulary, Continuous Speech

Recognition. Proc. ICASSP'91,1991,537～540

178 Ostendorf M, Rohlicek J R. Joint Quantizer Design and Parameter Estimation for Discrete Hidden Markov Models. Proc. ICASSP'90, 1990:705～708

179 Parsons T W. Voice and Speech Processing. Mc Graw-Hill Book Company,1986(有中译本)

180 Paul D B. The Lincoln Tied-Mixture HMM Continuous Speech Recognizer. Proc. ICASSP'91,1991:329～332

181 Peeling S M, Moore R K. Isolated Digit Recognition Experiments Using Multi-Layer Perceptron. Speech Communication, 1988,7(3): 403～409

182 Pepper D J, Barnwell T P, Clements M A. Using a Ring Paralell Processor for Hidden Markov Model Training. IEEE Trans. ASSP, 1990,38(2):366～369

183 Pepper D J, Clements M A. Phonemic Recognition Usinga Large Hidden Markov Model. IEEE Trans. SP, 1992,40(6):1590～1595

184 Pepper D J, Clements M A. On the Phonetic Structure of a Large Hidden Markov Model. Proc. ICASSP'91,1991:465～468

185 Picone J. Continuous Speech Recognition Using Hidden Markov Models. IEEE ASSP Magazine, 1990,7(3):26～41

186 Plannerer B, Ruske G. Recognition of Demisyllable Based Units Using Semicontinuous Hidden Markov Models. Proc. ICASSP'92, 1992, I: 581～584

187 Poritz A B. Linear Predictive Hidden Markov Models and the Speech Signal. Proc. ICASSP'82, 1982:1291～1294

188 Poritz A B. Hidden Markov Models, A Guided Tour. Proc. ICASSP'88,1988:7～13

189 Rabiner L R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the IEEE, 1989,77(2): 257～285

190 Rabiner L R, Bergh A, Wilpon J G. An Improved Training Procedure for Connected Digit Recognition. BSTJ,1982,61(6):981～1001

191 Rabiner L R, Juang B H. An Introduction to Hidden Markov Models. IEEE ASSP Magazine, 1986, 3(1):4～16

192 Rabiner L R, Juang B H, Levinson S E, et al. Recognition of Isolated Digit Using Hidden Markov Models with Continuous Mixture Densities. AT & T BellLab. Tech. J., 1985, 64(6):1211～1233

193 Rabiner L R, Juang B H, Levinson S E, et al. Some Properties of Continuous Hidden Markov Model Representations. AT & T Bell Lab. Tech. J., 1985, 64(6):1251～1270

194 Rabiner L R, Levinson S E. Isolated and Connected Word Recognition——Theory and Selected Applications. IEEE Trans. COM, 1981, 29(5):621～659

195 Rabiner L R, Levinson S E. A Speaker Independent, Syntax-Directed, Connected Word Recognition System Basedon Hidden Markov Models and Level Building. IEEE Trans. ASSP, 1985, 33(3):561～573

196 Rabiner L R, Levinson S E, Sondhi M M. On the Application of Vector Quantization and Hidden Markov Models to Speaker Independent, Isolated Word Recognition. BSTJ, 1983, 62(4):1075～1105

197 Rabiner L R, Levinson S E, Sondhi M M. On the Use of Hidden Markov Models for Speaker Independent Recognition of Isolated Word from a Medium-Size Vocabulary. AT & T Bell Lab. Tech. J., 1984, 63(4), 627～642

198 Rabiner L R, Schafer R W. Digital Processing of Speech Signals. Prentice-Hall, Inc., USA, 1978. (有中译本)

199 Rabiner L R, Wilpon J G, Soong F K. High Performance Connected Digit Recognition Using Hidden Markov Models. IEEE Trans. ASSP, 1989, 37(8):1214～1225

200 Rabiner L R, Wilpon J G, Juang B H. A Segmental k-Means Training Procedure for Connected Word Recognition Based on Whole Word Reference Patterns. AT & T Bell Lab. Tech. J., 1986, 65(3):21～31

201 Rabiner L R, Wilpon J G, Juang B H. A Model-Based Connected-

Digit Recognition System Using either Hidden Markov Modelsor Templates. Computer Speech and Language, 1986,1:167～197

202 Rabiner L R, Wilpon J G, Quinn A M, et al. On the Application of Embedded Digit Training to Speaker Independent, Connected Digit Recognition. IEEE Trans. ASSP,1984,32(2):272～280

203 Ramesh P, Wilpon J G. Modeling State Durationin Hidden Markov Models for Automatic Speech Recognition. Proc. ICASSP'92,1992,I: 381～384

204 Rigoll G. Information Theory-Based Supervised Learning Methods for Self-Organizing Maps in Combination with Hidden Markov Modeling. Proc. ICASSP'91,1991:65～68

205 Rose R C, Paul D B. A Hidden Markov Model Based Keyword Recognition System. Proc. ICASSP'90,1990,129～132

206 Rumelhart D E, McClelland J L. Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Vol 1: Foundations. MIT Press,1986

207 Russell M J, Moore R K. Explicit Modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition. Proc. ICASSP'85,1985:5～8

208 Russell M J, Ponting K M, Peeling S M, et al. The ARM Continuous Speech Recognition System. Proc. ICASSP'90, 1990:69～72

209 Sakoe H, Chiba S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Trans. ASSP,1978,26(1):43～49

210 Schroeder M R, At al B S. Code-Excited Linear Predictive (CELP): High Quality Speech at Very Low Bit Rates. Proc. ICASSP'85,1985, 937～940

211 Schwartz R, et al. Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech. Proc. ICASSP'85, 1985: 1205～1208

212 Sejnowski T J, Rosenberg C R. Parallel Networks that Learn to Pronounce English Text. Complex System, 1987,1(1):145～168

213 Shikano K, Lee K F, Reddy R. Speaker Adaptation through Vector

Quantization. Proc. ICASSP'86,1986,2643～2646

214 Srinath M D, Rajasekaran P K. An Introduction to Statistical Signal Processing with Application. John & Sons,1979.(有中译本)

215 Steinbiss V, Noll A, Paeseler A, et al. A 10 000-Word Continuous Speech Recognition System. Proc. ICASSP'90,1990,57～60

216 Sugawara K, Nishimura M, Kuroda A. Speaker Adaptation for a Hidden Markov Models. Proc. ICASSP'86,1986,2667～2670

217 Tamura S, Waibel A. Noise Reduction Using Connectionist models. Proc. ICASSP'88,1988,553～556

218 Tishby N Z. On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition. IEEE Trans. SP, 1991,39(3):563～569

219 The World Standard Signal Processing Software, Interactive Laboratory System(ILS) Application Note #1: Speech Analysis and Synthesis,Signal Technology, Inc.,(STI),1983,USA.(国家模式识别实验室内部资料)

220 Tohkura Y. A Weighted Cepstral Distance Measure for Speech Recognition. IEEE Trans. ASSP,1987,35(5):1414～1422

221 Vittorelli V, et al. Polyglot: Multilingual Speech Recognition and Synthesis. Proc. Inte. Conf. Spoken Language Processing, Kobe, Japan,Nov. 1990

222 Waibel A, Hanazawa T, Hinton G, et al. Phoneme Recognition Using Time-Delay Neural Networks. IEEE Trans. ASSP,1989,37(3):328～339

223 Wang M Q, Young S J. Speech Recognition Using Hidden Markov Model Decomposition and a General Background Speech Model. Proc. ICASSP'92,1992,I:253～256

224 Watrous R L, Shastri L. Learning Phonetic Features Using Connectionist Networks, an Experiment in Speech Recognition. Proc. 1st ICNN, SanDiego,USA,1987,(IV):381～388

225 Wellekens C J. Explicit Time Correlationin Hidden Markov Models for Speech Recognition. Proc. ICASSP'87,1987:384～386

226 White L B. Cartesian Hidden Markov Models with Applications. IEEE Trans. SP, 1992,40(6):1601～1604

227 Willsky A S. Digital Signal Processing and Control and Estimation Theory. The MIT Press,1979. (有中译本)

228 Wilpon J G, Rabiner L R, Lee C L. Automatic Recognition of Keywords in Unconstrainted Speech Using Hidden Markov Models. IEEE Trans. ASSP,1990,38(11):1870～1878

229 Wilpon J G, Miller L G, Modi P. Improvements and Applications for Key Word Recognition Using Hidden Markov Modeling Techniques. Proc. ICASSP'91,1991:309～312

230 Woodard J P. Modeling and Classification of Natural Sounds by Product Code Hidden Markov Models. IEEE Trans. SP, 1992, 40 (7):1833～1835

231 Woodland P C, Cole D R. Optimising Hidden Markov Models Using Discriminative Output Distributions. Proc. ICASSP'91, 1991: 545～548

232 Yang W J, Lee J C, Chang Y C, et al. Hidden Markov Model for Mandarin Tone Recognition. IEEE Trans. ASSP, 1988,36(7):988～992

233 Yeh H G. Adaptive Noise Cancellation for Speech with a TMS32020. Proc. ICASSP'87,1987:1171～1174

234 Young S J. Competitive Training in Hidden Markov Models. Proc. ICASSP'90,1990:681～684

235 Zhao Y, Atlas L, Zhuang X. Application of the Gibbs Diatribution to Hidden Markov Modeling in Isolated Word Recognition. Proc. ICASSP'88,1988:28～31

236 Zhao Y, Atlas L, Zhuang X. Application of the Gibbs Distribution to Hidden Markov Modeling in Speaker Independent Isolated Word Recognition. IEEE Trans. SP, 1991,39(6):1291～1299

237 谢锦辉. 线性预测 HMM 在语音识别、压缩和增强中的应用:〔博士学位论文〕. 武汉:华中理工大学,1990

238 李晖. HUST——汉语大词汇量连续语音识别系统:〔硕士学位论文〕.

武汉：华中理工大学，1994

239 祝永进. TMS320C25 语音录放系统，〔学士学位论文〕. 武汉：华中理工大学，1994

240 毛丰山. 汉语拼音-汉字转换系统，〔学士学位论文〕. 武汉：华中理工大学，1994

241 谢锦辉，黄载禄，万发贲. 一种新颖的有色噪声白化方法. 电子学报，1990，18(5)：109～111

242 谢锦辉，高雨青. 关于 HMM 相对可靠性量度. 自动化学报，1993，19(5)：637～640

243 谢锦辉. 采用上下文相关音素 HMM 的连续语音识别. 通信学报，1994，15(2)：83～87

244 谢锦辉，潘小兵. 连续语音识别系统性能评估软件. 计算机应用与软件，1994，11(2)：20～25

245 谢锦辉，黄载禄，万发贲. 基于线性均方误差估值的 LPC 距离量度. 华中理工大学学报，1990，18(4)：67～70

246 谢锦辉，黄载禄，万发贲. 有噪语音的线性预测 HMM 参数估计方法. 华中理工大学学报，1991，19(增刊 I)：129～133

247 谢锦辉，黄载禄，万发贲. 构造 HMM 观察值概率函数的一种方法. 华中理工大学学报，1992，20(1)：47～51

248 王舟，谢锦辉. 非特定人普通话孤立数字语音识别系统. 华中理工大学学报，1994，22(11)：36～39

249 李晖，谢锦辉，黄载禄. 基于 LPC 预测误差 DWT 的语音基音检测. 华中理工大学学报，1994，22(11)：40～44

250 高雨青，陈永彬，吴伯修等. 语音识别的 Robust 性及隐 Markov 模型的自适应和自学习. 第七届全国模式识别与机器智能学术会议论文集，武汉，1989：4. 34～4. 35

251 谢锦辉，高雨青，黄载禄等. 隐 Markov 模型的一种新的观察值概率函数. 第七届全国模式识别与机器智能学术会议论文集，武汉：1989：2. 117～2. 121

252 谢锦辉，黄载禄，万发贲. 线性预测隐 Markov 模型对语音特征的压缩作用及其在语音分析-综合中的应用. 第七届全国模式识别与机器智能学术会议论文集，武汉：1989：4. 29～4. 33

253 谢锦辉，黄载禄，万发贯. 基于 HMM 语音识别算法的 Systolic 结构. 第八届电路与系统年会论文集(第三分册)，成都：1989：45～50

254 高雨青，谢锦辉，陈永彬等. 连续参数隐 Markov 模型的一种新的训练方法. 第八届电路与系统年会论文集(第四分册)，成都：1989：118～121

255 黄载禄，谢锦辉. 关于语音处理中隐 Markov 模型与神经网络的联系. 1990 中国首届神经网络学术会议论文集，北京：1990：218

256 谢锦辉，李晖. 基于小波变换的语音信号分解和结构分析. 计算机智能接口与智能应用'93 会议论文集，黑龙江镜泊湖：1993：239～246

257 谢锦辉，李晖. 基于混合 SOFM/HMM 的语音特征矢量量化. 见：黄载禄等编. 神经网络理论及应用——'94 最新进度. 武汉：华中理工大学出版社，1994. 330～333

258 谢锦辉，李晖. 大型线性预测 HMM 及其在汉语音素识别中的应用. 见：杨家沅编. 语音识别与合成. 成都：四川科学技术出版社，1994. 169～174

259 谢锦辉，黄瑞光，黄正强等. HMM 及其在语音处理中的应用. 电子学科跨世纪优秀人才学术讨论会论文集，北京：1994：95～98

260 谢锦辉，李晖. 基于 Mallat 算法的语音共振峰提取. (待发表)

261 XIE Jin Hui. A Study on an HMM Relative Reliability Measure with Applications to Continuous Speech Recognition. Notes et Documents LIMSI-CNRS No. 91-16, FRANCE, Dec. 1991.

262 XIE Jin Hui, HUANG Zai Lu, WAN Fa Guan. On the Split Lattice Algorithm and the Line Spectral Pair (LSP) Coefficients. Proc. Inter. Conf. on Communication Systems, Singapore, 1988.

263 XIE Jin Hui, HUANG Zai Lu, WAN Fa Guan. A Speech Analysis-Synthesis System Based on Linear Predictive Hidden Markov Models. Proc. Inter. Conf. on Circuits and Systems, Nanjing, China, 1989: 517～520

264 XIE Jin Hui, GAO Yu Qing, TU Jian Hua. Speaker Adaptation Methods for Speech Recognition Systems Based on Linear Predictive Hidden Markov Models. Proc. Inter. Conf. on Computer Processing of Chinese and Oriental Language, Changsha China, 1990

265 HUANG Zai Lu, XIE Jin Hui, WAN Fa Guan. A New HMM-Based

Algorithm for Enhancing Noisy Speech. Proc. Inter. Conf . on Signal Processing, Beijing, 1990, 389～392

266 GAO Yu Qing, XIE Jin Hui. A Model Block-Training Method for HMM-Based Speech Recognition Systems. Proc. ICASSP'90, 1990: 541～544