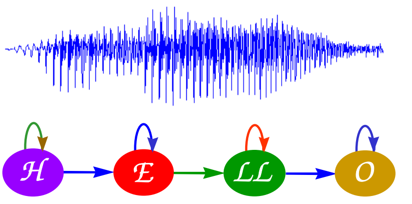
Advanced Digital Signal Processing and Noise Reduction, Second Edition.
Saeed V. Vaseghi

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

5



HIDDEN MARKOV MODELS

Hidden Markov models (HMMs) are used for the statistical modelling of non-stationary signal processes such as speech signals, image sequences and time-varying noise. An HMM models the time variations (and/or the space variations) of the statistics of a random process with a Markovian chain of state-dependent stationary subprocesses. An HMM is essentially a Bayesian finite state process, with a Markovian prior for modelling the transitions between the states, and a set of state probability density functions for modelling the random variations of the signal process within each state. This chapter begins with a brief introduction to continuous and finite state non-stationary models, before concentrating on the theory and applications of hidden Markov models. We study the various HMM structures, the Baum–Welch method for the maximum-likelihood training of the parameters of an HMM, and the use of HMMs and the Viterbi decoding algorithm for the classification and decoding of an unlabelled observation signal sequence. Finally, applications of the HMMs for the enhancement of noisy signals are considered.

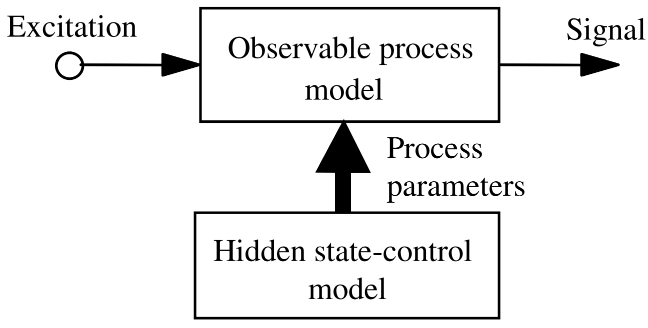


Figure 5.1 Illustration of a two-layered model of a non-stationary process.

5.1 Statistical Models for Non-Stationary Processes

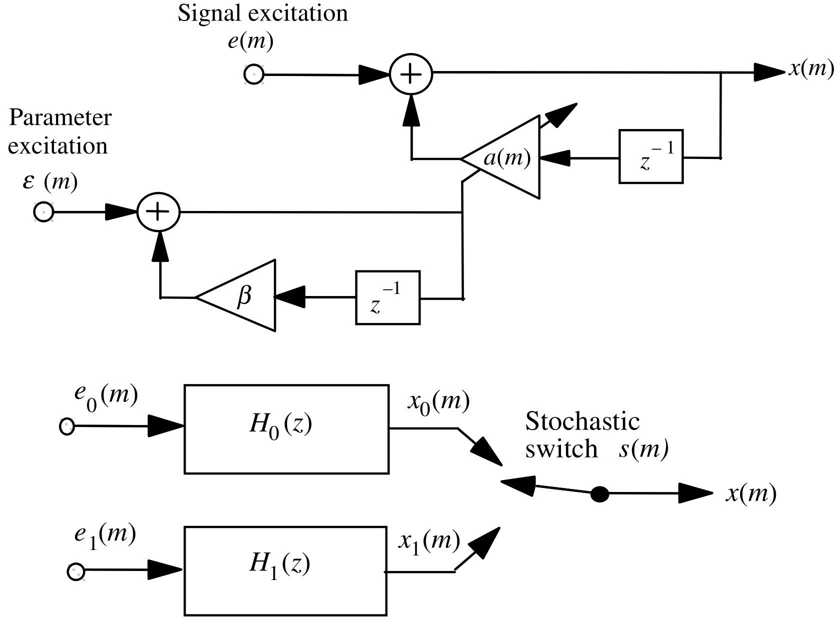
A non-stationary process can be defined as one whose statistical parameters vary over time. Most "naturally generated" signals, such as audio signals, image signals, biomedical signals and seismic signals, are non-stationary, in that the parameters of the systems that generate the signals, and the environments in which the signals propagate, change with time.

A non-stationary process can be modelled as a double-layered stochastic process, with a hidden process that controls the time variations of the statistics of an observable process, as illustrated in Figure 5.1. In general, non-stationary processes can be classified into one of two broad categories:

(a) *Continuously variable state* processes.
(b) *Finite state* processes.

A continuously variable state process is defined as one whose underlying statistics vary continuously with time. Examples of this class of random processes are audio signals such as speech and music, whose power and spectral composition vary continuously with time. A finite state process is one whose statistical characteristics can *switch* between a finite number of stationary or non-stationary states. For example, impulsive noise is a binary-state process. Continuously variable processes can be approximated by an appropriate finite state process.

Figure 5.2(a) illustrates a non-stationary first-order autoregressive (AR) process. This process is modelled as the combination of a *hidden* stationary AR model of the signal parameters, and an observable time-varying AR model of the signal. The hidden model controls the time variations of the



(a)

(b)

Figure 5.2 (a) A continuously variable state AR process. (b) A binary-state AR process.

parameters of the non-stationary AR model. For this model, the observation signal equation and the parameter state equation can be expressed as

$$x(m) = a(m)x(m-1) + e(m) \qquad \text{Observation equation} \qquad (5.1)$$

$$a(m) = \beta a(m-1) + \varepsilon(m) \qquad \text{Hidden state equation} \qquad (5.2)$$

where $a(m)$ is the time-varying coefficient of the observable AR process and β is the coefficient of the hidden state-control process.

A simple example of a finite state non-stationary model is the binary-state autoregressive process illustrated in Figure 5.2(b), where at each time instant a random switch selects one of the two AR models for connection to the output terminal. For this model, the output signal $x(m)$ can be expressed as

$$x(m) = \bar{s}(m)x_0(m) + s(m)x_1(m) \qquad (5.3)$$

where the binary switch $s(m)$ selects the state of the process at time m, and $\bar{s}(m)$ denotes the Boolean complement of $s(m)$.

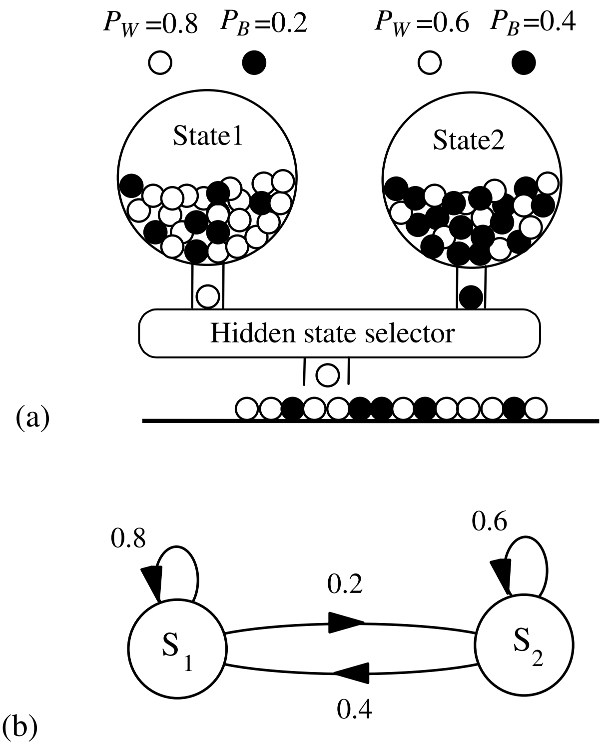


Figure 5.3 (a) Illustration of a two-layered random process. (b) An HMM model of the process in (a).

5.2 Hidden Markov Models

A hidden Markov model (HMM) is a double-layered finite state process, with a hidden Markovian process that controls the selection of the states of an observable process. As a simple illustration of a binary-state Markovian process, consider Figure 5.3, which shows two containers of different mixtures of black and white balls. The probability of the black and the white balls in each container, denoted as P_B and P_W respectively, are as shown above Figure 5.3. Assume that at successive time intervals a hidden selection process selects one of the two containers to release a ball. The balls released are replaced so that the mixture density of the black and the white balls in each container remains unaffected. Each container can be considered as an underlying state of the output process. Now for an example assume that the hidden container-selection process is governed by the following rule: at any time, if the output from the currently selected container is a white ball then the same container is selected to output the next ball, otherwise the other container is selected. This is an example of a Markovian process because the next state of the process depends on the current state as shown in the binary state model of Figure 5.3(b). Note that in this example the observable outcome does not unambiguously indicate the underlying hidden state, because both states are capable of releasing black and white balls.

In general, a hidden Markov model has N sates, with each state trained to model a distinct segment of a signal process. A hidden Markov model can be used to model a time-varying random process as a probabilistic Markovian chain of N stationary, or quasi-stationary, elementary sub-processes. A general form of a three-state HMM is shown in Figure 5.4. This structure is known as an *ergodic* HMM. In the context of an HMM, the term "ergodic" implies that there are no structural constraints for connecting any state to any other state.

A more constrained form of an HMM is the left–right model of Figure 5.5, so-called because the allowed state transitions are those from a left state to a right state and the self-loop transitions. The left–right constraint is useful for the characterisation of temporal or sequential structures of stochastic signals such as speech and musical signals, because time may be visualised as having a direction from left to right.

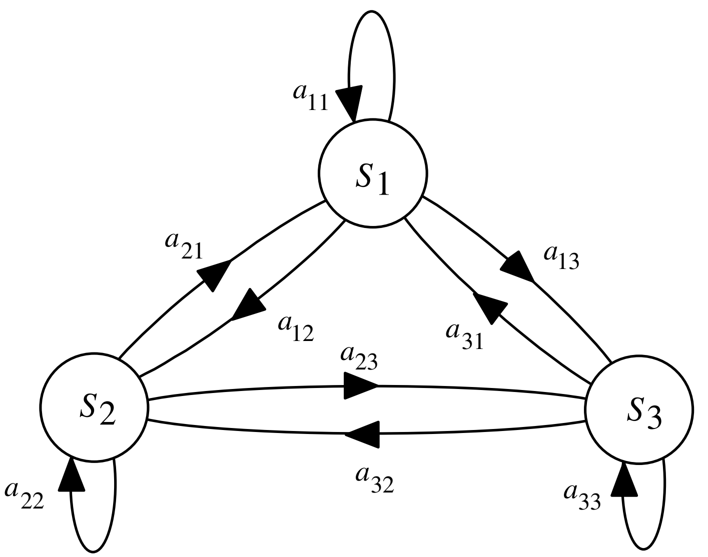


Figure 5.4 A three-state ergodic HMM structure.

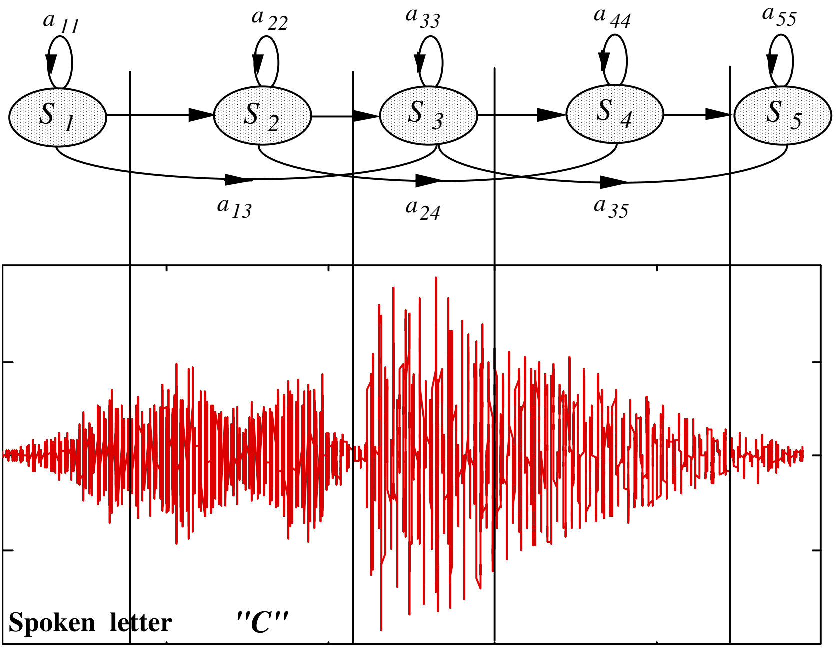


Figure 5.5 A 5-state left–right HMM speech model.

5.2.1 A Physical Interpretation of Hidden Markov Models

For a physical interpretation of the use of HMMs in modelling a signal process, consider the illustration of Figure 5.5 which shows a left–right HMM of a spoken letter "C", phonetically transcribed as 's-iy', together with a plot of the speech signal waveform for "C". In general, there are two main types of variation in speech and other stochastic signals: variations in the spectral composition, and variations in the time-scale or the articulation rate. In a hidden Markov model, these variations are modelled by the state observation and the state transition probabilities. A useful way of interpreting and using HMMs is to consider each state of an HMM as a model of a segment of a stochastic process. For example, in Figure 5.5, state S_1 models the first segment of the spoken letter "C", state S_2 models the second segment, and so on. Each state must have a mechanism to accommodate the random variations in different realisations of the segments that it models. The state transition probabilities provide a mechanism for connection of various states, and for the modelling the variations in the duration and time-scales of the signals in each state. For example if a segment of a speech utterance is elongated, owing, say, to slow articulation, then this can be accommodated by more self-loop transitions into the state that models the segment. Conversely, if a segment of a word is omitted, owing, say, to fast speaking, then the skip-next-state connection accommodates that situation. The state observation pdfs model the probability distributions of the spectral composition of the signal segments associated with each state.

5.2.2 Hidden Markov Model as a Bayesian Model

A hidden Markov model $\mathcal{M}$ is a Bayesian structure with a Markovian state transition probability and a state observation likelihood that can be either a discrete pmf or a continuous pdf. The *posterior* pmf of a state sequence $\boldsymbol{s}$ of a model $\mathcal{M}$, given an observation sequence $\boldsymbol{X}$, can be expressed using Bayes' rule as the product of a state *prior* pmf and an observation *likelihood* function:

$$P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(\boldsymbol{s}|\boldsymbol{X},\mathcal{M})=\frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})}P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M}) \tag{5.4}$$

where the observation sequence $\boldsymbol{X}$ is modelled by a probability density function $P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(\boldsymbol{s}|\boldsymbol{X},\mathcal{M})$.

The posterior probability that an observation signal sequence $\boldsymbol{X}$ was generated by the model $\mathcal{M}$ is summed over all likely state sequences, and may also be weighted by the model prior $P_{\mathcal{M}}(\mathcal{M})$:

$$P_{\mathcal{M}|\boldsymbol{X}}(\mathcal{M}|\boldsymbol{X})=\frac{1}{f_{\boldsymbol{X}}(\boldsymbol{X})}\underbrace{P_{\mathcal{M}}(\mathcal{M})}_{\textit{Model prior}}\sum_{\boldsymbol{s}}\underbrace{P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})}_{\textit{State prior}}\;\underbrace{f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})}_{\textit{Observation likelihood}} \tag{5.5}$$

The Markovian state transition prior can be used to model the time variations and the sequential dependence of most non-stationary processes. However, for many applications, such as speech recognition, the state observation likelihood has far more influence on the posterior probability than the state transition prior.

5.2.3 Parameters of a Hidden Markov Model

A hidden Markov model has the following parameters:

Number of states N. This is usually set to the total number of distinct, or elementary, stochastic events in a signal process. For example, in modelling a binary-state process such as impulsive noise, N is set to 2, and in isolated-word speech modelling N is set between 5 to 10.

State transition-probability matrix $\boldsymbol{A}=\{a_{ij},\ i,j=1,\ \dots\ N\}$. This provides a Markovian connection network between the states, and models the variations in the duration of the signals associated with each state. For a left–right HMM (see Figure 5.5), $a_{ij}=0$ for $i>j$, and hence the transition matrix $\boldsymbol{A}$ is upper-triangular.

State observation vectors $\{\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, \dots, \boldsymbol{\mu}_{iM}, i=1, \dots, N\}$. For each state a set of M prototype vectors model the centroids of the signal space associated with each state.

State observation vector probability model. This can be either a discrete model composed of the M prototype vectors and their associated probability mass function (pmf) $\boldsymbol{P}=\{P_{ij}(\cdot);\ i=1, \dots, N, j=1, \dots\ M\}$, or it may be a continuous (usually Gaussian) pdf model $\boldsymbol{F}=\{f_{ij}(\cdot);\ i=1, \dots, N, j=1, \dots, M\}$.

Initial state probability vector $\boldsymbol{\pi}=[\pi_1, \pi_2, \dots, \pi_N]$.

5.2.4 State Observation Models

Depending on whether a signal process is discrete-valued or continuous-valued, the state observation model for the process can be either a discrete-valued probability mass function (pmf), or a continuous-valued probability density function (pdf). The discrete models can also be used for the modelling of the space of a continuous-valued process quantised into a number of discrete points. First, consider a discrete state observation density model. Assume that associated with the i^{th} state of an HMM there are M discrete centroid vectors $[\boldsymbol{\mu}_{i1}, \dots, \boldsymbol{\mu}_{iM}]$ with a pmf $[P_{i1}, \dots, P_{iM}]$. These centroid vectors and their probabilities are normally obtained through clustering of a set of training signals associated with each state.

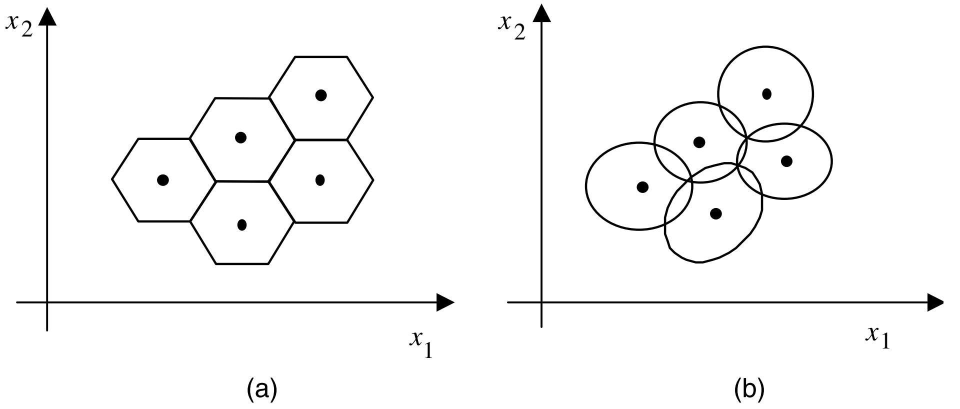


Figure 5.6 Modelling a random signal space using (a) a discrete-valued pmf and (b) a continuous-valued mixture Gaussian density.

For the modelling of a continuous-valued process, the signal space associated with each state is partitioned into a number of clusters as in Figure 5.6. If the signals within each cluster are modelled by a uniform distribution then each cluster is described by the centroid vector and the cluster probability, and the state observation model consists of M cluster centroids and the associated pmf $\{\boldsymbol{\mu}_{ik}, P_{ik}; i=1, ..., N, k=1, ..., M\}$. In effect, this results in a discrete state observation HMM for a continuous-valued process. Figure 5.6(a) shows a partitioning, and quantisation, of a signal space into a number of centroids.

Now if each cluster of the state observation space is modelled by a continuous pdf, such as a Gaussian pdf, then a continuous density HMM results. The most widely used state observation pdf for an HMM is the mixture Gaussian density defined as

$$f_{X|S}(\boldsymbol{x}|s=i)=\sum_{k=1}^{M} P_{ik}\,\mathcal{N}(\boldsymbol{x},\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik}) \tag{5.6}$$

where $\mathcal{N}(\boldsymbol{x},\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik})$ is a Gaussian density with mean vector $\boldsymbol{\mu}_{ik}$ and covariance matrix $\boldsymbol{\Sigma}_{ik}$, and P_{ik} is a mixture weighting factor for the k^{th} Gaussian pdf of the state i. Note that P_{ik} is the prior probability of the k^{th} mode of the mixture pdf for the state i. Figure 5.6(b) shows the space of a mixture Gaussian model of an observation signal space. A 5-mode mixture Gaussian pdf is shown in Figure 5.7.

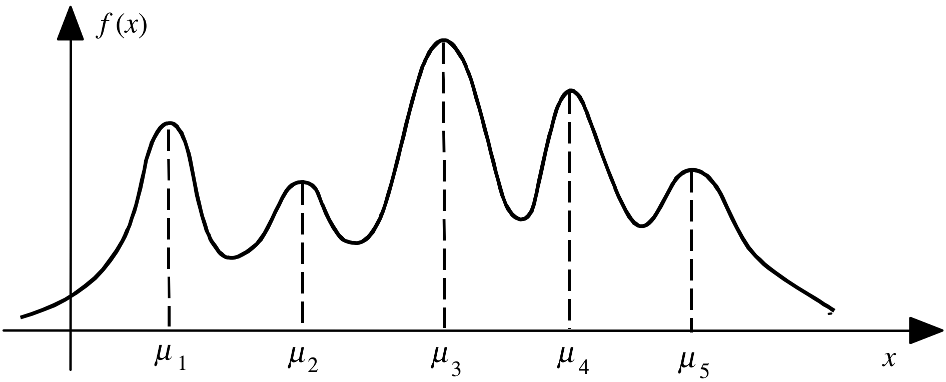


Figure 5.7 A mixture Gaussian probability density function.

5.2.5 State Transition Probabilities

The first-order Markovian property of an HMM entails that the transition probability to any state $s(t)$ at time t depends only on the state of the process at time t–1, $s(t–1)$, and is independent of the previous states of the HMM. This can be expressed as

$$\begin{aligned} &Prob(s(t)=j \mid s(t-1)=i, s(t-2)=k, \ldots, s(t-N)=l) \\ &=Prob(s(t)=j \mid s(t-1)=i)=a_{ij} \end{aligned} \tag{5.7}$$

where $s(t)$ denotes the state of HMM at time t. The transition probabilities provide a probabilistic mechanism for connecting the states of an HMM, and for modelling the variations in the duration of the signals associated with each state. The probability of occupancy of a state i for d consecutive time units, $P_i(d)$, can be expressed in terms of the state self-loop transition probabilities a_{ii} as

$$P_i(d)=a_{ii}^{d-1}(1-a_{ii}) \tag{5.8}$$

From Equation (5.8), using the geometric series conversion formula, the mean occupancy duration for each state of an HMM can be derived as

$$\text{Mean occupancy of state } i=\sum_{d=0}^{\infty} d P_i(d)=\frac{1}{1-a_{ii}} \tag{5.9}$$

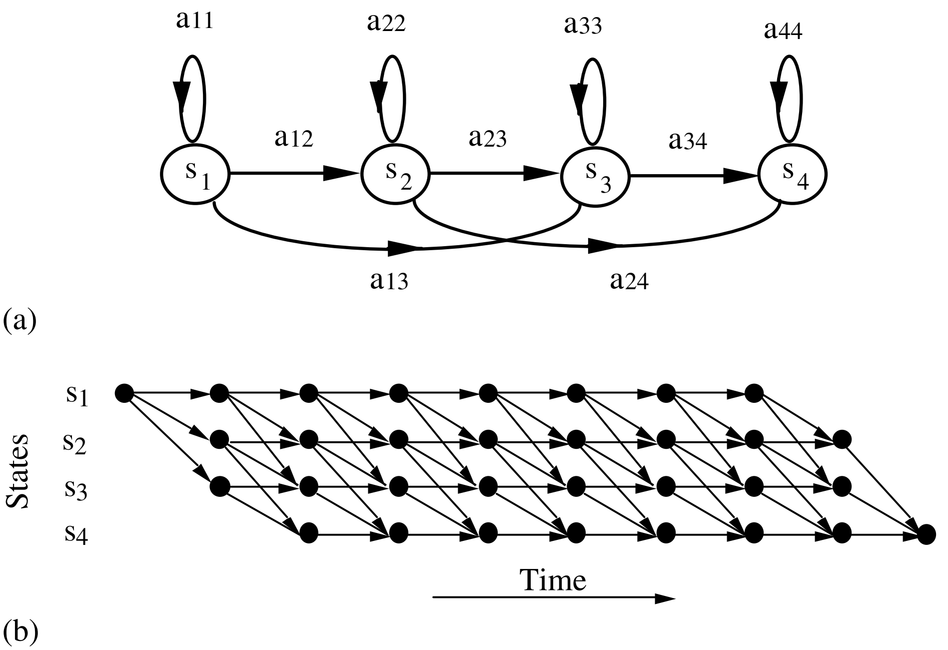


Figure 5.8 (a) A 4-state left–right HMM, and (b) its state–time trellis diagram.

5.2.6 State–Time Trellis Diagram

A state–time trellis diagram shows the HMM states together with all the different paths that can be taken through various states as time unfolds. Figure 5.8(a) and 5.8(b) illustrate a 4-state HMM and its state–time diagram. Since the number of states and the state parameters of an HMM are time-invariant, a state-time diagram is a repetitive and regular trellis structure. Note that in Figure 5.8 for a left–right HMM the state–time trellis has to diverge from the first state and converge into the last state. In general, there are many different state sequences that start from the initial state and end in the final state. Each state sequence has a prior probability that can be obtained by multiplication of the state transition probabilities of the sequence. For example, the probability of the state sequence $\boldsymbol{s} = [S_1, S_1, S_2, S_2, S_3, S_3, S_4]$ is $P(\boldsymbol{s}) = \pi_1 a_{11} a_{12} a_{22} a_{23} a_{33} a_{34}$. Since each state has a different set of prototype observation vectors, different state sequences model different observation sequences. In general an N-state HMM can reproduce N^T different realisations of the random process that it is trained to model.

5.3 Training Hidden Markov Models

The first step in training the parameters of an HMM is to collect a training database of a sufficiently large number of different examples of the random process to be modelled. Assume that the examples in a training database consist of L vector-valued sequences $[\boldsymbol{X}]=[\boldsymbol{X}_k;\ k=0, \ldots, L-1]$, with each sequence $\boldsymbol{X}_k=[\boldsymbol{x}(t);\ t=0, \ldots, T_k-1]$ having a variable number of T_k vectors. The objective is to train the parameters of an HMM to model the statistics of the signals in the training data set. In a probabilistic sense, the fitness of a model is measured by the posterior probability $P_{\mathcal{M}|X}(\mathcal{M}|\boldsymbol{X})$ of the model $\mathcal{M}$ given the training data $\boldsymbol{X}$. The training process aims to maximise the posterior probability of the model $\mathcal{M}$ and the training data $[\boldsymbol{X}]$, expressed using Bayes' rule as

$$P_{\mathcal{M}|X}(\mathcal{M}|\boldsymbol{X})=\frac{1}{f_X(\boldsymbol{X})} f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})P_{\mathcal{M}}(\mathcal{M}) \tag{5.10}$$

where the denominator $f_X(\boldsymbol{X})$ on the right-hand side of Equation (5.10) has only a normalising effect and $P_{\mathcal{M}}(\mathcal{M})$ is the prior probability of the model $\mathcal{M}$. For a given training data set $[\boldsymbol{X}]$ and a given model $\mathcal{M}$, maximising Equation (5.10) is equivalent to maximising the likelihood function $P_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$. The likelihood of an observation vector sequence $\boldsymbol{X}$ given a model $\mathcal{M}$ can be expressed as

$$f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})=\sum_{\boldsymbol{s}} f_{X|S,\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M})P_{s|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}) \tag{5.11}$$

where $f_{X|S,\mathcal{M}}(\boldsymbol{X}(t)|s(t),\mathcal{M})$, the pdf of the signal sequence $\boldsymbol{X}$ along the state sequence $\mathbf{s}=[s(0),s(1),\ldots,s(T-1)]$ of the model $\mathcal{M}$, is given by

$$f_{X|S,\mathcal{M}}(\boldsymbol{X}|\mathbf{s},\mathcal{M})= f_{X|S}(\boldsymbol{x}(0)|s(0))f_{X|S}(\boldsymbol{x}(1)|s(1))\cdots f_{X|S}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.12}$$

where $s(t)$, the state at time t, can be one of N states, and $f_{X|S}(\boldsymbol{X}(t)|s(t))$, a shorthand for $f_{X|S,\mathcal{M}}(\boldsymbol{X}(t)|s(t),\mathcal{M})$, is the pdf of $\boldsymbol{x}(t)$ given the state $s(t)$ of the model $\mathcal{M}$. The Markovian probability of the state sequence $\boldsymbol{s}$ is given by

$$P_{S|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})=\pi_{s(0)}a_{s(0)s(1)}a_{s(1)s(2)}\cdots a_{s(T-2)s(T-1)} \tag{5.13}$$

Substituting Equations (5.12) and (5.13) in Equation (5.11) yields

$$f_{X|\mathcal{M}}(X \mid \mathcal{M}) = \sum_{\mathbf{s}} f_{X|S,\mathcal{M}}(X \mid s, \mathcal{M}) P_{s|\mathcal{M}}(s \mid \mathcal{M})$$
$$= \sum_{\mathbf{s}} \pi_{s(0)} f_{X|S}(\boldsymbol{x}(0)|s(0))\, a_{s(0)s(1)} f_{X|S}(\boldsymbol{x}(1)|s(1)) \cdots a_{s(T-2)s(T-1)} f_{X|S}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.14}$$

where the summation is taken over all state sequences $\boldsymbol{s}$. In the training process, the transition probabilities and the parameters of the observation pdfs are estimated to maximise the model likelihood of Equation (5.14). Direct maximisation of Equation (5.14) with respect to the model parameters is a non-trivial task. Furthermore, for an observation sequence of length T vectors, the computational load of Equation (5.14) is $O(N^T)$. This is an impractically large load, even for such modest values as N=6 and T=30. However, the repetitive structure of the trellis state–time diagram of an HMM implies that there is a large amount of repeated computation in Equation (5.14) that can be avoided in an efficient implementation. In the next section we consider the forward-backward method of model likelihood calculation, and then proceed to describe an iterative maximum-likelihood model optimisation method.

5.3.1 Forward–Backward Probability Computation

An efficient recursive algorithm for the computation of the likelihood function $f_{X|\mathcal{M}}(X|\mathcal{M})$ is the forward–backward algorithm. The forward–backward computation method exploits the highly regular and repetitive structure of the state–time trellis diagram of Figure 5.8.

In this method, a forward probability variable $\alpha_t(i)$ is defined as the joint probability of the partial observation sequence $\boldsymbol{X}$=[$\boldsymbol{x}(0)$, $\boldsymbol{x}(1)$, ..., $\boldsymbol{x}(t)$] and the state i at time t, of the model $\mathcal{M}$:

$$\alpha_t(i) = f_{X,S|\mathcal{M}}(\boldsymbol{x}(0), \boldsymbol{x}(1), \ldots, \boldsymbol{x}(t), s(t) = i \,|\mathcal{M}) \tag{5.15}$$

The forward probability variable $\alpha_t(i)$ of Equation (5.15) can be expressed in a recursive form in terms of the forward probabilities at time t–1, $\alpha_{t-1}(i)$:

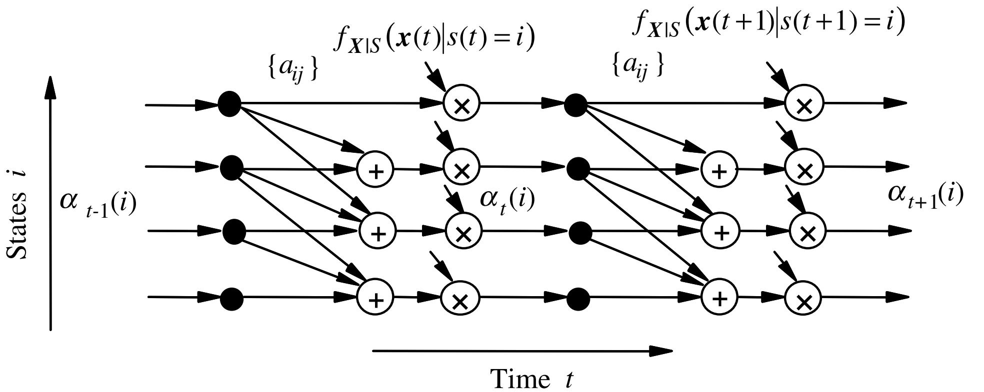


Figure 5.9 A network for computation of forward probabilities for a left-right HMM.

$$
\begin{aligned}
\alpha_t(i) &= f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\dots,\boldsymbol{x}(t),\, s(t)=i\big|\mathcal{M}\big)\\
&= \left(\sum_{j=1}^{N} f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\dots,\boldsymbol{x}(t-1),\, s(t-1)=j\big|\mathcal{M}\big)a_{ji}\right) f_{X|S,\mathcal{M}}\big(\boldsymbol{x}(t)\big|s(t)=i,\mathcal{M}\big)\\
&= \sum_{j=1}^{N}\big(\alpha_{t-1}(j)\,a_{ji}\big) f_{X|S,\mathcal{M}}\big(\boldsymbol{x}(t)\big|s(t)=i,\mathcal{M}\big)
\end{aligned}
\tag{5.16}
$$

Figure 5.9 illustrates, a network for computation of the forward probabilities for the 4-state left–right HMM of Figure 5.8. The likelihood of an observation sequence $\boldsymbol{X}=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{x}(T-1)]$ given a model $\mathcal{M}$ can be expressed in terms of the forward probabilities as

$$
\begin{aligned}
f_{X|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\dots,\boldsymbol{x}(T-1)\big|\mathcal{M}\big) &= \sum_{i=1}^{N} f_{X,S|\mathcal{M}}\big(\boldsymbol{x}(0),\boldsymbol{x}(1),\dots,\boldsymbol{x}(T-1),\, s(T-1)=i\big|\mathcal{M}\big)\\
&= \sum_{i=1}^{N}\alpha_{T-1}(i)
\end{aligned}
\tag{5.17}
$$

Similar to the definition of the forward probability concept, a backward probability is defined as the probability of the state i at time t followed by the partial observation sequence $[\boldsymbol{x}(t+1), \boldsymbol{x}(t+2), ..., \boldsymbol{x}(T-1)]$ as

$$\beta_t(i) = f_{X,S|\mathcal{M}}(s(t)=i, x(t+1), x(t+2), \ldots, x(T-1)|\mathcal{M})$$
$$= \sum_{j=1}^{N} a_{ij} f_{X,S|\mathcal{M}}(s(t+1)=j, x(t+2), x(t+3), \ldots, x(T-1))$$
$$\times f_{X|S}(x(t+1)|s(t+1)=j,|\mathcal{M})$$
$$= \sum_{j=1}^{N} a_{ij} \beta_{t+1}(j) f_{X|S,\mathcal{M}}(x(t+1)|s(t+1)=j,|\mathcal{M}) \tag{5.18}$$

In the next section, forward and backward probabilities are used to develop a method for the training of HMM parameters.

5.3.2 Baum–Welch Model Re-Estimation

The HMM training problem is the estimation of the model parameters $\mathcal{M}=(\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{F})$ for a given data set. These parameters are the initial state probabilities $\boldsymbol{\pi}$, the state transition probability matrix $\boldsymbol{A}$ and the continuous (or discrete) density state observation pdfs. The HMM parameters are estimated from a set of training examples $\{\boldsymbol{X}=[\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]\}$, with the objective of maximising $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$, the likelihood of the model and the training data. The Baum–Welch method of training HMMs is an iterative likelihood maximisation method based on the forward–backward probabilities defined in the preceding section. The Baum–Welch method is an instance of the EM algorithm described in Chapter 4. For an HMM $\mathcal{M}$, the posterior probability of a transition at time t from state i to state j of the model $\mathcal{M}$, given an observation sequence $\boldsymbol{X}$, can be expressed as

$$\gamma_t(i,j) = P_{S|X,\mathcal{M}}(s(t)=i, s(t+1)=j|\boldsymbol{X},\mathcal{M})$$
$$= \frac{f_{S,X|\mathcal{M}}(s(t)=i, s(t+1)=j, \boldsymbol{X}|\mathcal{M})}{f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})}$$
$$= \frac{\alpha_t(i) a_{ij} f_{X|S,\mathcal{M}}(\boldsymbol{x}(t+1)|s(t+1)=j,\mathcal{M})\beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_{T-1}(i)} \tag{5.19}$$

where $f_{S,X|\mathcal{M}}(s(t)=i, s(t+1)=j, \boldsymbol{X}|\mathcal{M})$ is the joint pdf of the states $s(t)$ and

$s(t+1)$ and the observation sequence $\boldsymbol{X}$, and $f_{X|S}(\boldsymbol{x}(t+1)|s(t+1)=i)$ is the state observation pdf for the state i. Note that for a discrete observation density HMM the state observation pdf in Equation (5.19) is replaced with the discrete state observation pmf $P_{X|S}(\boldsymbol{x}(t+1)|s(t+1)=i)$. The posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is

$$
\begin{aligned}
\gamma_t(i) &= P_{S|\boldsymbol{X},\mathcal{M}}\left(s(t)=i \middle| \boldsymbol{X},\mathcal{M}\right) \\
&= \frac{f_{S,\boldsymbol{X}|\mathcal{M}}\left(s(t)=i,\boldsymbol{X} \middle| \mathcal{M}\right)}{f_{\boldsymbol{X}|\mathcal{M}}\left(\boldsymbol{X} \middle| \mathcal{M}\right)} \\
&= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)}
\end{aligned}
\tag{5.20}
$$

Now the state transition probability a_{ij} can be interpreted as

$$
a_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \tag{5.21}
$$

From Equations (5.19)–(5.21), the state transition probability can be re-estimated as the ratio

$$
\bar{a}_{ij} = \frac{\sum_{t=0}^{T-2}\gamma_t(i,j)}{\sum_{t=0}^{T-2}\gamma_t(i)} \tag{5.22}
$$

Note that for an observation sequence $[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]$ of length T, the last transition occurs at time $T-2$ as indicated in the upper limits of the summations in Equation (5.22). The initial-state probabilities are estimated as

$$
\bar{\pi}_i = \gamma_0(i) \tag{5.23}
$$

5.3.3 Training HMMs with Discrete Density Observation Models

In a discrete density HMM, the observation signal space for each state is modelled by a set of discrete symbols or vectors. Assume that a set of M vectors $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, ..., \boldsymbol{\mu}_{iM}]$ model the space of the signal associated with the i^{th} state. These vectors may be obtained from a clustering process as the centroids of the clusters of the training signals associated with each state. The objective in training discrete density HMMs is to compute the state transition probabilities and the state observation probabilities. The forward–backward equations for discrete density HMMs are the same as those for continuous density HMMs, derived in the previous sections, with the difference that the probability density functions such as $f_{X|S}(\boldsymbol{x}(t)|s(t)=i)$ are substituted with probability mass functions $P_{X|S}(\boldsymbol{x}(t)|s(t)=i)$ defined as

$$P_{X|S}(\boldsymbol{x}(t)|s(t)=i) = P_{X|S}(Q[\boldsymbol{x}(t)]|s(t)=i) \tag{5.24}$$

where the function $Q[\boldsymbol{x}(t)]$ quantises the observation vector $\boldsymbol{x}(t)$ to the nearest discrete vector in the set $[\boldsymbol{\mu}_{i1}, \boldsymbol{\mu}_{i2}, ..., \boldsymbol{\mu}_{iM}]$. For discrete density HMMs, the probability of a state vector $\boldsymbol{\mu}_{ik}$ can be defined as the ratio of the number of occurrences of $\boldsymbol{\mu}_{ik}$ (or vectors quantised to $\boldsymbol{\mu}_{ik}$) in the state i, divided by the total number of occurrences of all other vectors in the state i:

$$\bar{P}_{ik}(\boldsymbol{\mu}_{ik}) = \frac{\text{expected number of times in state } i \text{ and observing } \boldsymbol{\mu}_{ik}}{\text{expected number of times in state } i} = \frac{\sum\limits_{t\in \boldsymbol{x}(t)\rightarrow\boldsymbol{\mu}_{ik}}^{T-1} \gamma_t(i)}{\sum\limits_{t=0}^{T-1} \gamma_t(i)} \tag{5.25}$$

In Equation (5.25) the summation in the numerator is taken over those time instants t where the k^{th} symbol $\boldsymbol{\mu}_{ik}$ is observed in the state i.

For statistically reliable results, an HMM must be trained on a large data set $\boldsymbol{X}$ consisting of a sufficient number of independent realisations of the process to be modelled. Assume that the training data set consists of L realisations $\boldsymbol{X}=[\boldsymbol{X}(0), \boldsymbol{X}(1), ..., \boldsymbol{X}(L-1)]$, where $\boldsymbol{X}(k)=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{x}(T_k-1)]$. The re-estimation formula can be averaged over the entire data set as

$$\hat{\pi}_i = \frac{1}{L}\sum_{l=0}^{L-1}\gamma_0^l(i) \tag{5.26}$$

$$\hat{a}_{ij} = \frac{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-2}\gamma_t^l(i,j)}{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-2}\gamma_t^l(i)} \tag{5.27}$$

and

$$\hat{P}_i(\boldsymbol{\mu}_{ik}) = \frac{\sum_{l=0}^{L-1}\sum_{t\in \boldsymbol{x}(t)\rightarrow\boldsymbol{\mu}_{ik}}^{T_l-1}\gamma_t^l(i)}{\sum_{l=0}^{L-1}\sum_{t=0}^{T_l-1}\gamma_t^l(i)} \tag{5.28}$$

The parameter estimates of Equations (5.26)–(5.28) can be used in further iterations of the estimation process until the model converges.

5.3.4 HMMs with Continuous Density Observation Models

In continuous density HMMs, continuous probability density functions (pdfs) are used to model the space of the observation signals associated with each state. Baum et al. generalised the parameter re-estimation method to HMMs with concave continuous pdfs such a Gaussian pdf. A continuous P-variate Gaussian pdf for the state i of an HMM can be defined as

$$f_{X|S}\left(\boldsymbol{x}(t)\middle|s(t)=i\right) = \frac{1}{(2\pi)^{P/2}\left|\boldsymbol{\Sigma}_i\right|^{1/2}}\exp\left\{[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}[\boldsymbol{x}(t)-\boldsymbol{\mu}_i]\right\} \tag{5.29}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and the covariance matrix associated with the state i. The re-estimation formula for the mean vector of the state Gaussian pdf can be derived as

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)\boldsymbol{x}(t)}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.30}$$

Similarly, the covariance matrix is estimated as

$$\bar{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=0}^{T-1} \gamma_t(i)(\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_i)(\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_i)^{\mathrm{T}}}{\sum_{t=0}^{T-1} \gamma_t(i)} \tag{5.31}$$

The proof that the Baum–Welch re-estimation algorithm leads to maximisation of the likelihood function $f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M})$ can be found in Baum.

5.3.5 HMMs with Mixture Gaussian pdfs

The modelling of the space of a signal process with a mixture of Gaussian pdfs is considered in Section 4.5. In HMMs with mixture Gaussian pdf state models, the signal space associated with the i^{th} state is modelled with a mixtures of M Gaussian densities as

$$f_{\boldsymbol{X}|S}(\boldsymbol{x}(t)|s(t)=i) = \sum_{k=1}^{M} P_{ik}\, \mathcal{N}(\boldsymbol{x}(t), \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \tag{5.32}$$

where P_{ik} is the prior probability of the k^{th} component of the mixture. The posterior probability of state i at time t and state j at time t+1 of the model $\mathcal{M}$, given an observation sequence $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(T-1)]$, can be expressed as

$$\begin{aligned} \gamma_t(i,j) &= P_{\boldsymbol{S}|\boldsymbol{X},\mathcal{M}}(s(t)=i, s(t+1)=j \mid \boldsymbol{X}, \mathcal{M}) \\ &= \frac{\alpha_t(i)a_{ij}\left[\sum_{k=1}^{M} P_{jk}\,\mathcal{N}(\boldsymbol{x}(t+1), \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk})\right]\beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_{T-1}(i)} \end{aligned} \tag{5.33}$$

and the posterior probability of state i at time t given the model $\mathcal{M}$ and the observation $\boldsymbol{X}$ is given by

$$\gamma_t(i) = P_{S|\boldsymbol{X},\mathcal{M}}(s(t)=i|\boldsymbol{X},\mathcal{M}) \\ = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)} \tag{5.34}$$

Now we define the joint posterior probability of the state i and the k^{th} Gaussian mixture component pdf model of the state i at time t as

$$\zeta_t(i,k) = P_{S,K|\boldsymbol{X},\mathcal{M}}\left(s(t)=i, m(t)=k \middle| \boldsymbol{X},\mathcal{M}\right) \\ = \frac{\sum_{j=1}^{N}\alpha_{t-1}(j)a_{ji}P_{ik}\mathcal{N}\left(\boldsymbol{x}(t),\boldsymbol{\mu}_{ik},\boldsymbol{\Sigma}_{ik}\right)\beta_t(i)}{\sum_{j=1}^{N}\alpha_{T-1}(j)} \tag{5.35}$$

where $m(t)$ is the Gaussian mixture component at time t. Equations (5.33) to (5.35) are used to derive the re-estimation formula for the mixture coefficients, the mean vectors and the covariance matrices of the state mixture Gaussian pdfs as

$$\overline{P}_{ik} = \frac{\text{expected number of times in state } i \text{ and observing mixture } k}{\text{expected number of times in state } i} \\ = \frac{\sum_{t=0}^{T-1}\xi_t(i,k)}{\sum_{t=0}^{T-1}\gamma_t(i)} \tag{5.36}$$

and

$$\overline{\boldsymbol{\mu}}_{ik} = \frac{\sum_{t=0}^{T-1}\xi_t(i,k)\boldsymbol{x}(t)}{\sum_{t=0}^{T-1}\xi_t(i,k)} \tag{5.37}$$

Similarly the covariance matrix is estimated as

$$\bar{\Sigma}_{ik} = \frac{\sum_{t=0}^{T-1} \xi_t(i,k)[\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_{ik}][\boldsymbol{x}(t) - \bar{\boldsymbol{\mu}}_{ik}]^{\mathrm{T}}}{\sum_{t=0}^{T-1} \xi_t(i,k)} \tag{5.38}$$

5.4 Decoding of Signals Using Hidden Markov Models

Hidden Markov models are used in applications such as speech recognition, image recognition and signal restoration, and for the decoding of the underlying states of a signal. For example, in speech recognition, HMMs are trained to model the statistical variations of the acoustic realisations of the words in a vocabulary of say size *V* words. In the word recognition phase, an utterance is classified and labelled with the most likely of the *V*+1 candidate HMMs (including an HMM for silence) as illustrated in Figure 5.10. In Chapter 12 on the modelling and detection of impulsive noise, a binary–state HMM is used to model the impulsive noise process.

Consider the decoding of an unlabelled sequence of *T* signal vectors $\boldsymbol{X}=[\boldsymbol{x}(0), \boldsymbol{x}(1), ..., \boldsymbol{X}(T-1)]$ given a set of *V* candidate HMMs $[\mathcal{M}_1, ..., \mathcal{M}_V]$. The probability score for the observation vector sequence $\boldsymbol{X}$ and the model $\mathcal{M}_k$ can be calculated as the likelihood:

$$f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_k) = \sum_{\mathbf{s}} \pi_{s(0)} f_{X|S}(\boldsymbol{x}(0)|s(0)) a_{s(0)s(1)} f_{X|S}(\boldsymbol{x}(1)|s(1)) \cdots a_{s(T-2)s(T-1)} f_{X|S}(\boldsymbol{x}(T-1)|s(T-1)) \tag{5.39}$$

where the likelihood of the observation sequence $\boldsymbol{X}$ is summed over all possible state sequences of the model $\mathcal{M}$. Equation (5.39) can be efficiently calculated using the forward–backward method described in Section 5.3.1. The observation sequence $\boldsymbol{X}$ is labelled with the HMM that scores the highest likelihood as

$$Label(\boldsymbol{X}) = \arg\max_k \left(f_{X|\mathcal{M}}(\boldsymbol{X}|\mathcal{M}_k) \right), \qquad k=1, ..., V+1 \tag{5.40}$$

In decoding applications often the likelihood of an observation sequence $\boldsymbol{X}$ and a model $\mathcal{M}_k$ is obtained along the *single* most likely state sequence of

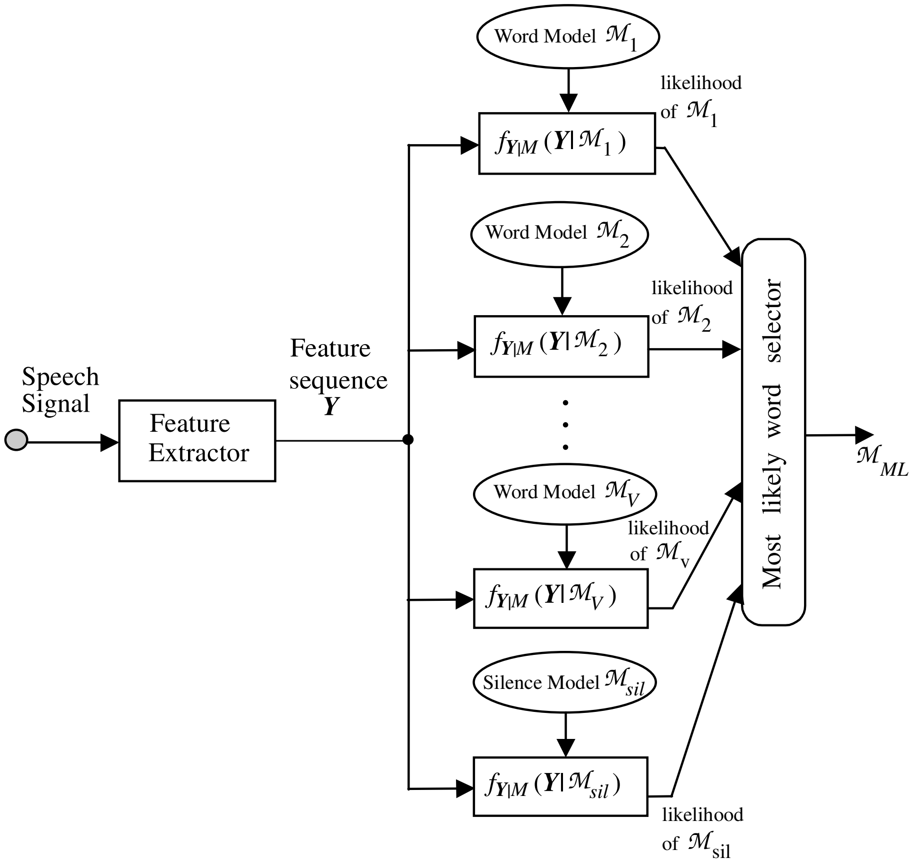


Figure 5.10 Illustration of the use of HMMs in speech recognition.

model $\mathcal{M}_k$, instead of being summed over all sequences, so Equation (5.40) becomes

$$Label(\boldsymbol{X}) = \arg\max_k \left[\max_{\boldsymbol{s}} f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(\boldsymbol{X},\boldsymbol{s}|\mathcal{M}_k) \right] \tag{5.41}$$

In Section 5.5, on the use of HMMs for noise reduction, the most likely state sequence is used to obtain the maximum-likelihood estimate of the underlying statistics of the signal process.

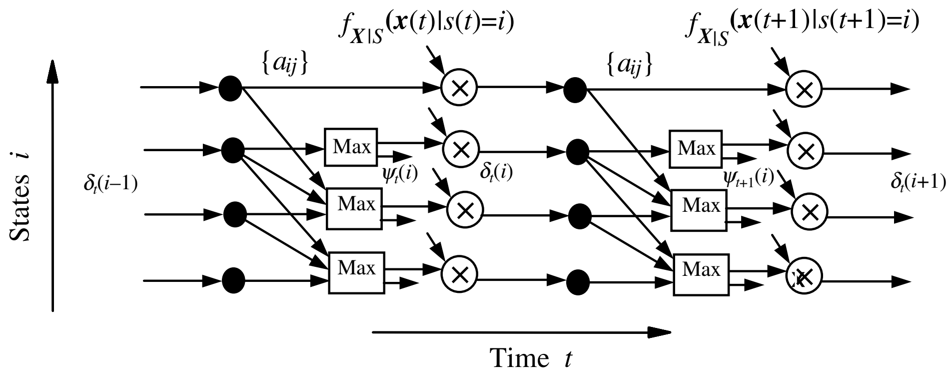


Figure 5.11 A network illustration of the Viterbi algorithm.

5.4.1 Viterbi Decoding Algorithm

In this section, we consider the decoding of a signal to obtain the maximum a posterior (MAP) estimate of the underlying state sequence. The MAP state sequence $\boldsymbol{s}^{MAP}$ of a model $\mathcal{M}$ given an observation signal sequence $\boldsymbol{X}$=[$\boldsymbol{x}$(0), ..., $\boldsymbol{x}$(T–1)] is obtained as

$$\begin{aligned} \boldsymbol{s}^{MAP} &= \arg\max_{\boldsymbol{s}} f_{\boldsymbol{X},\boldsymbol{S}|\mathcal{M}}(\boldsymbol{X},\boldsymbol{s}|\mathcal{M}) \\ &= \arg\max_{\boldsymbol{s}} \left(f_{\boldsymbol{X}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{X}|\boldsymbol{s},\mathcal{M}) P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}) \right) \end{aligned} \tag{5.42}$$

The MAP state sequence estimate is used in such applications as the calculation of a similarity score between a signal sequence $\boldsymbol{X}$ and an HMM $\mathcal{M}$, segmentation of a non-stationary signal into a number of distinct quasi-stationary segments, and implementation of state-based Wiener filters for restoration of noisy signals as described in the next section.

For an N-state HMM and an observation sequence of length T, there are altogether N^T state sequences. Even for moderate values of N and T say (N=6 and T=30), an exhaustive search of the state–time trellis for the best state sequence is a computationally prohibitive exercise. The Viterbi algorithm is an efficient method for the estimation of the most likely state sequence of an HMM. In a state–time trellis diagram, such as Figure 5.8, the number of paths diverging from each state of a trellis can grow exponentially by a factor of N at successive time instants. The Viterbi method prunes the trellis by selecting the most likely path to each state. At each time instant *t*, for each state *i*, the algorithm selects the most probable path to state *i* and prunes out the less likely branches. This procedure ensures that at any time instant, only a single path *survives* into each state of the trellis.

For each time instant *t* and for each state *i*, the algorithm keeps a record of the state *j* from which the maximum-likelihood path branched into *i*, and also records the cumulative probability of the most likely path into state *i* at time *t*. The Viterbi algorithm is given on the next page, and Figure 5.11 gives a network illustration of the algorithm.

Viterbi Algorithm

$\delta_t(i)$ records the cumulative probability of the best path to state *i* at time *t*.

$\psi_t(i)$ records the best state sequence to state *i* at time *t*.

Step 1: *Initialisation,* at time *t*=0, for states *i*=1, …, *N*

$$\delta_0(i)=\pi_i f_i(\boldsymbol{x}(0))$$

$$\psi_0(i)=0$$

Step 2: *Recursive calculation* of the ML state sequences and their probabilities

For time *t* =1, …, *T*–1

For states *i* = 1, …, *N*

$$\delta_t(i)=\max_j[\delta_{t-1}(j)a_{ji}]f_i(\boldsymbol{x}(t))$$

$$\psi_t(i)=\arg\max_j[\delta_{t-1}(j)a_{ji}]$$

Step 3: *Termination,* retrieve the most likely final state

$$s^{MAP}(T-1)=\arg\max_i[\delta_{T-1}(i)]$$

$$Prob_{\max}=\max_i[\delta_{T-1}(i)]$$

Step 4: *Backtracking* through the most likely state sequence:

For *t* = *T*–2, …, 0

$$s^{MAP}(t)=\psi_{t+1}\left[s^{MAP}(t+1)\right].$$

The backtracking routine retrieves the most likely state sequence of the model $\mathcal{M}$. Note that the variable $Prob_{\max}$, which is the probability of the observation sequence $\boldsymbol{X}=[\boldsymbol{x}(0), \ldots, \boldsymbol{x}(T-1)]$ and the most likely state sequence of the model $\mathcal{M}$, can be used as the probability score for the model $\mathcal{M}$ and the observation $\boldsymbol{X}$. For example, in speech recognition, for each candidate word model the probability of the observation and the most likely state sequence is calculated, and then the observation is labelled with the word that achieves the highest probability score.

5.5 HMM-Based Estimation of Signals in Noise

In this section, and the following two sections, we consider the use of HMMs for estimation of a signal $\boldsymbol{x}(t)$ observed in an additive noise $\boldsymbol{n}(t)$, and modelled as

$$\boldsymbol{y}(t)=\boldsymbol{x}(t)+\boldsymbol{n}(t) \tag{5.43}$$

From Bayes' rule, the posterior pdf of the signal $\boldsymbol{x}(t)$ given the noisy observation $\boldsymbol{y}(t)$ is defined as

$$\begin{aligned} f_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}(t)|\boldsymbol{y}(t)) &= \frac{f_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}(t)|\boldsymbol{x}(t))f_{\boldsymbol{X}}(\boldsymbol{x}(t))}{f_{Y}(\boldsymbol{y}(t))} \\ &= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y}(t))} f_{\boldsymbol{N}}(\boldsymbol{y}(t)-\boldsymbol{x}(t))f_{\boldsymbol{X}}(\boldsymbol{x}(t)) \end{aligned} \tag{5.44}$$

For a given observation, $f_{\boldsymbol{Y}}(\boldsymbol{y}(t))$ is a constant, and the maximum a posteriori (MAP) estimate is obtained as

$$\hat{\boldsymbol{x}}^{MAP}(t)=\arg\max_{\boldsymbol{x}(t)} f_{\boldsymbol{N}}(\boldsymbol{y}(t)-\boldsymbol{x}(t))f_{\boldsymbol{X}}(\boldsymbol{x}(t)) \tag{5.45}$$

The computation of the posterior pdf, Equation (5.44), or the MAP estimate Equation (5.45), requires the pdf models of the signal and the noise processes. Stationary, continuous-valued, processes are often modelled by a Gaussian or a mixture Gaussian pdf that is equivalent to a single-state HMM. For a non-stationary process an N-state HMM can model the time-varying pdf of the process as a Markovian chain of N stationary Gaussian subprocesses. Now assume that we have an N_s-state HMM $\mathcal{M}$ for the signal, and another N_n-state HMM η for the noise. For signal estimation, we need estimates of the underlying state sequences of the signal and the noise processes. For an observation sequence of length T, there are N_s^T possible signal state sequences and N_n^T possible noise state sequences that could have generated the noisy signal. Since it is assumed that the signal and noise are uncorrelated, each signal state may be observed in any noisy state; therefore the number of noisy signal states is on the order of $N_s^T \times N_n^T$.

Given an observation sequence $\boldsymbol{Y}=[\boldsymbol{y}(0), \boldsymbol{y}(1), \ldots, \boldsymbol{y}(T-1)]$, the most probable state sequences of the signal and the noise HMMs maybe expressed as

$$\boldsymbol{s}_{\text{signal}}^{MAP} = \arg\max_{\boldsymbol{s}_{\text{signal}}} \left(\max_{\boldsymbol{s}_{\text{noise}}} f_{\boldsymbol{Y}}\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} \middle| \mathcal{M}, \eta\right) \right) \tag{5.46}$$

and

$$\boldsymbol{s}_{\text{noise}}^{MAP} = \arg\max_{\boldsymbol{s}_{\text{noise}}} \left(\max_{\boldsymbol{s}_{\text{signal}}} f_{\boldsymbol{Y}}\left(\boldsymbol{Y}, \boldsymbol{s}_{\text{signal}}, \boldsymbol{s}_{\text{noise}} \middle| \mathcal{M}, \eta\right) \right) \tag{5.47}$$

Given the state sequence estimates for the signal and the noise models, the MAP estimation Equation (5.45) becomes

$$\hat{\boldsymbol{x}}^{MAP}(t) = \arg\max_{\boldsymbol{x}} \left(f_{N|\boldsymbol{S},\eta}\left(\boldsymbol{y}(t) - \boldsymbol{x}(t) \middle| \boldsymbol{s}_{\text{noise}}^{MAP}, \eta\right) f_{X|\boldsymbol{S},\mathcal{M}}\left(\boldsymbol{x}(t) \middle| \boldsymbol{s}_{\text{signal}}^{MAP}, \mathcal{M}\right) \right) \tag{5.48}$$

Implementation of Equations (5.46)–(5.48) is computationally prohibitive. In Sections 5.6 and 5.7, we consider some practical methods for the estimation of signal in noise.

Example Assume a signal, modelled by a binary-state HMM, is observed in an additive stationary Gaussian noise. Let the noisy observation be modelled as

$$\boldsymbol{y}(t) = \bar{s}(t)\boldsymbol{x}_0(t) + s(t)\boldsymbol{x}_1(t) + \boldsymbol{n}(t) \tag{5.49}$$

where $s(t)$ is a hidden binary-state process such that: $s(t) = 0$ indicates that the signal is from the state S_0 with a Gaussian pdf of $\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{\mu}_{\boldsymbol{x}_0},\boldsymbol{\Sigma}_{\boldsymbol{x}_0\boldsymbol{x}_0})$, and $s(t)=1$ indicates that the signal is from the state S_1 with a Gaussian pdf of $\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{\mu}_{\boldsymbol{x}_1},\boldsymbol{\Sigma}_{\boldsymbol{x}_1\boldsymbol{x}_1})$. Assume that a stationary Gaussian process $\mathcal{N}(\boldsymbol{n}(t),\boldsymbol{\mu}_n,\boldsymbol{\Sigma}_{nn})$, equivalent to a single-state HMM, can model the noise. Using the Viterbi algorithm the maximum a posteriori (MAP) state sequence of the signal model can be estimated as

$$\boldsymbol{s}_{\text{signal}}^{MAP}=\arg\max_{\boldsymbol{s}}\left[f_{\boldsymbol{Y}|\boldsymbol{S},\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{s},\mathcal{M})P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M})\right] \tag{5.50}$$

For a Gaussian-distributed signal and additive Gaussian noise, the observation pdf of the noisy signal is also Gaussian. Hence, the state observation pdfs of the signal model can be modified to account for the additive noise as

$$f_{\boldsymbol{Y}|s_0}(\boldsymbol{y}(t)|s_0)=\mathcal{N}\left(\boldsymbol{y}(t),(\boldsymbol{\mu}_{\boldsymbol{x}_0}+\boldsymbol{\mu}_{\boldsymbol{n}}),(\boldsymbol{\Sigma}_{\boldsymbol{x}_0\boldsymbol{x}_0}+\boldsymbol{\Sigma}_{\boldsymbol{nn}})\right) \tag{5.51}$$

and

$$f_{\boldsymbol{Y}|s_1}(\boldsymbol{y}(t)|s_1)=\mathcal{N}\left(\boldsymbol{y}(t),(\boldsymbol{\mu}_{\boldsymbol{x}_1}+\boldsymbol{\mu}_{\boldsymbol{n}}),(\boldsymbol{\Sigma}_{\boldsymbol{x}_1\boldsymbol{x}_1}+\boldsymbol{\Sigma}_{\boldsymbol{nn}})\right) \tag{5.52}$$

where $\mathcal{N}(\boldsymbol{y}(t),\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes a Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The MAP signal estimate, given a state sequence estimate $\boldsymbol{s}^{MAP}$, is obtained from

$$\hat{\boldsymbol{x}}^{MAP}(t)=\arg\max_{\boldsymbol{x}}\left[f_{\boldsymbol{X}|S,\mathcal{M}}\left(\boldsymbol{x}(t)\middle|\boldsymbol{s}^{MAP},\mathcal{M}\right)f_{\boldsymbol{N}}(\boldsymbol{y}(t)-\boldsymbol{x}(t))\right] \tag{5.53}$$

Substitution of the Gaussian pdf of the signal from the most likely state sequence, and the pdf of noise, in Equation (5.53) results in the following MAP estimate:

$$\hat{\boldsymbol{x}}^{MAP}(t)=\left(\boldsymbol{\Sigma}_{xx,s(t)}+\boldsymbol{\Sigma}_{nn}\right)^{-1}\boldsymbol{\Sigma}_{xx,s(t)}(\boldsymbol{y}(t)-\boldsymbol{\mu}_n)+\left(\boldsymbol{\Sigma}_{xx,s(t)}+\boldsymbol{\Sigma}_{nn}\right)^{-1}\boldsymbol{\Sigma}_{nn}\,\boldsymbol{\mu}_{x,s(t)} \tag{5.54}$$

where $\boldsymbol{\mu}_{x,s(t)}$ and $\boldsymbol{\Sigma}_{xx,s(t)}$ are the mean vector and covariance matrix of the signal $\boldsymbol{x}(t)$ obtained from the most likely state sequence $[s(t)]$.

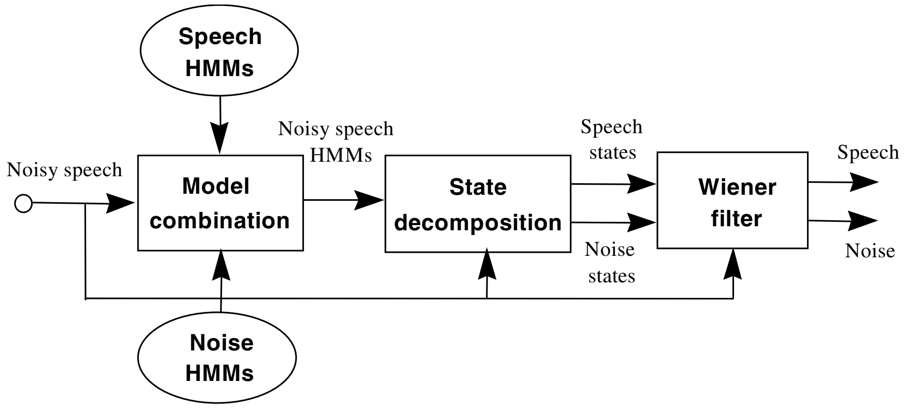


Figure 5.12 Outline configuration of HMM-based noisy speech recognition and enhancement.

5.6 Signal and Noise Model Combination and Decomposition

For Bayesian estimation of a signal observed in additive noise, we need to have an estimate of the underlying statistical state sequences of the signal and the noise processes. Figure 5.12 illustrates the outline of an HMM-based noisy speech recognition and enhancement system. The system performs the following functions:

(1) combination of the speech and noise HMMs to form the noisy speech HMMs;
(2) estimation of the best combined noisy speech model given the current noisy speech input;
(3) state decomposition, i.e. the separation of speech and noise states given noisy speech states;
(4) state-based Wiener filtering using the estimates of speech and noise states.

5.6.1 Hidden Markov Model Combination

The performance of HMMs trained on clean signals deteriorates rapidly in the presence of noise, since noise causes a mismatch between the clean HMMs and the noisy signals. The noise-induced mismatch can be reduced: either by filtering the noise from the signal (for example using the Wiener filtering and the spectral subtraction methods described in Chapters 6 and 11) or by combining the noise and the signal models to model the noisy signal. The model combination method was developed by Gales and Young. In this method HMMs of speech are combined with an HMM of noise to form HMMs of noisy speech signals. In the power-spectral domain, the mean vector and the covariance matrix of the noisy speech can be approximated by adding the mean vectors and the covariance matrices of speech and noise models:

$$\boldsymbol{\mu}_y = \boldsymbol{\mu}_x + g\boldsymbol{\mu}_n \tag{5.55}$$

$$\boldsymbol{\Sigma}_{yy} = \boldsymbol{\Sigma}_{xx} + g^2\boldsymbol{\Sigma}_{nn} \tag{5.56}$$

Model combination also requires an estimate of the current signal-to-noise ratio for calculation of the scaling factor g in Equations (5.55) and (5.56). In cases such as speech recognition, where the models are trained on cepstral features, the model parameters are first transformed from cepstral features into power spectral features before using the additive linear combination Equations (5.55) and (5.56). Figure 5.13 illustrates the combination of a 4-state left–right HMM of a speech signal with a 2-state ergodic HMM of noise. Assuming that speech and noise are independent processes, each speech state must be combined with every possible noise state to give the noisy speech model. It is assumed that the noise process only affects the mean vectors and the covariance matrices of the speech model; hence the transition probabilities of the speech model are not modified.

5.6.2 Decomposition of State Sequences of Signal and Noise

The HMM-based state decomposition problem can be stated as follows: given a noisy signal and the HMMs of the signal and the noise processes, estimate the underlying states of the signal and the noise.

HMM state decomposition can be obtained using the following method:

(a) Given the noisy signal and a set of combined signal and noise models, estimate the maximum-likelihood (ML) combined noisy HMM for the noisy signal.
(b) Obtain the ML state sequence of from the ML combined model.
(c) Extract the signal and noise states from the ML state sequence of the ML combined noisy signal model.

The ML state sequences provide the probability density functions for the signal and noise processes. The ML estimates of the speech and noise pdfs

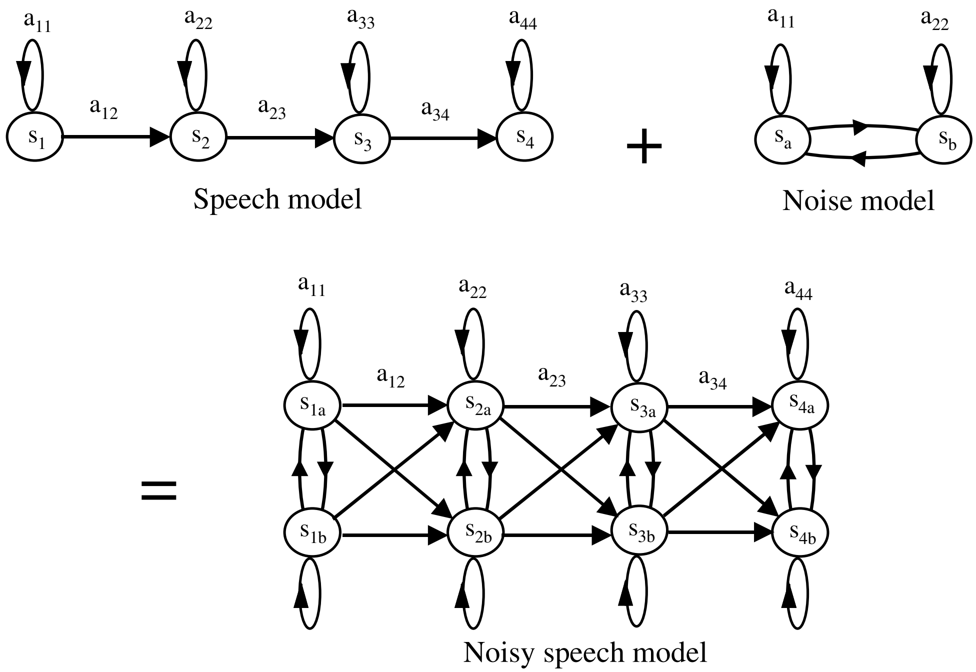


Figure 5.13 Outline configuration of HMM-based noisy speech recognition and enhancement. S_{ij} is a combination of the state *i* of speech with the state *j* of noise.

may then be used in Equation (5.45) to obtain a MAP estimate of the speech signal. Alternatively the mean spectral vectors of the speech and noise from the ML state sequences can be used to program a state-dependent Wiener filter as described in the next section.

5.7 HMM-Based Wiener Filters

The least mean square error Wiener filter is derived in Chapter 6. For a stationary signal $x(m)$, observed in an additive noise $n(m)$, the Wiener filter equations in the time and the frequency domains are derived as :

$$\boldsymbol{w} = (\boldsymbol{R}_{xx} + \boldsymbol{R}_{nn})^{-1}\boldsymbol{r}_{xx} \tag{5.55}$$

and

$$W(f) = \frac{P_{XX}(f)}{P_{XX}(f) + P_{NN}(f)} \tag{5.56}$$

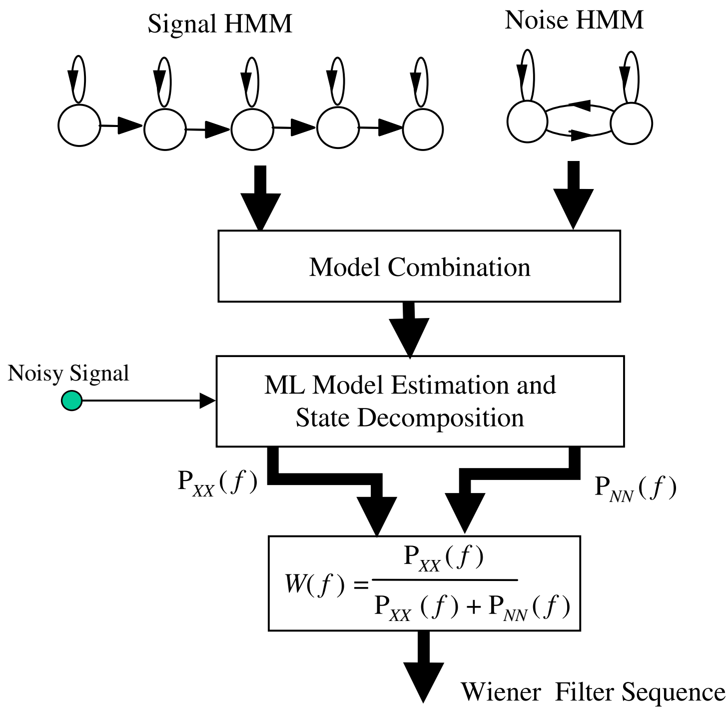


Figure 5.14 Illustrations of HMMs with state-dependent Wiener filters.

where $\boldsymbol{R_{xx}}$, $\boldsymbol{r_{xx}}$ and $P_{XX}(f)$ denote the autocorrelation matrix, the autocorrelation vector and the power-spectral functions respectively. The implementation of the Wiener filter, Equation (5.56), requires the signal and the noise power spectra. The power-spectral variables may be obtained from the ML states of the HMMs trained to model the power spectra of the signal and the noise. Figure 5.14 illustrates an implementation of HMM-based state-dependent Wiener filters. To implement the state-dependent Wiener filter, we need an estimate of the state sequences for the signal and the noise. In practice, for signals such as speech there are a number of HMMs; one HMM per word, phoneme, or any other elementary unit of the signal. In such cases it is necessary to classify the signal, so that the state-based Wiener filters are derived from the most likely HMM. Furthermore the noise process can also be modelled by an HMM. Assuming that there are V HMMs $\{\mathcal{M}_1, ..., \mathcal{M}_V\}$ for the signal process, and one HMM for the noise, the state-based Wiener filter can be implemented as follows:

Step 1: Combine the signal and noise models to form the noisy signal models.
Step 2: Given the noisy signal, and the set of combined noisy signal models, obtain the ML combined noisy signal model.
Step 3: From the ML combined model, obtain the ML state sequence of speech and noise.
Step 4: Use the ML estimate of the power spectra of the signal and the noise to program the Wiener filter Equation (5.56).
Step 5: Use the state-dependent Wiener filters to filter the signal.

5.7.1 Modelling Noise Characteristics

The implicit assumption in using an HMM for noise is that noise statistics can be modelled by a Markovian chain of N different stationary processes. A stationary noise process can be modelled by a single-state HMM. For a non-stationary noise, a multi-state HMM can model the time variations of the noise process with a finite number of quasi-stationary states. In general, the number of states required to accurately model the noise depends on the non-stationary character of the noise.

An example of a non-stationary noise process is the impulsive noise of Figure 5.15. Figure 5.16 shows a two-state HMM of the impulsive noise sequence where the state S_0 models the "off" periods between the impulses and the state S_1 models an impulse. In cases where each impulse has a well-defined temporal structure, it may be beneficial to use a multistate HMM to model the pulse itself. HMMs are used in Chapter 12 for modelling impulsive noise, and in Chapter 15 for channel equalisation.

5.8 Summary

HMMs provide a powerful method for the modelling of non-stationary processes such as speech, noise and time-varying channels. An HMM is a Bayesian finite-state process, with a Markovian state prior, and a state likelihood function that can be either a discrete density model or a continuous Gaussian pdf model. The Markovian prior models the time evolution of a non-stationary process with a chain of stationary sub-processes. The state observation likelihood models the space of the process within each state of the HMM.

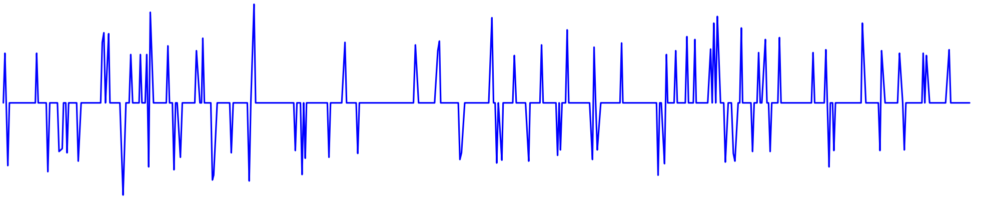


Figure 5.15 Impulsive noise.

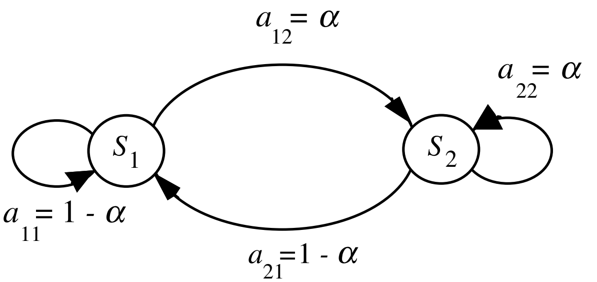


Figure 5.16 A binary-state model of an impulsive noise process.

In Section 5.3, we studied the Baum–Welch method for the training of the parameters of an HMM to model a given data set, and derived the forward–backward method for efficient calculation of the likelihood of an HMM given an observation signal. In Section 5.4, we considered the use of HMMs in signal classification and in the decoding of the underlying state sequence of a signal. The Viterbi algorithm is a computationally efficient method for estimation of the most likely sequence of an HMM. Given an unlabelled observation signal, the decoding of the underlying state sequence and the labelling of the observation with one of number of candidate HMMs are accomplished using the Viterbi method. In Section 5.5, we considered the use of HMMs for MAP estimation of a signal observed in noise, and considered the use of HMMs in implementation of state-based Wiener filter sequence.

Bibliography

Bahl L.R., Brown P.F., de Souza P.V. and Mercer R.L. (1986) Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition. IEEE Proc. Acoustics, Speech and Signal

Processing, ICASSP-86 Tokyo, pp. 40–43.

BAHL L.R., JELINEK F. and MERCER R.L. (1983) A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence, **5**, pp. 179–190.

BAUM L.E. and EAGON J.E. (1967) An Inequality with Applications to Statistical Estimation for Probabilistic Functions of a Markov Process and to Models for Ecology. Bull. AMS, **73**, pp. 360-363.

BAUM L.E. and PETRIE T. (1966) Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Ann. Math. Stat. **37**, pp. 1554–1563.

BAUM L.E., PETRIE T., SOULES G. and WEISS N. (1970) A Maximisation Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Ann. Math. Stat., **41**, pp. 164–171.

CONNER P.N. (1993) Hidden Markov Model with Improved Observation and Duration Modelling. PhD. Thesis, University of East Anglia, England.

EPHARAIM Y., MALAH D. and JUANG B.H.(1989) On Application of Hidden Markov Models for Enhancing Noisy Speech. IEEE Trans. Acoustics Speech and Signal Processing, **37(12)**, pp. 1846-1856, Dec.

FORNEY G.D. (1973) The Viterbi Algorithm. Proc. IEEE, **61**, pp. 268–278.

GALES M.J.F. and YOUNG S.J. (1992) An Improved Approach to the Hidden Markov Model Decomposition of Speech and Noise. Proc. IEEE, Int. Conf. on Acoust., Speech, Signal Processing, ICASSP-92, pp. 233–235.

GALES M.J.F. and YOUNG S.J. (1993) HMM Recognition in Noise using Parallel Model Combination. Eurospeech-93, pp. 837–840.

HUANG X.D., ARIKI Y. and JACK M.A. (1990) Hidden Markov Models for Speech Recognition. Edinburgh University Press, Edinburgh.

HUANG X.D. and JACK M.A. (1989) Unified Techniques for Vector Quantisation and Hidden Markov Modelling using Semi-Continuous Models. IEEE Proc. Acoustics, Speech and Signal Processing, ICASSP-89 Glasgow, pp. 639–642.

JELINEK F. and MERCER R. (1980) Interpolated Estimation of Markov Source Parameters from Sparse Data. Proc. of the Workshop on Pattern Recognition in Practice. North-Holland, Amesterdam.

JELINEK F, (1976) Continuous Speech Recognition by Statistical Methods. Proc. of IEEE, **64**, pp. 532–555.

JUANG B.H. (1985) Maximum-Likelihood Estimation for Mixture Multi-Variate Stochastic Observations of Markov Chain. AT&T Bell laboratories Tech J., 64, pp. 1235–1249.

JUANG B.H. (1984) On the Hidden Markov Model and Dynamic Time

Warping for Speech Recognition- A unified Overview. AT&T Technical J., **63**, pp. 1213–1243.

KULLBACK S. and LEIBLER R.A. (1951) On Information and Sufficiency. Ann. Math. Stat., **22**, pp. 79–85.

LEE K.F. (1989) Automatic Speech Recognition: the Development of SPHINX System. MA: Kluwer Academic Publishers, Boston.

LEE K.F. (1989) Hidden Markov Model: Past, Present and Future. Eurospeech-89, Paris.

LIPORACE L.R. (1982) Maximum Likelihood Estimation for Multi-Variate Observations of Markov Sources. IEEE Trans. IT, **IT-28**, pp. 729–735.

MARKOV A.A. (1913) An Example of Statistical Investigation in the text of *Eugen Onyegin* Illustrating Coupling of Tests in Chains. Proc. Acad. Sci. St Petersburg VI Ser., **7**, pp. 153–162.

MILNER B.P. (1995) Speech Recognition in Adverse Environments, PhD. Thesis, University of East Anglia, England.

PETERIE T. (1969) Probabilistic Functions of Finite State Markov Chains. Ann. Math. Stat., **40**, pp. 97–115.

RABINER L.R. and JUANG B.H. (1986) An Introduction to Hidden Markov Models. IEEE ASSP. Magazine, pp. 4–15.

RABINER L.R., JUANG B.H., LEVINSON S.E. and SONDHI M.M., (1985) Recognition of Isolated Digits using Hidden Markov Models with Continuous Mixture Densities. AT&T Technical Journal, **64**, pp. 1211-1235.

RABINER L.R. and JUANG B.H. (1993) Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NJ.

YOUNG S.J. (1999), HTK: Hidden Markov Model Tool Kit. Cambridge University Engineering Department.

VARGA A. and MOORE R.K., Hidden Markov Model Decomposition of Speech and Noise. in Proc. IEEE Int., Conf. on Acoust., Speech, Signal Processing, 1990, pp. 845–848

VITERBI A.J. (1967) Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. IEEE Trans. on Information theory, **IT-13**, pp. 260–269.