Hidden semi-Markov models
Shun-Zheng Yu

Department of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou 510275, PR China

A R T I C L E   I N F O

Article history:
Received 14 April 2009
Available online 17 November 2009

Keywords:
Hidden Markov model (HMM)
Hidden semi-Markov model (HSMM)
Explicit duration HMM
Variable duration HMM
Forward–backward (FB) algorithm
Viterbi algorithm

A B S T R A C T

As an extension to the popular hidden Markov model (HMM), a hidden semi-Markov model (HSMM) allows the underlying stochastic process to be a semi-Markov chain. Each state has variable duration and a number of observations being produced while in the state. This makes it suitable for use in a wider range of applications. Its forward–backward algorithms can be used to estimate/update the model parameters, determine the predicted, filtered and smoothed probabilities, evaluate goodness of an observation sequence fitting to the model, and find the best state sequence of the underlying stochastic process. Since the HSMM was initially introduced in 1980 for machine recognition of speech, it has been applied in thirty scientific and engineering areas, such as speech recognition/synthesis, human activity recognition/prediction, handwriting recognition, functional MRI brain mapping, and network anomaly detection. There are about three hundred papers published in the literature. An overview of HSMMs is presented in this paper, including modelling, inference, estimation, implementation and applications. It first provides a unified description of various HSMMs and discusses the general issues behind them. The boundary conditions of HSMM are extended. Then the conventional models, including the explicit duration, variable transition, and residential time of HSMM, are discussed. Various duration distributions and observation models are presented. Finally, the paper draws an outline of the applications.

1. Introduction (History)

A hidden Markov model (HMM) is defined as a doubly stochastic process. The underlying stochastic process is a discrete-time finite-state homogeneous Markov chain. The state sequence is not observable and so is called hidden. It influences another stochastic process that produces a sequence of observations. An excellent tutorial of HMMs can be found in Rabiner [150], a theoretic overview of HMMs can be found in Ephraim and Merhav [57] and a discussion on learning and inference in HMMs in understanding of Bayesian networks is presented in Ghahramani [66]. The HMMs are an important class of models that are successful in many application areas. However, due to the non-zero probability of self-transition of a non-absorbing state, the state duration of an HMM is implicitly a geometric distribution. This makes the HMM has limitations in some applications.

As an extension of the HMM, a hidden semi-Markov model (HSMM) is traditionally defined by allowing the underlying process to be a semi-Markov chain. Each state has a variable duration, which is associated with the number of observations produced while in the state. The HSMM is also called “explicit duration HMM” [60,150], “variable-duration HMM” [107, 155,150], “HMM with explicit duration” [124], “hidden semi-Markov model” [126], generalized HMM [94], segmental HMM [157] and segment model [135,136] in the literature, depending on their assumptions and their application areas.
The first approach to hidden semi-Markov model was proposed by Ferguson [60], which is partially included in the survey paper by Rabiner [150]. This approach is called the explicit duration HMM in contrast to the implicit duration of the HMM. It assumes that the state duration is generally distributed depending on the current state of the underlying semi-Markov process. It also assumes the “conditional independence” of outputs. Levinson [107] replaced the probability mass functions of duration with continuous probability density functions to form a continuously variable duration HMM. As Ferguson [60] pointed out, an HSMM can be realized in the HMM framework in which both the state and its sojourn time since entering the state are taken as a complex HMM state. This idea was exploited in 1991 by a 2-vector HMM [93] and a duration-dependent state transition model [179]. Since then, similar approaches were proposed in many applications. They are called in different names such as inhomogeneous HMM [151], non-stationary HMM [164], and recently triplet Markov chains [144]. These approaches, however, have the common problem of computational complexity in some applications. A more efficient algorithm was proposed in 2003 by Yu and Kobayashi [199], in which the forward–backward variables are defined using the notion of a state together with its remaining sojourn (or residual life) time. This makes the algorithm practical in many applications.

The HSMM has been successfully applied in many areas. The most successful application is in speech recognition. The first application of HSMM in this area was made by Ferguson [60]. Since then, there have been more than one hundred such papers published in the literature. It is the application of HSMM in speech recognition that enriches the theory of HSMM and develops many algorithms for HSMM.

Since the beginning of 1990’s, the HSMM started being applied in many other areas such as electrocardiograph (ECG) [174], printed text recognition [4] or handwritten word recognition [95], recognition of human genes in DNA [94], language identification [118], ground target tracking [88], document image comparison and classification at the spatial layout level [81], etc.

In recent years from 2000 to present, the HSMM has been obtained more and more attentions from vast application areas such as change-point/end-point detection for semi-conductor manufacturing [64], protein structure prediction [162], mobility tracking in cellular networks [197], analysis of branching and flowering patterns in plants [69], rain events time series model [159], brain functional MRI sequence analysis [58], satellite propagation channel modelling [112], Internet traffic modelling [198], event recognition in videos [79], speech synthesis [204,125], image segmentation [98], semantic learning for a mobile robot [167], anomaly detection for network security [201], symbolic plan recognition [54], terrain modelling [185], adaptive cumulative sum test for change detection in non-invasive mean blood pressure trend [193], equipment prognosis [14], financial time series modelling [22], remote sensing [147], classification of music [113], and prediction of particulate matter in the air [52], etc.

The rest of the paper is organized as follows: Section 2 is the major part of this paper that defines a unified HSMM and addresses important issues related to inference, estimation and implementation. Section 3 then presents three conventional HSMMs that are applied vastly in practice. Section 4 discusses the specific modelling issues, regarding duration distributions, observation distributions, variants of HSMMs, and the relationship to the conventional HMM. Finally, Section 5 highlights major applications of HSMMs and concludes the paper in Section 6.

2. Hidden semi-Markov model

This section provides a unified description of HSMMs. A general HSMM is defined without specific assumptions on the state transitions, duration distributions and observation distributions. Then the important issues related to inference, estimation and implementation of the HSMM are discussed. A general expression of the explicit-duration HMMs and segment HMMs can be found in Murphy [126], and a unified view of the segment HMMs can be found in Ostendorf et al. [136]. Detailed review for the conventional HMM can be found in the tutorial by Rabiner [150], the overview by Ephraim and Merhav [57], the Bayesian networks-based discussion by Ghahramani [66], and the book by Cappe et al. [29].

2.1. General model

A hidden semi-Markov model (HSMM) is an extension of HMM by allowing the underlying process to be a semi-Markov chain with a variable duration or sojourn time for each state. Therefore, in addition to the notation defined for the HMM, the duration $d$ of a given state is explicitly defined for the HSMM. State duration is a random variable and assumes an integer value in the set $D = \{1, 2, \ldots, D\}$. The important difference between HMM and HSMM is that one observation per state is assumed in HMM while in HSMM each state can emit a sequence of observations. The number of observations produced while in state $i$ is determined by the length of time spent in state $i$, i.e., the duration $d$. Now we provide a unified description of HSMMs.

Assume a discrete-time Markov chain with the set of (hidden) states $S = \{1, \ldots, M\}$. The state sequence is denoted by $S_{1:T} = s_1, s_2, \ldots, s_T$, where $S_t \in S$ is the state at time $t$. A realization of $S_{1:T}$ is denoted as $s_{1:T}$. For simplicity of notation in the following sections, we denote:

- $S_{t_1:t_2} = i$ – state $i$ that the system stays in during the period from $t_1$ to $t_2$. In other words, it means $S_{t_1} = i$, $S_{t_1+1} = i$, ..., and $S_{t_2} = i$. Note that the previous state $S_{t_1-1}$ and the next state $S_{t_2+1}$ may or may not be $i$. 

Then the set of the model parameters for the HSMM is defined by

\[ \lambda = \{ \pi, a(j,d), b(j,d,v_{k_1:k_d}), \gamma_{j,d} \} \]

where \( i, j \in S, d, d' \in D, \) and \( v_{k_1:k_d} \) represents \( v_{k_1}, \ldots, v_{k_d} \in \mathcal{V} \times \cdots \times \mathcal{V} \). This general HSMM is shown in Fig. 1.

The general HSMM is reduced to specific models of HSMM depending on the assumptions they made. For instance:

- If the state duration is assumed to be independent to the previous state, then the state transition probability can be further specified as
  \[ a(i,d') = a(i,d') \mathbb{P}(d) \]
  where
  \[ a(i,d') \triangleq \mathbb{P}(S[t+1] = j | S[t-d'+1:t] = i) \] (1)
is the transition probability from state $i$ that has stayed for duration $d'$ to state $j$ that will start at $t+1$, and

$$p_{j}(d) \triangleq P[S_{t+1:t+d} = j | S_{t+1} = j]$$

(2)

is the probability of duration $d$ that state $j$ takes. This is the model proposed by Marhasev et al. [119].

- If a state transition is assumed to be independent to the duration of the previous state, then the state transition probability becomes $a_{i(i',d')j,j} = a_{i,j}$, where

$$a_{i,j} \triangleq P[S_{t+1:t+d} = j | S_{t} = i]$$

(3)

is the transition probability that state $i$ ended at $t$ and transits to state $j$ having duration $d$. This is the posterior time HMM (see Section 3.3 for details). In this model, a state transition for $i \neq j$ is $(i, t) \rightarrow (j, \tau)$ and a self-transition is assumed to be $(i, t) \rightarrow (i, t-1)$ for $\tau > 1$, where $\tau$ represents the residential time of the state.

- If a self-transition is allowed and is assumed to be independent to the previous state, then the state transition probability becomes

$$a_{i(i',d')j,j} = a_{i,j} \prod_{\tau=1}^{d-1} a_{jj}[1 - a_{jj}(d)].$$

where $a_{jj}(d) \triangleq P[S_{t+1:t+d} = j | S_{t+1:t+d} = j]$ is the self-transition probability when state $j$ has stayed for $d$ time units, and $1 - a_{jj}(d) = P[S_{t+1:t+d} = j | S_{t+1:t+d} = j]$ is the probability state $j$ ends with duration $d$. This is the variable transition HMM (see Section 3.2 for details). In this model, a state transition is either $(i, d) \rightarrow (j, 1)$ for $i \neq j$ or $(i, d) \rightarrow (i, d+1)$ for a self-transition.

- If a transition to the current state is independent to the duration of the previous state and the duration is only conditioned on the current state, then $a_{i(i',d')j,j} = a_{ij}p_{j}(d)$, where $a_{ij} \triangleq P[S_{t+1} = j | S_{t} = i]$ is the transition probability from state $i$ to $j$, with the self-transition probability $a_{ii} = 0$. This is the explicit duration HMM (see Section 3.1 for details).

Besides, the state duration distributions, $p_{j}(d)$, can be parametric or non-parametric. The detailed discussion on various duration distributions can be found in Section 4.1. Similarly, the observation distributions $b_{j,d}(v_{k},q_{k})$ can be parametric or non-parametric, discrete or continuous, and dependent or independent on the state durations. It can also be a mixture of distributions. The detailed discussion on various observation distributions can be found in Section 4.2.

2.2. Inference

In this subsection we discuss the issues related to inference, including the forward–backward algorithm, calculation of probabilities and expectations, maximum a posteriori (MAP) estimate of states, maximum likelihood estimate (MLE) of state sequence, and constrained estimate of states.

2.2.1. The forward–backward algorithm

We define the forward variables for HSMM by:

$$\alpha_{t}(j, d) \triangleq P[S_{t:t+d} = j, o_{1:t} | \lambda]$$

and the backward variables by

$$\beta_{t}(j, d) \triangleq P[o_{t+1:T} | S_{t:t+d} = j, \lambda].$$

Similar to deriving the formulas for the HMM (see e.g., Rabiner [150], Ephraim and Merhav [57]), it is easy to obtain the forward–backward algorithm for a general HSMM:

$$\alpha_{t}(j, d) = \sum_{i \in S} \sum_{d' \in D} \alpha_{t-1}(i, d') \cdot a_{i,j}(d', j,d) \cdot b_{j,d}(o_{t+1:t+d}) \cdot \beta_{t+1}(i, d'),$$

(4)

for $t > 0$, $d \in D$, $j \in S$, and

$$\beta_{t}(j, d) = \sum_{i \in S} \sum_{d' \in D} a_{i,j}(d', j,d) \cdot b_{i,d}(o_{t+1:t+d'}) \cdot \beta_{t+1}(i, d'),$$

(5)

for $t < T$.

The initial conditions generally can have two different assumptions:

- The general assumption: assumes that the first state begins at or before observation $o_{1}$ and the last state ends at or after observation $o_{T}$. In this case, we can assume that the process starts at $-\infty$ and terminates at $+\infty$. The observations out of the sampling period $[1, T]$ can be any possible values, i.e., $b_{j,d}(\cdot) = 1$ for any $j \in S$, $d \in D$. Therefore, in the forward
formula (4) \( b_{j,d}(o_{t-d+1:t}) \) is replaced with the marginal distribution \( b_{j,d}(o_{1:t}) \) if \( t - d + 1 \leq 1 \) and \( t \geq 1 \), and in the backward formula (5) \( b_{j,d}(o_{t+1:t+d}') \) is replaced with \( b_{j,d}(o_{t+1:t}) \) if \( t + 1 \leq T \) and \( t + d' \geq T \). We then have the initial conditions for the forward recursion formula given by (4) as follows:

\[
\alpha_t(j, d) = P[S_{t-d+1:t} = j|\lambda] = \pi_{j,d}, \quad \tau \leq 0, \ d \in \mathcal{D},
\]

where \( \{\pi_{j,d}\} \) can be the equilibrium distribution of the underlying semi-Markov process. Because, for \( t + d' \geq T \),

\[
P[S_{t+1:t+d'} = 1, \alpha_{t+1:T} | S_{t-d+1:t} = j, \lambda] \triangleq a_{i,d}(i,d')b_{i,d}(o_{t+1:t})
\]

then from the backward recursion formula (5) we can see that \( \beta_t(i, d') = 1 \), for \( t + d' \geq T \). Therefore, the initial conditions for the backward recursion formula given by (5) are as follows:

\[
\beta_t(i, d) = 1, \quad \tau \geq T, \ d \in \mathcal{D}.
\]

If the model assumes that the first state begins at \( t = 1 \) and the last state ends at or after observation \( o_T \), it is a right-censored HMM introduced by Guedon [70]. Because this is desirable for many applications, it is taken as a basis for an R package for analyzing HSMMs [23].

- **The simplifying assumption**: assumes that the first state begins at time 1 and the last state ends at time \( T \). This is the most popular assumption one can find in the literature. In this case, the initial conditions for the forward recursion formula given by (4) are:

\[
\alpha_0(j, d) = \pi_{j,d}, \quad d \in \mathcal{D},
\]

\[
\alpha_{T}(j, d) = 0, \quad \tau < 0, \ d \in \mathcal{D},
\]

and the initial conditions for the backward recursion formula given by (5) are:

\[
\beta_{1}(i, d) = 1, \quad d \in \mathcal{D},
\]

\[
\beta_{T}(i, d) = 0, \quad \tau > T, \ d \in \mathcal{D}.
\]

Note that the initial distribution of states can be assumed as \( \pi'_{j,d} \equiv P[S_{1:d} = j|\lambda] \), which obviously equals to \( \sum_{i,d} \pi_{i,d} a_{i,d}(j,d) \). Therefore, the initial conditions for the forward recursion formula can also be \( \alpha_{d}(j, d) = \pi'_{j,d} b_{j,d}(o_{1:d}), \) for \( d \in \mathcal{D}. \)

### 2.2.2 Probabilities and expectations

- After the forward variables \( \{\alpha_t(j, d)\} \) and the backward variables \( \{\beta_t(j, d)\} \) are determined, all other probabilities of interest can be computed. For instance, the filtered probability that state \( j \) started at \( t - d + 1 \) and ends at \( t \), with duration \( d \), given partial observed sequence \( o_{1:t} \) can be determined by

\[
P[S_{t-d+1:t} = j|o_{1:t}, \lambda] = \frac{\alpha_t(j, d)}{\sum_{j,d} \alpha_t(j, d)}
\]

and the predicted probability that state \( j \) will start at \( t + 1 \) and end at \( t + d \), with duration \( d \), given partial observed sequence \( o_{1:t} \) by

\[
P[S_{t+1:t+d} = j|o_{1:t}, \lambda] = \frac{\sum_{j,d} \alpha_t(i, d')a_{i,d'}(j,d)\beta_{t+d}(j,d)}{\sum_{i,d} \alpha_t(i, d')}. \tag{6}
\]

These readily yield the filtered probability of state \( j \) ending at \( t \), \( P[S_{t} = j|o_{1:T}, \lambda] = \sum_{d} P[S_{t-d+1:t} = j|o_{1:t}, \lambda] \), and the predicted probability of state \( j \) starting at \( t + 1 \), \( P[S_{t+1} = j|o_{1:T}, \lambda] = \sum_{d} P[S_{t+1:t+d} = j|o_{1:t}, \lambda] \).

The posterior probabilities \( P[S_{t} = j|o_{1:T}, \lambda], P[S_{t+1} = j|o_{1:T}, \lambda] \) and \( P[S_{t-d+1:t} = j|o_{1:T}, \lambda] \) for given entire observation sequence \( o_{1:t} \) can be determined by the following equations

\[
\eta_t(j, d) \triangleq P[S_{t-d+1:t} = j, o_{1:T}|\lambda] = \alpha_t(j, d)\beta_t(j, d), \tag{6}
\]

\[
\xi_t(i, d': j, d) \triangleq P[S_{t-d'+1:t} = i, S_{t-t+1} = j, o_{1:T}|\lambda] = \alpha_t(i, d')a_{i,d'}(j,d)\beta_{t+d}(j,d), \tag{7}
\]

\[
\xi_t(i, j) \triangleq P[S_{t} = i, S_{t+1} = j, o_{1:T}|\lambda] = \sum_{d' \in \mathcal{D}} \sum_{d \in \mathcal{D}} \xi_t(i, d': j, d), \tag{7}
\]

\[
\gamma_t(j) \triangleq P[S_{t} = j, o_{1:T}|\lambda] = \sum_{d' \geq t} \sum_{d = t - d + 1}^{D} \eta_t(j, d)
\]

and
\[ P[o_{1:T}|\lambda] = \sum_{j \in S} P[S_t = j, o_{1:T}|\lambda] = \sum_{j \in S} \gamma_t(j), \]

for \( d, d' \in D, j \in S, i \in S \setminus \{j\} \) and \( t = 1, \ldots, T \), where \( \eta_t(j, d)/P[o_{1:T}|\lambda] \) represents the probability of being in state \( j \) having duration \( d \) by time \( t \) given the model and the observation sequence; \( \xi_t(i, d'; j, d)/P[o_{1:T}|\lambda] \) the probability of transition at time \( t \) from state \( i \) occurred with duration \( d' \) to state \( j \) having duration \( d \) given the model and the observation sequence; \( \gamma_t(i, j)/P[o_{1:T}|\lambda] \) the probability of transition at time \( t \) from state \( i \) given the model and the observation sequence; \( \gamma_t(j)/P[o_{1:T}|\lambda] \) the probability of state \( j \) at time \( t \) given the model and the observation sequence; and \( P[o_{1:T}|\lambda] \) the probability that the observed sequence \( o_{1:T} \) is generated by the model \( \lambda \). Obviously, the conditional factor \( P[o_{1:T}|\lambda] \) is common for all the posterior probabilities, which will be eliminated when the posterior probabilities are used in parameter estimation. Therefore, it is often omitted for simplicity in the literature. Similarly, in the rest of this paper, we sometimes will not explicitly mention this conditional factor in calculating the posterior probabilities by \( \eta_t(j, d), \xi_t(i, d'; j, d), \gamma_t(i, j) \), and \( \gamma_t(j) \).

In considering the following identity
\[
P[S_{t+1} = j, o_{1:T}|\lambda] = P[S_t = j, o_{1:T}|\lambda] - P[S_t = j, o_{1:T}|\lambda] \]
\[
P[S_{t+1} = j, o_{1:T}|\lambda] = P[S_{t+1} = j, o_{1:T}|\lambda] - P[S_{t+1} = j, o_{1:T}|\lambda] \]

we have a recursive formula for calculating \( \gamma_t(j) \):
\[
\gamma_t(j) = \gamma_{t+1}(j) + P[S_t = j, o_{1:T}|\lambda] - P[S_{t+1} = j, o_{1:T}|\lambda] = \gamma_{t+1}(j) + \sum_{i \in S \setminus \{j\}} [\xi_t(i, j) - \xi_t(i, j)].
\]

Denote \( P[o_{1:T}|\lambda] \) by \( L \) in the following expressions. Then using the forward and backward variables, one can compute various expectations [60]:

(a) The expected number of times state \( i \) ends before \( t \):
\[
\frac{1}{T} \sum_{t' = 1}^{t-1} \sum_{j \in S \setminus \{i\}} \xi_t(i, j).
\]
(b) Expected total duration spent in state \( i \):
\[
\frac{1}{T} \sum_{t'} \gamma_t(i).
\]
(c) Expected number of times that state \( i \) occurred with observation \( o_t = v_k \):
\[
\frac{1}{T} \sum_{t} \gamma_t(i) I(o_t = v_k),
\]
where the indicator function \( I(x) = 1 \) if \( x \) is true and zero otherwise.
(d) Estimated average observable values of state \( i \):
\[
\frac{1}{T} \sum_{t} \frac{\gamma_t(i) D_t}{\sum_{j} \gamma_t(j)}.
\]
(e) Probability that state \( i \) was the first state:
\[
\frac{1}{T} \gamma_t(i).
\]
(f) Expected total number of times state \( i \) commenced:
\[
\frac{1}{T} \sum_{t} \sum_{j \in S \setminus \{i\}} \xi_t(i, j) \text{ or terminated: } \frac{1}{T} \sum_{t} \sum_{j \in S \setminus \{i\}} \xi_t(i, j).
\]
For the simplifying assumption for the boundary conditions described in the last subsection, we have
\[
\sum_{t=0}^{T-1} \sum_{j \in S \setminus \{i\}} \xi_t(i, j) = \sum_{t=1}^{T} \sum_{j \in S \setminus \{i\}} \xi_t(i, j).
\]
(g) Estimated average duration of state \( i \):
\[
\frac{1}{T} \sum_{t} \frac{\gamma_t(i) d_t}{\sum_{d} \gamma_t(i) d_t}.
\]

2.2.3. MAP and MLE estimate of states

The maximum a posteriori (MAP) estimate of state \( S_t \) given a specific observation sequence \( o_{1:T} \) can be obtained [60] by maximizing \( \gamma_t(j) \) given by (8), i.e.,
\[
\hat{s}_t = \arg \max_{i \in S} \{ \gamma_t(i) \}.
\]

If we choose \( \eta_t(i, d) \) of (6), instead of \( \gamma_t(i) \), as the MAP criterion, we obtain the joint MAP estimate of the state that ends at time \( t \) and the duration of this state, when a specific sequence \( o_{1:T} \) is observed:
\[
(\hat{s}_t, \hat{d}_t) = \arg \max_{(i,d)} \eta_t(i, d).
\]

(10)

Viterbi algorithms are the most popular dynamic programming algorithms for the maximum likelihood estimate (MLE) of state sequence of HMMs. There exist the similar algorithms for the HSMM [115,151,35,26]. Define the forward variable for the extended Viterbi algorithm by
\[
\delta_t(j, d) \triangleq \max_{s_{t-d}} P[s_{t-d}, s_{t-d+1:s} = j, o_{1:s}|\lambda] = \max_{i \in S \setminus \{j\}, d' \in D \setminus \{d\}} \{\delta_{t-d}(i, d') a(i,d')(j,d) b(j,a(o_{t-d+1:s}))\},
\]
for \( 1 \leq t \leq T, j \in S, d \in D \). \( \delta_t(j, d) \) represents the maximum likelihood that the partial state sequence ends at \( t \) in state \( j \) of duration \( d \). Record the previous state that \( \delta_t(j, d) \) selects by \( \Psi(t, j, d) \triangleq (t-d, t^*, d^*) \), where \( t^* \) is the previous state survived, \( d^* \) its duration, and \( t^* \) its ending time. \( \Psi(t, j, d) \) is determined by letting
(i^*, d^*) = \arg \max_{i \in S, d \in D} \left\{ \delta_t - d(i, d) a(i, d') b(j, d') (o_{t-d+1:t}) \right\}.

Now the maximum likelihood state sequence can be determined by finding the last state that maximizes the likelihood. For the general assumption of the boundary conditions on page 5, the last ML state is

\[ (t_1, j_1^*, d_1^*) = \arg \max_{t \geq T \atop d \geq t - T + 1, d \in D} \delta_t (i, d), \]

or, for the simplifying assumption of the boundary conditions, \( t_1 = T \) and

\[ (j_1^*, d_1^*) = \arg \max_{i \in S} \delta_T (i, d). \]

Trace back the state sequence by letting

\[ (t_2, j_2^*, d_2^*) = \Psi (t_1, j_1^*, d_1^*), \]

\[ \ldots \]

\[ (t_n, j_n^*, d_n^*) = \Psi (t_{n-1}, j_{n-1}^*, d_{n-1}^*), \]

until the first state \( S_1 \) is determined, where \( S_1 = j_1^* \) and \( (j_1^*, d_1^*), \ldots, (j_n^*, d_n^*) \) is the maximum likelihood state sequence.

If the state duration density function is log-convex parametric, which is fulfilled by the commonly used parametric functions, Bonafonte et al. [17] empirically showed that the computational complexity can be reduced to about 3.2 times of the conventional HMM. If the model is a left–right HSMM or the particular state sequence, interpreted as the probabilities that the path taken (a random variable) passes through the constraint states for the given observations. It is expected that the posterior probabilities \( P \) for given entire observation sequence \( o_{1:T} \) are used for the computation of various expectations and the MAP estimation of states. These posterior probabilities can be defined as the probabilities that the path taken (a random variable) passes through the constraint states for the first state \( S_1 \) is determined, where \( S_1 = j_1^* \) and \( (j_1^*, d_1^*), \ldots, (j_n^*, d_n^*) \) is the maximum likelihood state sequence.

2.2.4. Constrained estimate of states

As discussed in the previous subsections, the posterior probabilities \( P[S_t = j| o_{1:T}, \lambda] \), \( P[S_t = i, S_{t+1} = j| o_{1:T}, \lambda] \) and \( P[S_{t-d+1:t} = j| o_{1:T}, \lambda] \) for given entire observation sequence \( o_{1:T} \) are determined by the forward–backward algorithm, and are used for the computation of various expectations and the MAP estimation of states. These posterior probabilities can be interpreted as the probabilities that the path taken (a random variable) passes through the constraint states for the given observation sequence \( o_{1:T} \). For example, \( P[S_{t-d+1:t} = j, o_{1:T}\lambda] = \alpha_t(j, d) \beta_t(j, d) \) counts for all the paths that pass through the constraint state \( j \) during the constraint period of \( t - d + 1 \) to \( t \), where \( \alpha_t(j, d) \) is given by (4) and \( \beta_t(j, d) \) by (5). This is useful for the confidence calculation in the state estimation. The confidence can be simply defined as

\[ \delta_t (i_m, d) = \max_{d' \in D} \delta_t - d (i_m, d') a_{(i_m, d')} (i_m, d) b_{i_m, d} (o_{t-d+1:t}), \]

for \( 1 \leq m \leq n, 1 \leq t \leq T, d \in D \).

2.2.4. Constrained estimate of states

As discussed in the previous subsections, the posterior probabilities \( P[S_t = j| o_{1:T}, \lambda] \), \( P[S_t = i, S_{t+1} = j| o_{1:T}, \lambda] \) and \( P[S_{t-d+1:t} = j| o_{1:T}, \lambda] \) for given entire observation sequence \( o_{1:T} \) are determined by the forward–backward algorithm, and are used for the computation of various expectations and the MAP estimation of states. These posterior probabilities can be interpreted as the probabilities that the path taken (a random variable) passes through the constraint states for the given observation sequence \( o_{1:T} \). For example, \( P[S_{t-d+1:t} = j, o_{1:T}\lambda] = \alpha_t(j, d) \beta_t(j, d) \) counts for all the paths that pass through the constraint state \( j \) during the constraint period of \( t - d + 1 \) to \( t \), where \( \alpha_t(j, d) \) is given by (4) and \( \beta_t(j, d) \) by (5). This is useful for the confidence calculation in the state estimation. The confidence can be simply defined as

\[ \max_{j,d} \alpha_t(j, d) \beta_t(j, d) / \sum_{j',d'} \alpha_t(j', d') \beta_t(j', d'), \]

where \( \alpha_t(j, d) \beta_t(j, d) \) is used for the MAP estimate of the state as given by (10). Calculating the confidence over every \( t, o \) can find out in practice when the errors are most likely to occur in the state estimation.

Now we compute the posterior probability \( P[S_{t+1:t+k}| o_{1:T}, \lambda] \) corresponding to a segment of \( k \) observations. It is expected useful in some applications. For example, this probability can be used to estimate the confidence of an individual field of words for information extraction [41]. It can also be used for computing some expectations, which can be used for estimating the states corresponding to the segment of \( k \) observations.

If one assumes that the first state of the subsequence \( S_{t+1:t+k} \) starts at \( t + 1 \) and the last state ends at \( t + k \), i.e., \( S_{t+1:t+k} = \{ j_1, d_1 \} \ldots \{ j_n, d_n \} \), s.t. \( d_1 + \ldots + d_n = k \), it is easy to compute the posterior probability by

\[ P[S_{t+1:t+k}, o_{1:T}\lambda] = \alpha_t + d_1 (j_1, d_1) \left( \prod_{m=2}^n a_{(j_{m-1}, d_{m-1}) (j_m, d_m)} b_{j_m, d_m} (o_{t-d_{m-1}+1:t}) \right) \beta_{t+k} (j_n, d_n), \]

where \( t_m = t + d_1 + \ldots + d_m \). When \( n = 1 \) the equation is reduced to (6). If we release the condition that the first state of the subsequence starts at \( t + 1 \) or the one that the last state of the subsequence ends at \( t + k \), the computation can be done by allowing the first state duration \( d \geq d_1 \) and the last state duration \( d' \geq d_n \).

A simple way for computing the posterior probability of a segment is modifying the forward–backward algorithm to conform the constraints. Similar to the constrained forward–backward algorithm for a CRF (conditional random field) proposed by Culotta and McCallum [41], the forward–backward formulas given by (4) and (5) for the HSM can be modified. Let \( \alpha_t(j, d) = 0 \) and \( \beta_t(j, d) = 0 \) when \( t + 1 \leq t' \leq t + k \) and \( j \neq j_t \), where \( j_t \in S_{t+1:t+k} \) is a constraint that each path must pass through and \( S_{t+1:t+k} \) is the constraint subpath. If we further constrain that the first state of the constraint subpath starts at
2.3. Estimation

In the preceding problems, such as the forward–backward algorithm, MAP and ML estimation of states, we assumed that the set of model parameters is given. If \( \lambda \) is unknown, we need to learn about \( \lambda \) from the observations \( o_{1:T} \). \( \lambda \) is initially estimated and then re-estimated so that the likelihood function \( L(\lambda) = P[o_{1:T} | \lambda] \) increases and converges to its maximum value. If the system is slowly varying (i.e., non-stationary), the model parameters \( \lambda \) may need to be updated adaptively. Such training and updating process is referred to as parameter re-estimation.

2.3.1. Parameter estimate of HSMM

For the parameter estimation/re-estimation problem, there is no known analytical method to find the \( \lambda \) that maximizes the likelihood function. Thus, some iterative procedure must be employed.

The model parameters \( \lambda \) can be re-estimated using the expectations. For instance,

1) the initial distribution \( \tilde{\pi}_{j,d} \) can be updated by \( \eta_{j}(j,d) / \sum_{j,d} \eta_{j}(j,d) \) for \( t \leq 0 \),
2) the transition probabilities \( \tilde{a}_{ij}(j,d) \) by \( \sum_{i,d} \xi_{i}(,i,j) / \sum_{i,d} \sum_{j'=d} \xi_{i}(,i',j') \) and \( \eta_{j}(d,j) / \sum_{d} \sum_{j} \eta_{j}(d,j) \), and
3) the observation probabilities \( \tilde{b}_{j,d}(v_{k_{1:k_{d}}}) \) by \( \sum_{j} \eta_{j}(j,d) / \sum_{j} \sum_{d} \eta_{j}(d,j) \).

Except these parameters for the general HSMM, parameters for other HSMMs can be estimated as well, such as

i) the transition probabilities \( \hat{a}_{ij} \) by \( \sum_{i,j} \xi_{i}(i,j) / \sum_{i,d} \sum_{j} \xi_{i}(i',j') \),
ii) the duration probabilities \( \hat{p}_{j}(d) \) of state \( j \) by \( \sum_{d} \eta_{j}(d,j) / \sum_{d} \sum_{j} \eta_{j}(d,j) \),
iii) the observation probabilities \( \hat{b}_{j}(v_{k}) \) by \( \sum_{j} \gamma_{j}(j) \cdot \sum_{d} \sum_{j} \gamma_{j}(j) \), and
iv) the initial distribution \( \hat{\pi}_{j} \) by \( \gamma_{o}(j) / \sum_{j} \gamma_{o}(j) \).

Those probability mass function or probability density function satisfy: \( \sum_{j,d} \hat{\pi}_{j,d} = 1 \), \( \sum_{d} \hat{p}_{j}(d) = 1 \), \( \sum_{j',d} \hat{a}_{j',d}(j',d) = 1 \), \( \sum_{j} \hat{b}_{j}(v_{k}) = 1 \).

The re-estimation procedure:

a) Assume an initial model parameter set \( \lambda_0 \);
b) For given model parameter set \( \lambda_k \), use the forward–backward formulas (4) and (5) to compute the forward and backward variables \( \alpha_{t}(j,d) \) and \( \beta_{t}(j,d) \). Then use the forward and backward variables to compute the related probabilities \( \eta_{j}(j,d) \), \( \xi_{i}(i,d'; j, j) \), \( \xi_{i}(i,j) \) and \( \gamma_{j}(j) \) by (6) through (9). Finally re-estimate the model parameters to get \( \hat{\lambda}_{k+1} \);
c) Let \( \lambda_{k+1} = \hat{\lambda}_{k+1} \), \( k = + \), and go back to step b);
d) Repeat b) and c) until the likelihood \( L(\lambda_k) = P[o_{1:T} | \lambda_k] \) converges to a fixed point.

2.3.2. Order estimate of HSMM

In the re-estimation algorithms discussed above, the number of hidden states, \( M \), the maximum length of state duration, \( D \), the number of observable values, \( K \), and the length of the observation sequence, \( T \), are usually assumed known in the context of applications. However, the learning issues when the order of an HSMM is unknown is sometimes particularly important in practice. A detailed discussion on the order estimate of HMMs can be found in Ephraim and Merhav [57, Section VIII], and the issues of overfitting and model selection in Grahame-Mani [66, Section 7]. However, the order estimate of HSMMs is somewhat different from that of HMMs, because HSMMs have variable durations. Therefore, for an HSMM we must estimate both the number of states, \( M \), and the maximum length of state durations, \( D \).

In fact, some special HSMMs can be described by a dynamic Bayesian network (DBN) using a directed graphical model. For simplicity, one usually assumes the observations are conditionally independent, i.e.,

\[
\hat{b}_{j,d}(o_{t+1:t+d}) = P[o_{t+1:t+d} | S_{t+1:t+d} = j] = \prod_{t'=t+1}^{t+d} b_{j}(o_{t'}). \tag{12}
\]

where \( b_{j}(v_{k}) \triangleq P[o_{t} = v_{k} | S_{t} = j] \). To easily identify when a segment of states starts, one usually further assumes a state transition is independent to the previous state duration. This is just the assumption made for the explicit duration HMM and
the residential time HMM, as described on page 3. The conditional probability distribution (CPD) function for the explicit duration HMM is [126]:

\[
P[S_t = j|S_{t-1} = i, R_{t-1} = \tau] = \begin{cases} a_{ij} & \text{if } \tau = 1 \text{ (transition)}, \\ \mathbb{I}(i = j) & \text{if } \tau > 1 \text{ (decrement)}, \end{cases}
\]

\[
P[R_t = \tau'|S_t = j, R_{t-1} = \tau] = \begin{cases} p_{j}(\tau') & \text{if } \tau = 1 \text{ (transition)}, \\ 1(\tau' = \tau - 1) & \text{if } \tau > 1 \text{ (decrement)}, \end{cases}
\]

and for the residential time HMM

\[
P[Q_t = (j, \tau')|Q_{t-1} = (i, \tau)] = \begin{cases} a_{(i,j),\tau'} & \text{if } \tau = 1 \text{ (transition)}, \\ \mathbb{I}(\tau' = \tau - 1) & \text{if } \tau > 1 \text{ (decrement)} \end{cases}
\]

where \( R_t \) is the remaining duration of state \( S_t \), \( Q_t = (j, \tau) \) represents \( S_t = j \) and \( R_t = \tau \), and the self-transition probability \( a_{(i,i),\tau} = 0 \) as defined in the beginning of this section. The indicator function \( \mathbb{I}(x) = 1 \) if \( x \) is true and zero otherwise. Several DBNs for HSMMs are presented in Murphy [126].

As discussed in Gähramani [66], a Bayesian approach to learning treats all unknown quantities as random variables. These unknown quantities comprise the number of states, the parameters, and the hidden states. By integrating over both the parameters and the hidden states, the unknown quantities can be estimated. For the explicit duration HMM, the number of states, \( M \), and the maximum length of state durations, \( D \), can be determined after \( S_t \) and \( R_t \), for \( t = 1 \) to \( T \), are estimated. For the residential time HMM, after the set of hidden states that \( Q_t \) can take and the transition probabilities are estimated, the values of \( M \) and \( D \) can be determined by checking the transition probabilities of \( P[q_t|q_{t-1}] \), where \( q_t \) is the estimated hidden state of \( Q_t \). Obviously from the CPDs, \( P[q_t|q_{t-1}] = 1 \) represents a self-transition, and \( P[q_t|q_{t-1}] < 1 \) a state transition. Therefore, by counting the number of consecutive self-transitions we can determine the maximum duration of states, \( D \), and then determine the number of HSMM states, \( M \).

Sometimes, one uses a simple method to find out the order of an HSMM by trying various values of \( M \) and \( D \). Denote \( \lambda^{(M,D)} \) as the model parameter with assumed order \( M \) and \( D \). The maximum likelihood estimate of \( \lambda^{(M,D)} \) is

\[
\hat{\lambda}^{(M,D)} = \arg \max_{\lambda^{(M,D)}} \log P[o_{1:T}|\lambda^{(M,D)}],
\]

which can be determined using the re-estimation algorithms discussed in this subsection for given \( M \) and \( D \). Then the order estimators given in Ephraim and Merhav [57] can be used in the selection of the model order. For instance, the order estimator proposed by Finesso [61] can be used as the objective function for the selection of the model order:

\[
(M, D) = \min \left\{ \arg \min_{M,D \geq 1} \left\{ -\frac{1}{T} \log P[o_{1:T}|\hat{\lambda}^{(M,D)}] + 2c_{MD} \frac{\log T}{T} \right\} \right\},
\]

where \( c_{MD} = MD(MD + K - 2) \) is a penalty term that favors simpler models over more complex models, \( T \) the total number of observations, and \( K \) the total number of values that an observation can take.

In fact, we can alternatively use an undirected graphical model to describe the HSMMs and to learn the unknown quantities, such as semi-Markov conditional random fields (semi-CRFs) introduced by Sarawagi and Cohen [161]. In this model, the assumption that the observations are conditional independent is not needed.

2.4. EM algorithm and online estimation

Using the theory associated with the well-known EM (expectation–maximization) algorithm [42], it can be proved that the re-estimation procedure for the HSMMs increases the likelihood function of the model parameters. However, these algorithms require the backward procedures and the iterative calculations, and so are not practical for online learning. A few of online algorithms for HSMM have been developed in the literature, including an adaptive EM algorithm by Ford et al. [62], an online algorithm based on recursive prediction error (RPE) techniques by Azimi et al. [9,11], and recently a recursive maximum likelihood estimation (RMLE) algorithm by Squire and Levinson [168].

2.4.1. Re-estimation vs. EM algorithm

Let \( \lambda \) represent the complete set of the model parameters to be estimated in the re-estimation procedure. The purpose is to find maximum likelihood estimates of the model parameter set \( \lambda \) such that the likelihood function \( P[o_{1:T}|\lambda] \) is maximized for given \( o_{1:T} \).

Let us consider two \textit{a posteriori probabilities} (APPs) of the state sequence variable \( s_{1:T} = s_1, \ldots, s_T \), given an instance of the observation sequence \( o_{1:T} \); one under model parameter \( \lambda \) and the other under its improved version \( \lambda' \). Denote \( L(\lambda) \triangleq P[o_{1:T}|\lambda] \) as the \textit{likelihood function} of the model parameter \( \lambda \). Following the discussion given in Ferguson [60], an auxiliary function is defined as a conditional expectation [121]

\[
Q(\lambda, \lambda') \triangleq E[\log P[s_{1:T}, o_{1:T}|\lambda']|o_{1:T}, \lambda] = \sum_{s_{1:T} \in \mathcal{X}} P[s_{1:T}, o_{1:T}|\lambda] \log P[s_{1:T}, o_{1:T}|\lambda'].
\]
Because 
\[
\frac{Q(\lambda, \lambda') - Q(\lambda, \lambda)}{L(\lambda)} \leq \log \frac{L(\lambda')}{L(\lambda)},
\]
if \(Q(\lambda, \lambda') > Q(\lambda, \lambda)\), then \(L(\lambda') > L(\lambda)\). This implies that the best choice of \(\lambda'\) (for given \(\lambda\)) is found by solving the maximization problem of \(Q(\lambda, \lambda')\). Therefore, by iterating the expectation step (E-step) and the maximization step (M-step), as in the expectation–maximization (EM) algorithm, an optimum \(\lambda'\) can be found. In fact, EM algorithm holds in a more general setting than HSMMs.

Instead of using the forward–backward algorithms for the E-step and the re-estimate formulas for the M-step, Krishnamurthy and Moore [92] use the EM algorithm to directly re-estimate the model parameters. In this case, the unknowns are the model parameters \(\lambda\) as well as the state sequence \(s_{1:T}\). When the order of the model is large, the computational amount involved in the re-estimate is huge. To reduce this computational amount, MCMC sampling is used for approximate re-estimate [46,47]. MCMC sampling is a general methodology for generating samples from a desired probability distribution function and the obtained samples are used for various types of inference. Therefore, MCMC sampling can also be used in the estimation of the state sequence and the model parameters. MCMC sampling draws samples of the unknowns from their posteriors so that the posteriors can be approximated using these samples. The prior distributions of all the unknowns are required to specify before applying the MCMC sampling methods. The emission probabilities and the durations of various states are often modelled using some parametric distributions.

### 2.4.2. Forward algorithm for online estimation

Define the maximum log-likelihood of the observation sequence \(o_{1:t}\) as [168]
\[
L_t(\lambda) \triangleq \max_{d_1, \ldots, d_n} \log P[o_{1:t}, d_1, \ldots, d_n | \lambda] = \max_{d_1, \ldots, d_n} L_t^{(0)}(\lambda)
\]
where \(d_1, \ldots, d_n\) with \(d_k \in D\), and
\[
L_t^{(0)}(\lambda) \triangleq \max_{d_1, \ldots, d_n} \log P[o_{1:t}, d_1 | n, \lambda].
\]
By maximizing \(L_t(\lambda)\) with respect to \(\lambda = [a_{ij}, b_i(v_k), p_i(d), \pi_i]\), the set of model parameters \(\hat{\lambda}\) can be re-estimated [168].

Signal modelling using HMM proposed by Azimi et al. [9–11] is a different online estimation algorithm. The state \(s_t\) of the signal at time \(t\) is assumed to take its values from the set \([e_1, \ldots, e_M]\) where \(e_i\) is an \(M \times 1\) vector with unity as the \(i\)th element and zeros elsewhere. In this case, the state at time \(t\) is appropriate to be denoted using an \(M \times 1\) vector, i.e., \(s_t \in [e_1, \ldots, e_M]\).

Define the log-likelihood of the observations up to time \(t\) given \(\lambda\) [9,11]:
\[
L_t(\lambda) \triangleq \log P[o_{1:t} | \lambda] = \sum_{\tau=1}^{t} \log P[o_{\tau} | o_{1:{\tau-1}}, \lambda] = \sum_{\tau=1}^{t} x_{\tau}(\lambda),
\]
where \(x_{\tau+1}(\lambda) \triangleq \log P[o_{\tau+1} | o_{1:t}, \lambda]\). Denote the estimate of the Hessian matrix by \(R_t \triangleq \frac{\partial^2}{\partial \lambda^2} L_t(\lambda)\), and the gradient of \(x_t(\lambda)\) with respect to \(\lambda\) by \(\psi_t \triangleq \frac{\partial}{\partial \lambda} x_t(\lambda)\), then the model parameters can be updated using \(\dot{\lambda} = \lambda + e_{t+1} \cdot R_{t+1}^{-1} \cdot \psi_{t+1}\), where \(e_{t+1}\) is a step size.

A sequential online learning of HMM state duration using quasi-Bayes (QB) estimate was presented in Chien and Huang [36,37], in which the Gaussian, Poisson, and gamma distributions were investigated to characterize the duration models.

### 2.5. Implementation

It is well known that the joint probabilities associated with observation sequence often decay exponentially as the sequence length increases. The implementation of the forward–backward algorithms by programming in a real computer would suffer a severe underflow problem. This subsection considers the issues related to the practical implementation of the algorithms.

A general heuristic method to solve the underflow problem is to re-scale the forward–backward probabilities by multiplying a large factor whenever an underflow is likely to occur [106,40]. However, the scale factors cannot guarantee the backward variables being bounded or immunizing from the underflow problem, as pointed out by Murphy [126].

If the forward–backward algorithm is implemented in the logarithmic domain, like the MAP and Viterbi algorithms used for turbo-decoding in digital communications, then the multiplications become additions and the scaling becomes unnecessary [20]. In fact, the logarithmic form of the extended Viterbi algorithm can be considered as an approximation to that of the forward–backward algorithm.

The notion of posterior probabilities is used to overcome the underflow problem involved in the recursive calculation of the joint probabilities of observations [68,202]. This is similar to the standard HMM. The HMM’s forward–backward
algorithms can automatically avoid the underflow problem by replacing the joint probabilities with conditional ones [45,6]. The refined forward–backward algorithm for the HSMM becomes robust against the underflow problem, without increasing the computational complexity.

In notion of posterior probabilities, the forward variables are redefined [202] using the predicted probabilities by

\[ \tilde{\alpha}_t(j, d) \triangleq P[S_{t-d+1:t} = j | \mathbf{o}_{1:t-d}, \lambda], \]

and the backward variables by

\[ \tilde{\beta}_t(j, d) \triangleq \frac{P[\mathbf{o}_{t-d+1:T} | S_{t-d+1:t} = j, \lambda]}{P[\mathbf{o}_{t-d+1:T} | \mathbf{o}_{1:t-d}, \lambda]}. \]

Denote

\[ \tilde{b}_{j,d}(o_{t-d+1:t}) \triangleq \frac{b_{j,d}(o_{t-d+1:t})}{P[\mathbf{o}_{t-d+1:T} | \mathbf{o}_{1:t-d}, \lambda]}.
\]

We have \( \alpha_t(j, d) = P[o_{1:t-d} | \lambda] \tilde{\alpha}_t(j, d) b_{j,d}(o_{t-d+1:t}) \) and \( \tilde{\beta}_t(j, d) P[o_{t-d+1:T} | o_{1:t-d}, \lambda] = b_{j,d}(o_{t-d+1:t}) \tilde{\beta}_t(j, d). \) Then the forward–backward formulas (4) and (5) become

\[ \tilde{\alpha}_t(j, d) = \sum_{i \in \mathcal{S} \setminus \{j\}} \sum_{d' \in \mathcal{D}} \tilde{\alpha}_{t-d}(i, d') \tilde{b}_{i,d'}(o_{t-d+1:t}) a_{i,d'}(j,d), \]

and

\[ \tilde{\beta}_t(j, d) = \tilde{b}_{j,d}(o_{t-d+1:t}) \sum_{i \in \mathcal{S} \setminus \{j\}} \sum_{d' \in \mathcal{D}} a_{(j,d'),(i,d')} \tilde{\beta}_{t+d'}(i, d'). \]

The probability \( P[o_{1:t} | \lambda] \) can be determined by

\[ P[o_{1:t} | \lambda] = \sum_{j \in \mathcal{S}} \sum_{d \in \mathcal{D}} P[S_{t-d+1:t} = j, o_{1:t} | \lambda] = \sum_{j \in \mathcal{S}} \sum_{d \in \mathcal{D}} P[o_{1:t-d} | \lambda] \tilde{\alpha}_t(j, d) b_{j,d}(o_{t-d+1:t}), \]

and \( P[o_{t-d+1:t} | o_{1:t-d}, \lambda] \) by \( P[o_{1:t} | \lambda] / P[o_{1:t-d} | \lambda]. \)

After the forward–backward variables are determined, the probabilities defined by (6) through (9) can be yielded, such as

\[ \frac{\eta_t(j, d)}{P[o_{1:t} | \lambda]} = \tilde{\alpha}_t(j, d) \tilde{\beta}_t(j, d) \]

and

\[ \frac{\xi_t(i, d'; j, d)}{P[o_{1:t} | \lambda]} = \tilde{\alpha}_t(i, d') b_{i,d'}(o_{t-d'+1:t}) a_{i,d'}(j,d) \tilde{\beta}_{t+d}(j, d). \]

If we denote \( \tilde{\alpha}_t(j, d) = \tilde{\alpha}_t(j, d) b_{j,d}(o_{t-d+1:t}) \) and \( \tilde{\beta}_{t+1}(j, d) = \tilde{\beta}_{t+d}(j, d), \) then the forward recursion (13) becomes

\[ \tilde{\alpha}'_t(j, d) = \tilde{b}_{j,d}(o_{t-d+1:t}) \sum_{i \in \mathcal{S} \setminus \{j\}} \sum_{d' \in \mathcal{D}} \tilde{\alpha}'_{t-d}(i, d') a_{i,d'}(j,d), \]

and the backward recursion (14) becomes

\[ \tilde{\beta}'_t(j, d) = \tilde{b}_{j,d}(o_{t-d+1:t}) \sum_{i \in \mathcal{S} \setminus \{j\}} \sum_{d' \in \mathcal{D}} \tilde{\beta}'_{t+d}(i, d') a_{i,d}(j,d). \]

That is, the backward recursion is symmetric to the forward one in the time reversed form. This can potentially reduce the requirement for the silicon area on a chip if the backward logic module uses the forward one. A symmetric forward–backward algorithm for the residential time model was introduced by Yu and Kobayashi [199].

3. Conventional models

As pointed out in the last section on page 3, the general HSMM is reduced to the conventional models when specific assumptions are made on the dependency of states and durations. This section overviews three conventional models, including the explicit duration HMM, variable transition HMM and residential time HMM. These models have fewer parameters and lower computational complexity than the general model. They are the HSMMs often found in the literature.
3.1. Explicit duration HMM

Ferguson [60] was the first to consider the HSMM, which he called an “HMM with variable duration.” Since then a number of studies have been reported on the subject. See for example, Mitchell and Jamieson [123], Yu and Kobayashi [199,202] and references therein.

The explicit duration hidden Markov model assumes that a state transition is independent to the duration of the previous state, i.e., \( a_{i,j}(d) = 0 \), without self-transitions, i.e., \( a_{i,i}(d) = 0 \). The state duration is assumed to be dependent on the current state and independent to the previous state. That is, state \( j \) will last for duration variable \( d \) according to the conditional probability \( p_j(d) \), as defined in (2). Therefore, we have \( a_{i,j}(d) = a_{i,j}p_j(d) \) with \( a_{i,i} = 0 \), for \( i, j \in S, d \in D \), where \( a_{i,j} \triangleq P[S_t = j | S_{t-1} = i] \) is the state transition probability from state \( i \) to state \( j \). It also assumes the “conditional independence” of outputs as defined in (12). Due to all those independent assumptions, the explicit duration HMM is one of the most simple models among all the HSMMs. Therefore, it is the most popular HSMM in applications.

Replace \( a_{i,j}(d) \) with \( a_{i,j}d(p_j(d), b_j(d)ot_{t+1}d+1) \) with \( \prod_{t=1}^{tid} b_j(o_t) \), \( \beta_t(d) \) with \( \beta_t(j) \triangleq P[ot_{t+1}d | S_t = j, ot_{1,t}d] = \sum_{d \in D} a_{t}(j,d) \). Then we readily obtain the forward–backward formulas for the explicit duration HMM [60]:

\[
\alpha_t(j) = \sum_{d \in D} \alpha_{t-d}^i(j)p_j(d)ut(j,d), \tag{15}
\]

\[
\alpha_t^i(j) \triangleq P[S_{t+1} = j, ot_{1,t}d] = \sum_{i \in S(j)} a_{t}(i)a_{ij}, \tag{16}
\]

for \( j \in S, t = 1, \ldots, T \), and

\[
\beta_t^i(j) \triangleq P[ot_{t+1}d | S_{t+1} = j, ot_{1,t}d] = \sum_{d \in D} p_j(d)ut_{t+1}(j,d)\beta_{t+1}(j). \tag{17}
\]

\[
\beta_t(j) = \sum_{i \in S(j)} a_{ji}^{*} \beta_t^i(i), \tag{18}
\]

for \( j \in S, t = T - 1, \ldots, 0 \), where

\[
ut(j,d) \triangleq \prod_{t=1}^{t} b_j(o_t). \tag{19}
\]

The forward variable \( \alpha_t(j) \) represents the joint probability that state \( j \) ends at \( t \) and the partial observation sequence is \( ot_{1,t}d \), and \( \alpha_t^i(j) \) the joint probability that state \( j \) starts at \( t + 1 \) and the partial observation sequence is \( ot_{1,t}d \). The backward variable \( \beta_t(j) \) represents the conditional probability that given state \( i \) ending at \( t \), the future observation sequence is \( ot_{t+1}d \). and \( \beta_t^i(j) \) the conditional probability that given state \( j \) starting at \( t + 1 \) the future observation sequence is \( ot_{t+1}d \).

The boundary conditions use the simplifying assumption described on page 5, i.e., \( \alpha_0(i) = \pi_i \) and \( \alpha_t(i) = 0 \) for \( t < 0 \), and \( \beta_T(i) = 1 \) and \( \beta_T(i) = 0 \) for all \( t > T \), \( i \in S \), where \( \pi_i \) is the initial distribution of state \( i \).

3.1.1. Computational complexity

From (19), we can see that \( ut(j,d) \) for \( j \in S, d \in D \), given \( t \) require \( O(MD^2/2) \) multiplications, and the forward–backward formulas (16), (15), (18) and (17) require extra \( O(M^2 + MD) \) multiplications. Therefore, the computational complexity of the explicit duration HMM is \( O((M^2 + MD + MD^2)T) \). The storage requirement is \( O(M^2 + MD + MK + MT) \), where \( K \) is the total number of observable values that \( O_t \) can take. Due to the computational complexity is high, the explicit duration HMM is not appropriate to be applied in some applications when \( D \) is large. To reduce this computational complexity, the key is to reduce the computational complexity of \( ut(j,d) \). Levinson [107] suggested a recursive method that can calculate the product more efficiently, i.e.,

\[
ut(j,d) = \prod_{t=1}^{t} b_j(o_t) = ut(j,d - 1) \cdot b_j(o_{t-d+1}) \tag{20}
\]

with \( ut(j,1) = b_j(o_1) \), which requires \( O(MD) \) multiplications. This recursive method was also used by Mitchell et al. [124]. However, in their method \( D \) recursive steps must be performed at every \( t \). Therefore, the total number of recursive steps required in their method increases by a factor of \( D \) compared with the Ferguson algorithm [60]. In fact, a better way to reduce both the computational complexity and the total number of recursive steps is letting

\[
ut(j,d) = ut_{t-1}(j,d - 1) \cdot b_j(o_t) \tag{21}
\]

which can be implemented in parallel manner and has no need to retrieve previous observation probabilities \( b_j(o_{t-d+1}) \). This idea was realized in a parallel implementation of the explicit duration HMM for spoken language recognition on a
3.2. Variable transition HMM

In this model, an HSMM is realized in the HMM framework, including the 2-vector HMM [93], the duration-dependent state transition model [179,181,180], the inhomogeneous HMM [151], and the non-stationary HMM [164,46,47]. These approaches take the vector \((\mathbf{i}, \mathbf{d})\) as an HMM state, where \(\mathbf{i}\) is one of the HSMM states and \(\mathbf{d}\) sojourn time since entering the state. The explicit duration HMM can also be expressed in this model by letting the triples \((i, w, d)\) to be HMM states, where \(d\) is a duration and \(w\) a counter, \(1 \leq w \leq d\), which indicates the number of observations produced so far while in state \(i\) [60]. In addition to the state and its sojourn time, Pieczynski et al. [144] added the observation as the third component. This makes it possible to generalize the model to the triplet Markov chain [144–147,98–101,2]. The constraints among the three components are released in the triplet Markov chain model and the components are extended to be general processes. The price is the loss of physical meaning in the sense of hidden semi-Markov process. One has to add some constraints back on the triplet Markov chain and re-define the meaning of the three processes when it is applied for the HSM. The triplet Markov chain model can be further generalized to be a non-stationary fuzzy Markov chain by letting the underlying Markov chain be a fuzzy Markov random chain [158].

Compared with the explicit duration HMM, this model assumes the state transition is dependent on the state duration, and hence it is more suitable for describing inhomogeneous or non-stationary hidden Markov processes. This makes it useful for some applications that cannot be modelled by a homogeneous process.

A state transition is allowed only for either \((i, d) \rightarrow (j, 1)\), for \(i \neq j\), or \((i, d) \rightarrow (i, d+1)\) for self-transitions. It assumes the “conditional independence” of outputs as given by (12). The boundary conditions use the simplifying assumption described on page 5. The state transition probability from state \(i\) to state \(j\) given that the sojourn time in state \(i\) at time \(t\) is \(d\) is defined by [151,93]

\[
a_{ij}(d) \triangleq P[S_{t+1} = j | S_{t-d+1:t} = i],
\]

subject to \(\sum_{j \in S, d \in D} a_{ij}(d) = 1\), for \(i \in S, d \in D\), where the self-transition with probability \(a_{ii}(d) > 0\) can occur. We note that \(a_{ij}(d)\) is different from \(a_{i,d}j\) defined by (1). The latter does not allow self-transition. We have \(a_{ij}(d) = (1 - a_{ii}(d))a_{ii}(d)\).

It is not straightforward to derive the forward–backward formulas for the variable transition HMM from the general ones given by (4) and (5) because \(a_{i,d}(i,d') = a_{ij}(d) \prod_{t=1}^{d-1} a_{jj}(t)[1 - a_{jj}(d')]/[1 - a_{ii}(d)]\). Instead, they are easy to derive from the definitions of the forward variables \(\alpha_t(j, d)\) and the backward variables \(\beta_t(j, d)\) [151,93], i.e.,

\[
\alpha_t(j, d) \triangleq P[S_{t-d+1:t} = j, o_{1:t} | \lambda] = \begin{cases} \sum_{d' \in D} \sum_{i \notin S_t} \alpha_{t-1}(i, d') a_{ij}(d') b_j(\alpha_t), & d = 1, \\ \alpha_{t-1}(j, d-1) a_{jj}(d-1) b_j(\alpha_t), & d > 1 \end{cases}
\]

for \(j \in S, d \in D, t = 2, \ldots, T\), and

\[
\beta_t(j, d) \triangleq P[o_{t+1:T} | S_{t-d+1:t} = j, \lambda] = \sum_{i \in S_t} a_{ji}(d) \beta_{t+1}(i, 1) b_i(\alpha_{t+1}) + a_{jj}(d) \beta_{t+1}(j, d+1) b_j(\alpha_{t+1})
\]

for \(j \in S, d \in D, t = T - 1, \ldots, 1\). The forward variable \(\hat{\alpha}_t(j, d)\) represents the joint probability that the sojourn time in state \(j\) at time \(t\) is \(d\) and the partial observation sequence is \(o_{1:t}\). The backward variable \(\hat{\beta}_t(j, d)\) is the conditional probability that the future observation sequence is \(o_{t+1:T}\) given that the sojourn time in state \(j\) at time \(t\) is \(d\).

The boundary conditions are \(\hat{\alpha}_1(j, 1) = \pi_j b_j(\alpha_1)\), \(\hat{\alpha}_1(j, d) = 0\) for \(d > 1\) and \(\hat{\beta}_T(j, d) = 1\), for \(j \in S, d \in D\). Similar to the forward recursion formula, a Viterbi algorithm for the inhomogeneous HMM can be readily obtained by replacing the sum \(\sum_{d' \in D} \sum_{i \notin S_t} \alpha_{t-1}(i, d') a_{ij}(d') b_j(\alpha_t)\) of (22) with the maximum operations \(\max_{d' \in D} \max_{i \notin S_t} \alpha_{t-1}(i, d')\).

Though the super state space of the pairwise process \((i, d)\) is \(S \times D\) in the order of \(MD\), the computational complexity is \(O((MD + M^2D)T)\), where \(M\) is the number of HSMM states, \(D\) the maximum duration of any HSMM state and \(K\) the total number of observable values. The storage requirement is \(O(M^2D + MK + MDT)\). Compared with \(O((M^2D + MD + MD^2)T)\) of the explicit duration HMM, the computational complexity of the variable transition HMM is higher when the order of the state space is higher, and is lower when the maximum length of the state space is smaller. However, its space complexity is definitely higher than \(O(M^2D + MK + MDT)\) of the explicit duration HMM.

The variable transition HMM and the explicit duration HMM have different assumptions for their models. However, the model parameters of the variable transition HMM can be expressed by those of the explicit duration HMM [47,11]. Reversely, the model parameters of the explicit duration HMM can be expressed by those of the variable transition HMM, i.e., \(\pi_i(d) = \prod_{t=1}^{d-1} a_{ii}(t) [1 - a_{ii}(d)]\) and \(a_{ij} = a_{ij}(d)/[1 - a_{ii}(d)]\). The model [119] defined by (1) and (2) can be considered as a combination of the variable transition HMM and the explicit duration HMM. Therefore, it is more general.
3.3. Residential time HMM

The residential time HMM [199] assumes a state transition is either \((i, 1) \rightarrow (j, \tau)\) for \(i \neq j\) or \((i, \tau) \rightarrow (i, \tau - 1)\) for a self-transition with \(\tau > 1\), where \(\tau\) is the residential time of state \(i\). The state transition probabilities are assumed to be independent to the duration of the previous state. The residential time HMM also assumes the "conditional independence" of outputs as yielded by (12). The boundary conditions use the simplifying assumption described on page 5. Therefore, this model is useful in the application areas when the residential time that the current state will stay in the future is of interest. This is useful for predicting the residential time of the current state. It is contrast to the variable transition HMM which concerns the sojourn time that the current state has been stayed in the past.

As defined by (3), the state transition probability from state \(i\) to state \(j\) that will have residential time \(\tau\) is \(a_{i,j,\tau} \triangleq P[S_{t+\tau-1} = j|S_{t-1} = i]\) for \(i \neq j\), with \(\sum_{j} P(S_{t+\tau-1} = j|S_{t-1} = i) = 1\). The self-transition probability from \((i, \tau)\) to \((i, \tau - 1)\) is \(P[S_{t+\tau+1} = i|S_{t+\tau-1} = i] = 1\), for \(\tau > 1\). Therefore, the general forward–backward formulas given by (4) and (5) can be reduced to the ones for the residential time HMM. Instead, we derive the formulas directly from the definitions of the forward and backward variables. Define the forward variable and backward variable by [199]

\[
\tilde{\alpha}_t(i, \tau) \triangleq P[S_{t:t+\tau-1} = i, o_{1:t}|\lambda] \\
\text{and} \\
\tilde{\beta}_t(i, \tau) \triangleq P[o_{t+1:T}|S_{t:t+\tau-1} = i, \lambda].
\]

The forward variable \(\tilde{\alpha}_t(i, \tau)\) is the joint probability that the partial observation sequence is \(o_{1:t}\) and the current state \(i\) will stay for the next \(\tau\) steps and end at \(t + \tau - 1\). The backward variable \(\tilde{\beta}_t(i, \tau)\) is the conditional probability that the future observations will be \(o_{t+1:T}\) given the current state \(i\) that has \(\tau\) steps of remaining time.

The following forward and backward recursion formulas can be readily obtained:

\[
\tilde{\alpha}_t(i, \tau) = \tilde{\alpha}_t-1(i, \tau + 1)b_1(o_t) + \sum_{j \in \mathcal{S} \setminus \{i\}} \tilde{\alpha}_t-1(j, 1)a_{j,i,\tau}b_1(o_t),
\]

(24)

for \(i \in \mathcal{S}, \tau \in \mathcal{D}, t = 1, \ldots, T\), and

\[
\tilde{\beta}_t(i, \tau) = b_j(o_{t+1})\tilde{\beta}_t+1(i, \tau - 1), \quad \tau > 1,
\]

(25)

\[
\tilde{\beta}_t(i, 1) = \sum_{j \in \mathcal{S}} \sum_{\tau \geq 1} a_{i,j,\tau}b_j(o_{t+1})\tilde{\beta}_t+1(j, \tau),
\]

(26)

for \(i \in \mathcal{S}, \tau \in \mathcal{D}, t = T - 1, \ldots, 1\). The boundary conditions are \(\tilde{\alpha}_0(i, 1) = \pi_i\), \(\tilde{\alpha}_0(i, \tau) = 0 \quad \text{for} \quad \tau > 1\), and \(\tilde{\beta}_T(i, 1) = 1\), \(\tilde{\beta}_T(i, \tau) = 0 \quad \text{for} \quad \tau > 1\).

3.3.1. Computational complexity

The computational complexity involved in the residential time HMM is in the same order of the variable transition HMM. However, it can be reduced significantly if the state duration is assumed to be independent to the previous state. In this case, we have \(a_{i,j,\tau} = a_{ij}p_j(\tau)\). From the definition of \(\alpha^*_t(i)\) given by (16) and the definition of \(\beta^*_t(j)\) given by (17), we have

\[
\alpha^*_{t-1}(i) = P[S_t = i, o_{1:t-1}|\lambda] = \sum_{j \in \mathcal{S} \setminus \{i\}} \tilde{\alpha}_{t-1}(j, 1)a_{ji}
\]

(27)

and

\[
\beta^*_t(j) = P[o_{t+1:T}|S_{t+\tau-1} = j] = b_j(o_{t+1})\sum_{\tau \in \mathcal{D}} p_j(\tau)\tilde{\beta}_{t+1}(j, \tau).
\]

(28)

Then the forward formula (24) and the backward formula (26) are reduced to [199]

\[
\tilde{\alpha}_t(i, \tau) = \tilde{\alpha}_t-1(i, \tau + 1)b_1(o_t) + \alpha^*_{t-1}(i)p_1(\tau)b_1(o_t)
\]

(29)

and

\[
\tilde{\beta}_t(i, 1) = \sum_{j \in \mathcal{S} \setminus \{i\}} a_{ij}\beta^*_t(j).
\]

(30)

Now the forward recursion is given by (27) and (29), and the backward recursion by (28), (30) and (25).

In this case, computing the forward variables \(\alpha^*_{t-1}(i)\) for all \(i\) requires \(O(M^2)\) steps, and \(\tilde{\alpha}_t(i, \tau)\) for all \(i\) and \(\tau\) requires extra \(O(\lambda)\) steps. Similarly, computing the backward variables \(\beta^*_t(j)\) for all \(j\) requires \(O(M\lambda)\) steps, and \(\tilde{\beta}_t(i, 1)\) for
all i requires extra $O(M^2)$ steps. Hence, the total number of computation steps for evaluating the forward and backward variables is $O((MD + M^2)T)$. This computational complexity is much lower than those of the explicit duration HMM and the variable transition HMM.

Because the backward variables $\hat{p}_i(t, \tau)$ and the probabilities $\eta_i(t, \tau), \xi_i(i, j)$ and $\gamma_i(i)$ do not have to be stored for estimate of the model parameters $\{p_j(\tau), a_{ij}, b_i(\eta_0), \pi_i\}$, and only the forward variables $\hat{a}_i(i, 1)$ and $\alpha_{i-1}^*(i)$ for all i and t need to be stored, with the storage requirement of $O(\tau)$. Therefore, the storage requirement for the residential time HMM is $O(M^2 + MD + MK + MT)$, similar to the explicit duration HMM.

The Matlab code for the forward–backward algorithm is quite simple which can be found from the website http://sist.sysu.edu.cn/~syu/. An R package for analyzing hidden semi-Markov models can be found in Bulla et al. [23].

4. Specific modelling issues

This section discusses the issues related to the duration distributions, the observation distributions, variants of the HSMMs and the relationship with standard HMMs.

4.1. Different duration distributions

The choice of distribution family for the state duration is central to the use of the HSMM [56]. So far, we have discussed the general distribution of state duration, where $p_i(d)$ is non-parametric. In some applications, a parametric distribution may be preferred so that only a few parameters that specify the selected distribution functions are required to be estimated. This subsection presents exponential family distributions, convex monotonic distributions, and discrete Coxian distributions of duration. A uniform distribution of duration can be found in Hongeng and Nevatia [79]. More complex duration models can be found in Ostendorf et al. [136], and a discussion on the capacity and complexity of duration modelling techniques can be found in Johnson [86]. Besides, the state duration distributions can be taken into account in the Viterbi algorithm as in Burshtein [25,26], Yoma and McInnes [195] and Yoma and Sanchez [196], where the state duration distributions can be estimated efficiently for a left–right HMM.

4.1.1. Exponential family distribution of duration

The state duration can be modelled by Poisson [155], Gaussian [5] and gamma distributions [107]. All these distributions belong to the exponential family [107,123]. The probability density function (pdf) or probability mass function (pmf) for the duration of state $j$ belonging to the exponential family can be expressed as [123]

$$p_j(d) = \frac{1}{B(\theta_j)} \xi(d) \exp\left(-\sum_{p=1}^{P} \theta_{j,p} S_p(d)\right)$$

where $P$ is the number of natural parameters, $\theta_{j,p}$ is the $p$th natural parameter for state $j$ and $\theta_j = [\theta_{j,1}, \ldots, \theta_{j,P}]$. $S_p(d)$ and $\xi(d)$ are sufficient statistic, and $B(\theta_j)$ is a normalizing term. Since HSMMs are discrete-time stochastic processes, a pmf is used rather than a pdf. For simplicity, the pmf is obtained by letting

$$B(\theta_j) = \sum_{d=1}^{D} \xi(d) \exp\left(-\sum_{p=1}^{P} \theta_{j,p} S_p(d)\right).$$

According to Ferguson [60], the new duration parameters for state $j$ can be found by maximizing

$$f(\theta_j) \triangleq \sum_{d=1}^{D} \hat{p}_j(d) \log p_j(d)$$

subject to the constraint $\sum_{d=1}^{D} p_j(d) = 1$, where $\{\hat{p}_j(d)\}$ is the non-parametric probability mass function estimated by the re-estimation formulas, and $\sum_{d=1}^{D} \hat{p}_j(d) = 1$. Since the exponential family is log-concave, the global maximum can be found by setting the derivative equal to zero, yielding the maximum likelihood equations [123]

$$\frac{\partial}{\partial \theta_{j,p}} f(\theta_j) = \sum_{d=1}^{D} \hat{p}_j(d) \left[-\frac{\partial \log B(\theta_j)}{\partial \theta_{j,p}} - S_p(d)\right] = 0$$

where, for the pmf,

$$-\frac{\partial \log B(\theta_j)}{\partial \theta_{j,p}} = \sum_{d=1}^{D} p_j(d) S_p(d).$$

Therefore, the new duration parameters can be found by solving the following equations:
The MAP estimation of the mean and variance of Gaussian and gamma distributions of state durations for a left–right HMM can be found in Yoma and Sanchez [196].

4.1.2. Convex monotonic distributions
For the explicit duration HMM, the extended Viterbi algorithm for HSMM given by (11) becomes
\[
\delta_t(j) = \max_{d \in D} \max_{i \in S(t)} \left\{ \delta_{t-d}(i) a_{ij} p_j(d) \prod_{t' = t-d+1}^t b_j(o_{t'}) \right\},
\]
(31)
for \(1 \leq t \leq T\), \(j \in S\), where \(\delta_t(j) \triangleq \max_{s_{t-1},...} P[s_{t-1}, S_1 = j, o_{t-1},...,o_t]\). Tweed et al. [178] find that if \(p_j(d)\) is concave monotonic, i.e.,
\[
C_1 p_j(d_1) \leq C_2 p_j(d_2) \implies C_1 p_j(d_1 + \tau) \leq C_2 p_j(d_2 + \tau),
\]
for \(\tau > 0\) and arbitrary constants \(C_1\) and \(C_2\), then
\[
\delta_{t-d}(i_1) a_{i_1 j} p_j(d_1) \prod_{t' = t-d+1}^t b_j(o_{t'}) \leq \delta_{t-d}(i_2) a_{i_2 j} p_j(d_2) \prod_{t' = t-d+1}^t b_j(o_{t'})
\]
\[
\implies \delta_{t-d}(i_1) a_{i_1 j} p_j(d_1 + \tau) \prod_{t' = t-d+1}^t b_j(o_{t'}) \leq \delta_{t-d}(i_2) a_{i_2 j} p_j(d_2 + \tau) \prod_{t' = t-d+1}^t b_j(o_{t'})
\]
(32)
for \(d_1 > d_2\) and any \(\tau > 0\). This means that for given state \(j\) if the longer segmentation has a lower probability, then it will always have a lower probability as both segmentations are further extended [178]. This fact can be used to reduce the number of items in (31). That is, if (32) is satisfied, then \(\delta_{t-d}(i_1)\) can never give optimal solutions in the future and the index \((t - d_1, i_1)\) can be removed from the \(j\)th set, \(Q(j)\), for state \(j\). Therefore, there exists \(t_n^* = \min_{t-d \in Q(j)} (t - d_1)\) such that \((t_0, i_1) \notin Q(j)\), for any \(t_0 < t_n^*, i_1 \in S\). This implies that \(Q(j)\) is a queue that only older ones can be discarded. The newer items \((t, i), i \in S\), are pushed into the queue after \(\delta_t(j)\) for all \(j \in S\) are determined by
\[
\delta_t(j) = \max_{(t_0, i) \in Q(j)} \left\{ \delta_{t_0}(i) a_{i j} p_j(t - t_0) \prod_{t' = t_0+1}^t b_j(o_{t'}) \right\},
\]
for \(1 \leq t \leq T\), \(j \in S\), where for simplicity it assumes \(D \geq T\).

4.1.3. Discrete Coxian distribution of duration
A discrete Coxian distribution of duration is an extension of the non-parametric distribution of duration for a conventional HMM that the duration distribution is \(p_j(d)_t\), for given state \(i \in S\).

Denote the discrete Coxian distribution by \(\text{Cox}(\mu, \theta)\), where \(\mu = \mu_1, ..., \mu_N\) and \(\theta = \theta_1, ..., \theta_N\) are parameters [54, 55]. A left-to-right Markov chain with \(N\) states (phases) is used to describe \(\text{Cox}(\mu, \theta)\). Each phase \(n\) has the self-transition probability \(A_{nn} = 1 - \theta_n\), \(0 < \theta_n < 1\), \(n \in \{1, ..., N\}\). The duration distribution of geometric distribution \(X_n \sim \text{Geom}(\theta_n)\). If the left-to-right Markov chain starts from phase \(n\) with initial probability \(\mu_n\), then \(X_n + \cdots + X_N\) is the duration of the left-to-right Markov chain, for \(0 \leq \mu_n \leq 1, \sum \mu_n = 1\).

In this state, the state duration distribution, \(p_j(d)_t\), of the HSMM becomes \(\text{Cox}(\mu(j), \theta(j))\) with \(d = X_n + \cdots + X_N\), for \(i \in S\), where \(\mu(j)\) and \(\theta(j)\) are the Coxian parameter set for the HSMM state \(i\). When \(\theta_1 = 1\) for all \(n\), it reduces to the conventional Coxian distribution of HSMM with \(X_n \sim \text{Geom}(\theta)\). Therefore, the traditional forward-backward algorithm for HMM can be applied for the model parameter re-estimation and state sequence estimation.

An HMM state \((i, n)\) can transit to \((i, n + 1)\) with self-transition probability \(A_{nn} = 1 - \theta_n\) for any \(n, (i, n + 1)\) with probability \(A_{nn} = \theta_n\) for \(n < N\), or \((j, n')\) with probability \(a_{ij} \mu(n)\) for \(n = N\), where \(A_{nn} = \theta_n\) are parameters for state \(i\). The computational complexity is \(O(M^2 NT)\).

A modification of this model is assuming that the left-to-right Markov chain of each HSMM state always starts from phase 1 and ends at any phase \(n\). Then the forward algorithm can be expressed as
\[
\alpha_t[(i, n), d] = \begin{cases} \sum_{j \in S} \sum_{n'=1}^N \alpha_{t-1}[(j, n'), d'] p_j(d') a_{ji} b_{i,1}(o_t), & n = 1, d = 1, \\ \alpha_{t-1}[(i, 1), d-1] A_n^{(i)} b_{i,1}(o_t), & n = 1, d > 1, \\ 0, & n > 1, d = 1, \\ \sum_{n'=n-1}^n \alpha_{t-1}[(i, n'), d-1] A_{n'}^{(i)} b_{i,n}(o_t), & n > 1, d > 1 \end{cases}
\]
where \( p_i(d) \) is the duration probability, \( b_{i,n}(o_t) \) the observation probability, \( A_{n,n}^{(i)} \) the transition probability from phase \((i, n')\) to \((i, n)\), and \( o_t(i, n, d) \) the forward variable at time \( t \) when the state is \( i \), the phase is \( n \), and the duration of state \( i \) is \( d \) \([182,183,166]\). The Viterbi version of this forward formula can be straightforwardly replaced by a probability \( \frac{\exp(-\mu_i)}{\mu_i!} \) of the Gaussian distribution. This model was called “exponentially decay state” by Krishnamurthy and Moore \[92\]. An application in detection of shape anomalies can be found in Z. Liu et al. \[114\].

Another modification of the model is allowing the Markov chain of each HSMM state is not necessary to be left-to-right. That is, the transition probability from phase \( n \) to any phase \( m \) is also allowed, with \( A_{nm} \geq 0 \), for \( n \neq m \) and \( n, m \in \{1, \ldots, N\} \). This is called “extended HMM” in Russell and Cook \[156\]. In this case, the Markov chain of each HSMM state is in fact a series-parallel network of geometric processes. Based on Coxian theory, the overall duration pdf of the series-parallel network can construct any discrete pdf with rational \( z \)-transform \([182,183]\).

### 4.2. Different observation distributions

This subsection presents various observation distributions that have been used in applications.

#### 4.2.1. Parametric distribution of observations

The observation variable \( O_t \) is usually assumed as a discrete variable with finite alphabet \( |V| = K \). In some applications, however, a parametric distribution with possibly infinite support may be required or preferred. For instance, the probability distribution \( b_{i,j}(v_k) \) may be represented as a Poisson distribution. In some other applications, the observation variable \( O_t \) may be treated as a continuous variable, e.g., Gaussian random variable. In such cases the number of model parameters can be reduced substantially \([60]\).

For example, if the probability density function \( b_i(v_k) \), for \( v_k = 0, 1, \ldots, \infty \), is Poisson with mean \( \mu_i \), i.e., \( b_i(k) = \mu_i^k e^{-\mu_i}/k! \), then the parameter \( \mu_i \) can be re-estimated by \( \hat{\mu}_i = \sum_{k} b_i(k)k \), or equivalently,

\[
\hat{\mu}_i = \frac{\sum_{t=1}^{T} Y_t(i) o_t}{\sum_i \sum_{t=1}^{T} Y_t(i)}.
\]

A similar result can be obtained by directly maximizing the likelihood function \( P[o_{1:T} | \lambda] \) \([108]\).

For the continuous random variable \( O_t = v_k \), the probability distribution \( b_j(v_k) \) should then be replaced by a probability density function. More generally, the observation distribution \( b_j(v_k) \) of any state \( j \) is often represented by a mixture of distributions such as Gaussian distributions \([137]\). For a given state \( j \), the probability density function that state \( j \) produces an observation \( v_k \) can then be written as \([83]\)

\[
b_j(v_k) = \sum_{n=1}^{N} p_{jn} f_{jn}(v_k)
\]

where \( N \) denotes the number of the mixture probability density functions, \( \{f_{jn}(v_k)\} \), and \( \sum_{n=1}^{N} p_{jn} = 1 \). A linear mixed models for the observation distribution can be found in Chaubert-Pereira et al. \[30\].

#### 4.2.2. Segmental model

Usually the observation distributions are assumed to be dependent on the states. The segmental model \([92,157,63,87,43,74,136,138,78,203,64,1]\) extend them to be dependent on the states as well as the state durations.

Suppose there is a sequence of distribution regions corresponds to the sequence of observations for a given state. Then the observation probabilities are defined by \( b_{i,d}(o_t) \triangleq P[o_t | S_{t-d+d} = i] \), where \( b_{i,d}(\cdot) \) denotes the \( d \)th distribution region of given state \( i \), for \( d = 1, \ldots, D \). \( \{b_{i,d}(\cdot)\} \) can be a set of model regions or a continuum of distributions via trajectory sampling. An instance of the continuum of distributions is

\[
b_{i,d}(o_t) = \frac{1}{\sqrt{2\pi} \sigma_i} \exp\left(-\frac{(o_t - c_i d^{-1} \mu_i)^2}{2\sigma_i^2}\right),
\]

where \( c_i, \mu_i, \sigma_i \) are the parameters of the Gaussian distribution. This model was called “exponentially decay state” by Krishnamurthy and Moore \[92\]. An application in detection of shape anomalies can be found in Z. Liu et al. \[114\].

More generally, the observation probabilities can be defined by \( b_{i,d}^{(\tau)}(o_{t+d}) \triangleq P[o_{t+d} | S_{t+1:t+d+\tau} = i] \), for \( d = 1, \ldots, \tau \), where \( \tau \) is given and denotes the length of the segment or the duration of state \( i \), \( o_{t+d} \) is the \( d \)th observation of the segment \( o_{t+1:t+d+\tau} \), and \( b_{i,d}^{(\tau)}(\cdot) \) the \( d \)th distribution region for given state \( i \) and the length \( \tau \). According to this definition, \( b_{i,d}^{(\tau)}(o_{t+d}) \) may differ from \( b_{i,d}^{(\tau_2)}(o_{t+d}) \) if \( \tau_1 \neq \tau_2 \). For instance, Kim and Smyth \[89\] use the segmental HMM in waveform modelling, which models the \( r \)th segment of observations of length \( \tau \), \( o_r = o_{t+1:t+r} \), generated by state \( i \), as a linear function of time

\[
o_r = a_r + c_r x_r + e_r, \quad e_r \sim N_r(0, \sigma_r^2 I_r)
\]

where \( a_r \) and \( c_r \) are regression coefficients for the intercept and slope of the waveform, respectively. \( x_r \) is a \( \tau \times 1 \) vector representing discrete time values. \( e_r \) a \( \tau \times 1 \) vector of Gaussian noise with variance \( \sigma_r^2 \) for each component. If random effects
are added to the segment distribution to model parameter variability across waveforms, then the regression coefficients can be \( a_i \sim N(\hat{\alpha}_i, \sigma_i^2) \) and \( c_i \sim N(\hat{\gamma}_i, \sigma_i^2) \) [89].

Generally, extra-variability associated with a state \( i \) can be characterized by a probability density function \( g_i \) called the state target PDF [157,77]. A target distribution \( b_i(\cdot) \) is chosen according to \( g_i \) [7]. Then the joint probability of the segment \( o_{t+1:t+r} \) and a particular target \( b_i(\cdot) \) given state \( i \) is given by [157]:

\[
P[o_{t+1:t+r}, b_i | S_{[t+1:t+r]} = i] = g_i(b_i) \prod_{d=1}^{r} b_i(o_{t+d}).
\]

Therefore, \( P[o_{t+1:t+r} | S_{[t+1:t+r]} = i] = \sum_{b_i} P[o_{t+1:t+r}, b_i | S_{[t+1:t+r]} = i] \).

If the distribution regions \( [b_{i,d}(\cdot)] \) are given, then it is a deterministic distribution mappings that associate the \( d \)th observation \( o_{t+d} \), for \( d = 1, \ldots, \tau \), in the \( \tau \)-length segment \( o_{t+1:t+r} \) with the \( d \)th specific region \( b_{i,d}(\cdot) \); otherwise, it is a dynamic distribution mappings, which can be implemented using dynamic programming to find the maximum likelihood mapping that associate the segment \( o_{t+1:t+r} \) to a fixed number of regions. If we still assume that observations are conditionally independent given the segment length, i.e.,

\[
P[o_{t+1:t+r} | S_{[t+1:t+r]} = i] = \prod_{d=1}^{\tau} b_{i,d}(o_{t+d})
\]

then a segment model with an unconstrained dynamic mapping is equivalent to an HMM network [103]. A detailed discussion of segmental models can be found in Ostendorf et al. [136].

An extension of the segmental model is defining the observation distributions to be dependent on both the state and its substates [97]. This model is in fact a special case of the original HSMM if we define a complex state that includes both the state and its substate.

4.2.3. Event sequence model

Thoraval et al. [175], Thoraval [176] and Faisan et al. [58,59] introduced a hidden semi-Markov event sequence model (HSMESM), which is a special instance of hidden semi-Markov model for the modelling and analysis of event-based random processes. At a given time \( t \) there may be an event \( o_t \in V_k \) observed, with an occurring probability \( 1 - e_{s_t} \), or a null observation (missing observation) \( o_t = \phi \) with missing probability \( e_{s_t} \), where \( s_t \) is the state at time \( t \). Therefore, the modified observation probabilities in the case that there exist missing observations, denoted by \( b_{i,t}^+(o_t) \), are given by

\[
b_{i,t}^+(o_t) = (1 - e_t) b_i(o_t) \cdot \mathbb{I}(o_t \in V) + e_t \cdot \mathbb{I}(o_t \notin V), \quad i \in S,
\]

where \( V \) is the set of observable events, and \( \mathbb{I}(x) = 1 \) if \( x \) is true and zero otherwise.

This event sequence model is called “state-dependent observation misses” by Yu and Kobayashi [200] because the null observation \( \phi \) is treated as one of the observable, i.e., the full set of observations is \( \phi \cup V \). They classify patterns of observation misses into five types, where except the state-dependent observation misses, the other four types are as follows:

a) Output-dependent observation misses: the probability that a given \( o_t \) becomes “null” depends on the output value \( o_t \) itself. For instance, when the output is too weak (in comparison with noise) at time \( t \), such output may not be observed. In this case, the “output-dependent miss probability” is defined by \( e(v_k) = P[\phi | v_k] \). Then the probability for a null observation \( o_t = \phi \) is \( \phi = \sum_i b_i(v_k)e(v_k) \).

b) Regular observation misses: the outputs \( \{o_t\} \) are observed only at predetermined epochs. Regular or periodic sampling is a typical example. The rest of \( \{o_t\} \) will be missed, and such portion will be considerable, if the sampling is done infrequently. In this case, the observation probability for a null observation \( o_t = \phi \) is \( \phi = \sum b_i(v_k) \).

c) Random observation misses: the outputs \( \{o_t\} \) are observed at randomly chosen instants. Such observation pattern may apply, when the measurement is costly or we are not interested in keeping track of state transitions so closely. In this case, some outputs may become “null” observations randomly. If the sampling probability is \( 1 - e \), then the modified observation probability is \( b_{i,t}^+(o_t) = \mathbb{I}(o_t \in V)(1 - e) b_i(o_t) + \mathbb{I}(o_t \notin V)e \).

d) Miss match between multiple observation sequences: multiple observation sequences are associated with the hidden state sequence, and these observations may not be synchronized to each other. For instance, two sequences \( \{o_t\} \) and \( \{q_t\} \) are available as the outputs of an HSMM state sequence, but there exits some random delay \( \tau \) between the two output sequences. Therefore, the observations we can obtain at time \( t \) are \( o_t \) and \( q_{t-\tau} \) though the emissions of a given state at time \( t \) are \( o_t \) and \( q_t \). In this case, delay \( \tau \) has to be estimated by maximizing the joint likelihood of the two observation sequences.

4.3. Variants of HSMM

This subsection discusses the extension of HSMMS, which includes switching HSMM, multi-channel HSMM, and adaptive factor HSMM.
4.3.1. Switching HSMM

A switching hidden semi-Markov model is defined as the concatenation of many HSMMs, with model parameter sets \( \lambda_1, \ldots, \lambda_{|Q'|} \), each initiated by a different “switching” state \( q \in Q' \), where the set of states \( Q' \) defines a Markov chain, as described in Duong et al. [54] and Phung et al. [142,143]. A two stage inhomogeneous HMM was proposed by Sitaram and Sreenivas [165] to capture the variabilities in speech for phoneme recognition. The first stage models the acoustic and durational variabilities for all distinct sub-phonemic segments and the second for the whole phoneme.

Assumes at each time step the computational complexity is \( C_q \) for computing the forward–backward variables of the qth HSMM. Because the current state sequence of the qth HSMM may start at any time step before \( t \), the computational complexity for the switching HSMM is in the order of \( O(T^3 C_q) \). Therefore, the total computational complexity for the switching HSMM is

\[
O(T^3 \sum_{q=1}^{|Q'|} C_q + |Q'|^2),
\]

where \( T \) is the total length of the observation sequence. This means the switching HSMM is applicable only for short observation sequences.

4.3.2. Multi-channel HSMM

Multi-channel HSMM was proposed to model multiple interacting processes [129]. In contrast to the basic HSMM that a process has a single state at any instant, this extension generalizes the HSMM state to be a vector \( s_t = [s_t^{(1)}, \ldots, s_t^{(C)}] \) representing the states of multiple processes. Each of the processes, say process \( c \), has an observation sequence, \( \theta_t^{(c)} \), produced by a hidden semi-Markov state sequence, \( s_t^{(c)} \), taking values in \( S_t^{(c)} \), with a set of model parameters, \( \lambda^{(c)} \), where \( T \) is the length of each of the observation sequences. Though this model can be realized in the framework of inhomogeneous HMM, the state space will be \( M^C \). To reduce the complexity of the model, some simplifying assumptions should be made for the transition probabilities. Assume \( P(s_{t+1}, d_{t+1} | s_t, d_t, \lambda) \) and the observation probabilities, \( P(o_t | s_t, \lambda) \), where

\[
d_t = [d_t^{(1)}, \ldots, d_t^{(C)}]
\]

and \( d_t^{(c)} \) is the duration having been spent in state \( s_t^{(c)} \) by time \( t \). \( o_t = [o_t^{(1)}, \ldots, o_t^{(C)}] \) is the observation vector at time \( t \). Usually, each channel is allowed to evolve independently, i.e.,

\[
P(s_{t+1}, d_{t+1} | s_t, d_t, \lambda) = \prod_{c=1}^{C} P(s_{t+1}^{(c)}, d_{t+1}^{(c)} | s_t^{(c)}, d_t^{(c)}, \lambda)
\]

and

\[
P(o_t | s_t, \lambda) = \prod_{c=1}^{C} P(o_t^{(c)} | s_t^{(c)}, \lambda).
\]

To further simplify the expressions, \( P(s_{t+1}^{(c)}, d_{t+1}^{(c)} | s_t^{(c)}, d_t^{(c)}, \lambda) \) is approximated by [129]

\[
P(s_{t+1}^{(c)}, d_{t+1}^{(c)} | s_t^{(c)}, d_t^{(c)}, \lambda) \approx \prod_{c=1}^{C} P(s_{t+1}^{(c)} | s_t^{(c)}, \lambda)
\]

or

\[
P(s_{t+1}^{(c)}, d_{t+1}^{(c)} | s_t^{(c)}, d_t^{(c)}, \lambda) = \sum_{c'=1}^{C} \theta_{c,c'} P(s_t^{(c')}, d_t^{(c')}, d_t^{(c')}, \lambda),
\]

where \( \theta_{c,c'} \) is the weight.

According to this simplified assumption, the current state-duration pair \( s_t^{(c)} \), \( d_t^{(c)} \) is dependent on the previous state-duration pair \( s_{t-1}^{(c')}, d_{t-1}^{(c')} \). There are \( MD \) current state-duration pairs for given \( c \) and \( MD \) previous state-duration pairs for given \( c' \). Therefore, the computational complexity for evaluating the forward or backward variables at each time step is \( O(M^2 D^2 C^2) \).

4.3.3. Adaptive factor HSMM

Adaptive factor HSMM is assumed to have variable model parameters for its parametric distributions of observation and/or state durations. For instance, the parameters of the observation distributions \( b_i(v_k) \) and the duration distributions \( p_i(d) \) can be changed with time or in different situations. For example, the mean \( \mu_i \) of \( b_i(v_k) \) given state \( i \) is changed for different speakers, such that for the fth speaker the mean becomes \( a^{(f)}_{i} + c^{(f)}_{i} \mu_i \), for \( f = 1, \ldots, F \), where \( \{a^{(f)}_{i}, c^{(f)}_{i}\} \) are adaptive factor, \( \mu_i \) the common parameters for all speakers, and \( F \) the total number of speakers. Let

\[
O^{(f)} = (o^{(f)}_1, o^{(f)}_2, \ldots, o^{(f)}_T_f)
\]

be the fth observation sequence of length \( T_f \), for \( f = 1, \ldots, F \), \( \lambda \) be the set of the common parameters of the HSMM, and \( A = [A^{(1)}, \ldots, A^{(F)}] \) be the set of the adaptive factors, where \( \mu_i \subset \lambda \) and \( \{a^{(f)}_{i}, c^{(f)}_{i}\} \subset A^{(f)} \)

Then the model parameters can be jointly estimated by

\[
\{\hat{\lambda}, \hat{A}\} = \arg \max_{\lambda, A} P(O^{(1)}, \ldots, O^{(F)} | \lambda, A) = \arg \max_{\lambda, A} \prod_{f=1}^{F} P(O^{(f)} | \lambda, A^{(f)}).
\]
Specific examples for the joint estimate of the model parameters are given in Yamagishi and Kobayashi [189] and Yamazaki et al. [192], where \( a_i^t \), \( a_j^t \) and \( \mu_i \) are vectors, and \( c_i^t \) is a matrix.

4.4. HSMM vs. HMM

HSMM can be considered as an extension of an HMM. Therefore, the HMM can be reversely considered as a special case of the HSMM, in considering that the duration distributions of HMM states are implicitly geometric. This subsection discusses the relationship between HSMM and the conventional HMM. A discussion about both HMM and HSMM can also be found in Kobayashi and Yu [91].

A hybrid HMM/HSMM proposed by Guedon [71,72] can be viewed as a special case of the HSMM where the occupancy distributions of some states are constrained to be geometric distributions while others are still generally distributed.

4.4.1. HSMM using HMM algorithms

In some application areas, such as speech recognition, one often uses HMM algorithms to estimate HSMM parameters to obtain benefits from both the small computational complexity of HMM algorithms and the explicit duration expression of HSMM states. As discussed in the last subsection, HMM implicitly has a geometric distribution that limits its capability in many applications whose duration is not geometrically distributed.

Duration estimation using HMM algorithms. After the optimum state sequence, \( i_t^1 \ldots i_T^t = (j_1^t, d_1^t) \ldots (j_N^t, d_N^t) \), is found via the ordinary Viterbi algorithm or the Baum–Welch algorithm [13], based on the current HMM parameters \( \{ \pi_i, a_{ij}, b_i(v_k) \} \), where \( \sum_{n=1}^{N} d_n^t = T \) and \( N \leq T, i_t, j_n^t \in S, d_n^t \in D \), the model parameters \( \hat{\pi}_i, \hat{a}_{ij}, \hat{b}_i(v_k) \), and \( \hat{p}_i(d) \) (for HSMM) can be estimated [36] by letting

\[
\eta_t(i, d) = \mathbb{I}(j_n^t = i) \cdot \mathbb{I}(d_n^t = d) \cdot \mathbb{I} \left( \sum_{k=1}^{n} d_k^t = t \right),
\]

\[
\xi_t(i, j) = \mathbb{I}(i_n^t = i) \cdot \mathbb{I}(i_{t+1}^t = j),
\]

and

\[
\gamma_t(i) = \mathbb{I}(i_n^t = i),
\]

where \( \eta_t(i, d), \xi_t(i, j) \) and \( \gamma_t(i) \) are defined by (6), (7) and (9), and \( \mathbb{I}(x) = 1 \) if \( x \) is true and zero otherwise. Then the model parameters \( \hat{a}_{ij}, \hat{b}_i(v_k), \hat{p}_i(d) \), and \( \hat{\pi}_i \) can be re-estimated. The re-estimated model parameters can be used to produce a new segmentation from which new sets of parameters are obtained until the model converges.

Certainly, the optimum state sequence, \( i_t^1 \ldots i_T^t = (j_1^t, d_1^t) \ldots (j_N^t, d_N^t) \), can also be found via the extended Viterbi algorithm for the HSMM [138,195,196], as discussed in Section 2.2.3, and then the model parameters are similarly re-estimated.

The estimated duration probabilities can be applied to modify the scores in the Viterbi algorithm on each departure from a state. This approach is non-optimal but can be ensured that the resulting state segmentation sequence is at least reasonable according to the duration specifications. During the recognition phase, the distance metric used in the Viterbi algorithm is modified as [150]:

\[
\delta_t(j) = \max_{d \in D} \max_{i \in S} \left\{ \delta_{t-d}(i)a_{ij}[p_j(d)]^\alpha \prod_{r=t-d+1}^{t} b_j(o_r) \right\},
\]

for \( 2 \leq t \leq T, j \in S \), where \( \alpha \) is a modification factor, which is usually assumed as 1. A similar modification to the forward algorithm of HMM was given in Hanazawa et al. [73]. With state duration-dependent transition probabilities the recursion in Viterbi algorithm becomes [179]

\[
\delta_t(j, d_t^t) = \max_{i \in S} \delta_{t-1}(i, d_{t-1}^t)a_{ij}(d_{t-1}^t) \cdot b_j(o_t) \},
\]

for \( j \in S, 2 \leq t \leq T \), where \( a_{ij}(d_t^t) \) denotes the probability of a transition from state \( i \) to state \( j \) of the model, given that state \( i \) has been in the current state for \( d_t^i \) consecutive time units. \( d_t^i \) denotes the current residency duration of state \( j \). \( d_t^t = d_{t-1}^t + 1 \) if the state does not change and \( d_t^t = 1 \) otherwise.

A modification to the conventional Viterbi algorithm suggested by Lee and Rabiner [104] is letting:

\[
\delta_t(i) = b_i(o_t) \max_{j \in S \setminus \{i\}} \delta_{t-1}(i) a_{ji}, \max_{j \in S \setminus \{i\}} \delta_{t-1}(j) p_j(d_{t-1}^t) a_{ji} \).
\]

The problem of this algorithm is obviously that it gives preference to the first case that there is no state change. To rectify this situation Preez [149] suggested a modified recursion by letting
\[ \delta_t(i) = b_i(\alpha_t)\delta_{t-1}(i)a_{ii}, \quad d_t = d_{t-1} + 1 \]

if \( \delta_{t-1}(i)a_{ii} \sum_{d \geq d_{t-1}} p_i(d) > \max_{j \in S_1(i)} \delta_{t-1}(j)p_j(d_{t-1})a_{ji} \); otherwise \( d_t = 1 \) and

\[ \delta_t(i) = b_i(\alpha_t) \max_{j \in S_1(i)} \delta_{t-1}(j)p_j(d_{t-1})a_{ji}. \]

Another way of using \( \{p_i(d)\} \) in the Viterbi algorithm is through introducing temporal constraints in the HMM, that is, let [194]

\[
\begin{align*}
\alpha_{ii} &= \frac{\sum_{d=t+1}^{D} p_i(d)}{\sum_{d=t}^{D} p_i(d)}, \\
\alpha_{ij} &= \frac{p_i(1)}{\sum_{d=t}^{D} p_i(d)}
\end{align*}
\]

where \( \tau \) is the duration of a given state \( i \).

**Bounded state durations.** It is possible to implement upper and lower bounds on duration without specific probabilistic modelling [67,181,90,102]. The bounded state durations can be estimated in the training phase [102] or after the training phase by finding the global minimum and maximum durations for each state [67,90]. Following the definitions in the previous sections, which define \( \alpha_t(j) = P[s_t = j, o_{1:t} | \lambda] \) and \( \alpha_t(j, d) = P[s_{t-d+1:t} = j, o_{1:t} | \lambda] \), Laurila [102] yields

\[
\begin{align*}
\alpha_{t+1}(j, 1) &= \sum_{i \in S(j)} \alpha_t(i)a_{ij}b_j(\alpha_{t+1}), \\
\alpha_{t+1}(j, d + 1) &= \alpha_t(j, d)a_{jj}b_j(\alpha_{t+1}),
\end{align*}
\]

and

\[
\alpha_{t+1}(j) = \sum_{d = D_{\min}}^{D_{\max}} \alpha_{t+1}(j, d),
\]

where \( D_{\min} \) and \( D_{\max} \) are the bounds of the state durations. The bounded state duration model resulted in quite loose state duration constraints and were not effective enough in the recognition phase.

### 5. Applications of HSMM

The HSMM has been applied to thirty areas, such as human activity recognition [38,55,56,79,80,119,120,128–131,140,197,200,205,206], handwriting recognition or printed text recognition [4,15,16,27,32–35,95,96,163], network traffic characterisation and anomaly detection [105,111,116,117,139,153,172,177,187,188,198,201], speech recognition [31,39,67,76,107,127,137,148,152,155], speech synthesis [125,132–134,169–171,186,189–192,204], functional MRI brain mapping [58,59,174–176], electrocardiograph (ECG) [53,85,84,174], recognition of human genes in DNA [19,24,173], language identification [118], ground target tracking [88], document image comparison and classification at the spatial layout level [81,82], change-point/end-point detection for semi-conductor manufacturing [64,65], protein structure prediction [8,12,162], the analysis of branching and flowering patterns in plants [69–72], rain events time series model [3,159,160], satellite propagation channel modelling [112], event recognition in videos [79], mobility tracking in cellular networks [120,197,200], Internet traffic modelling [198], image segmentation [21,98,184], semantic learning for a mobile robot [167], symbolic plan recognition [118], adaptive cumulative sum test for change detection in non-invasive mean blood pressure trend [193], equipment prognosis [14,48–51,75], financial time series modelling [22], classification of music [113], remote sensing [147], and prediction of particulate matter in the air [52].

Among those applications, the major applications include speech recognition, speech synthesis, human activity recognition, handwriting recognition, network traffic modelling & anomaly detection, and functional MRI brain mapping.

#### 5.1. Human activity recognition

How to learn and recognize human activities of daily living (ADL) is an important research issue in building a pervasive and smart environment. Yu and Kobayashi [200] proposed an explicit duration HMM for missing data and multiple observation sequences, and applied the model into mobility tracking in wireless networks [120,197]. Hongeng and Nevatia [79], Hongeng et al. [80] and Zhang et al. [206] applied HSMM for recognizing events in a video surveillance. Pavel et al. [140] used the model in unobtrusive assessment of mobility. Marhasev et al. [119] used a non-stationary HSMM in activity recognition. Niwase et al. [131] used in human walking motion synthesis.

Different from the usual application of HSMM, Duong et al. [54–56] introduced a two-layered extension of HSMM for modelling the ADL. In this model, sequences of major activities of human daily routine, such as making breakfast, eating
breakfast, going to work, coming back home, are modelled by a Markov chain, with each state representing a major activity. For a given major activity, atomic activities such as spending time at the cupboard, stove, fridge, or moving between these designated places, are described by a hidden semi-Markov model. Each HSMM state represents an atomic activity and its duration represents the atomic action time.

Similarly, Zhang et al. [205] used a layered HSMM in an intelligent surveillance system deployed in parking lots. Chung and Liu [38] applied a hierarchical context hidden Markov model for behavior understanding from video streams in a nursing center. The two-layered HSMM has been discussed in Section 4.3.1 “Switching HSMM” of this paper. A three-layered variable transition hidden Markov model was introduced by Natarajan and Nevatia [130] for representing the composite actions at the top-most layer, the primitive actions at the middle layer and the body pose transitions at the bottom-most layer.

As an extension to the layered HSMM, Natarajan and Nevatia [128] proposed a hierarchical multi-channel HSMM. A multi-channel HSMM is used to model multiple interacting processes, which has been discussed in Section 4.3.2 “Multi-channel HSMM.” As a special case of multi-channel HSMM, a coupled HSMM is used for continuous sign-language recognition in Natarajan and Nevatia [129].

5.2. Handwriting recognition

In the application of HSMM in handwriting recognition, the 26 letters in the alphabet are defined as 26 different states of an HSMM, and the number of subcharacter symbols in a letter as the duration of the state. For instance, “B” has three subcharacter symbols and “P” has two. Therefore, the key problem is to develop a segmentation algorithm to translate the 2-D image of written text into a 1-D sequence of subcharacter symbols. The 1-D sequence is used as the observation sequence of the HSMM.

Chen et al. [32,33,35] proposed a robust segmentation algorithm based on mathematical morphology and used a mixture Gaussian distribution to model the subcharacter symbol probability distribution for each state. The Viterbi algorithm given by (11) is used in finding the best path of states for the recognition of letters or words [34,35,95,96,27,28,15]. Senior et al. [163] compared normal, gamma and Poisson distributions to the original histogram of lexeme durations. They found that the Poisson distribution most closely matched the desired distribution, giving improved performance over the un-smoothed histogram. The other distributions did not perform as well.

5.3. Network traffic characterization and anomaly detection

In this application, the HSMM is applied to characterize the network traffic. Measurements of real traffic often indicate that a significant amount of variability is present in the traffic observed over a wide range of time scales, exhibiting self-similar or long range dependent characteristics [105]. Such characteristics can have a significant impact on the performance of networks and systems [177,139]. Therefore, better understanding of the nature of network traffic is critical to the proper design and implementation of servers and anomaly detection systems for network security purpose. A major advantage of using an HSMM is the capability of capturing various statistical properties of the traffic, including the long-range dependence [198]. They can also be used together with, for example, matrix-analytic methods to obtain analytically tractable solutions to queueing-theoretic models of server performance [153].

In this application, the observation sequence \( (o_t) \) represents the number of user requests, packets, bytes, connections, etc., arriving in the \( t \)th second, or the interarrival time between requests or packets. The observation sequence is characterized as a discrete-time random process modulated by an underlying (hidden state) semi-Markov process. The hidden state represents the density of traffic or mass of active users.

Using the model parameters of the HSMM trained by the normal behavior, one can detect anomaly embedded in the network behavior according to its likelihood or entropy against the model [201,111,116,117,187,188].

5.4. Speech recognition and speech synthesis

In this application, observation vectors are obtained by extracting features from the speech signal through a spectral and/or temporal analysis. The observation vectors can be used to train the HSMMs which characterize various speech units. Speech units can be linguistically based sub-word units such as phones and acoustic units, whole word units, and units which contain a group of words. Each unit is characterized by an HSMM whose states can be considered as its distinct sounds (e.g., phonemes, syllables). A lot of applications of HSMM in speech recognition can be found in the literature, such as Levinson [107], Codogno and Fissore [39], Nakagawa and Hashimoto [127], Gu et al. [67], Ratnayake et al. [152], Hieronymus et al. [76] and Oura et al. [137]. The applications in speech synthesis can be found in Zen et al. [204], Yamagishi and Kobayashi [189,191], Yamagishi et al. [190], Yamazaki et al. [192], Tachibana et al. [169–171], Nose et al. [132–134], Moore and Savic [125] and Wu et al. [186].

To associate time with model states, a left–right model of HSMM is usually assumed. The number of states can be selected to correspond roughly to the number of sounds (e.g., phonemes) within the unit, or to the average number of observations in a spoken version of the speech unit. The output probability \( b_j(v_k) \) of any state \( j \) is often represented by a mixture of Gaussian distributions with a diagonal covariance matrix [137]. The duration distribution of state \( j \) is
usually assumed as a parametric one such as Gaussian distribution [137], Poisson distribution [155], and gamma distribution [107,39]. The uniform distribution with lower and upper bounding parameters was also applied, which inhibited a state occupying too few or too many speech frames [67].

The model parameters can be trained by the re-estimation algorithms for an HSMM. But in fact in the area of speech recognition, they are often estimated using the Viterbi algorithms, as discussed in Section 4.4.1.

5.5. Functional MRI brain mapping

Applying an HMM in functional magnetic resonance imaging (fMRI) brain mapping is to reveal components of interest in the fMRI data, as did in Faisan et al. [58,59], Thoraval [176] and Thoraval et al. [175,174]. This enables the model to automatically detect neural activation embedded in a given set of fMRI signals. It allows to enrich brain mapping with activation lag mapping, activation mode visualizing, and hemodynamic response function analysis. The three problems of HRF (the hemodynamic response function) shape variability, neural event timing, and fMRI response linearity can be solved by the model.

In this application, the sequence of hemodynamic response onsets (HROs) observed in the fMRI signal is used as the observation sequence, which is usually composed of events mixed with missing observations (null). A left–right Markov chain is selected for the hidden process of task-induced neural activations. The state index $i$ reflects the order of appearance of the event in the chain. The chain is registered temporally with a sequence of discrete time OFF–ON blocks. The probability that the first observation at the start time $t_1$ of state $i$ is $o_{i1}$ is denoted as $b_i(o_{i1})$. The inter-state duration distributions are specified as one-dimensional gaussians. At time $t$ during the transition time interval from state $i$ to state $j$, the probability of missing an observation is $1 - e_{ij}$ and having an observed event $o_{it} = e_{ij}$ with the observation probability $b_{ij}(e_{ij})$. All observation distributions are specified as one-dimensional gaussians.

6. Conclusions and remarks

The issues related to a general HSMM include (a) the forward–backward algorithm; (b) the computation of the predicted/filtered/smoothed probabilities, expectations and the likelihood of observations; (c) the MAP estimate of states and the MLE estimate of state sequence by extended Viterbi algorithm; (d) the parameter estimate/update and the order estimate of the model; (e) the implementation of the forward–backward algorithms.

By introducing certain assumptions and some constraints on the state transitions, the general HSMM becomes the traditional explicit duration HMM, variable transition HMM, or residential time HMM. Those conventional models have different capability in modelling applications. They also have different computational complexity and memory requirement involved in the forward–backward algorithms and the model estimate, as shown in the following table:

<table>
<thead>
<tr>
<th>The model</th>
<th>Complexity</th>
<th>Mem. for F–B</th>
<th>Mem. for estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>E. D. HMM</td>
<td>$O(\frac{M^2 + MD^2 + D}{2}T)$</td>
<td>$O(MT)$</td>
<td>$O(M^2 + MD + MK + MT)$</td>
</tr>
<tr>
<td>V. T. HMM</td>
<td>$O(\frac{(M^2 + MD^2)}{2}T)$</td>
<td>$O(MDT)$</td>
<td>$O(M^2D + MK + MDT)$</td>
</tr>
<tr>
<td>R. T. HMM</td>
<td>$O(\frac{(MD + M^2)}{2}T)$</td>
<td>$O(MT)$</td>
<td>$O(M^2D + MD + MK + MT)$</td>
</tr>
</tbody>
</table>

where $M$ is the number of hidden states, $D$ the maximum duration between successive state transitions, $K$ the number of observable values, and $T$ the period of the observation data. However, the three models are interchangeable to each other if we express the state transition probabilities and the state duration distributions of one model with the parameters of the other one.

The state duration distributions and the observation distributions can be non-parametric or parametric dependent on the specific preference of the applications. Among the parametric distributions, the most popular ones are the exponential family distribution, such as Poisson, exponential, Gaussian, and the mixture of Gaussian distributions. The Coxian distribution of duration can represent any discrete pdf, and the underlying series-parallel network also reveals the structure of different HSMMs.

Observations are usually assumed to be dependent on states that produce them and are conditionally independent to each other for given states. However, the segmental model assumes the observations to be dependent not only on the emission state but also on the state duration.

The variants of HSMM include switching HSMM, multi-channel HSMM, and adaptive factor HSMM, which are suitable for the applications that cannot be described by a homogenous process. In considering that the duration of an HMM state is geometric, it is possible to use the algorithms of HSMMs for the HMM, which show different forms from the conventional Baum–Welch algorithm and the Viterbi algorithm.

There are three methods that can be used for overcoming the underflow problem embedded in the implementation of the forward–backward algorithms for the HSMMs. Among those three methods, the posterior notion performs best because the heuristic scaling cannot guarantee the backward variables from underflow and the logarithmic formulation cannot be very accurate based on the lookup table.

It has proved that the iterative re-estimation procedure of the model parameters using the expectations is equivalent to the EM algorithm. It maximizes the likelihood function for given observations. In a non-stationary situation, the model
parameters have to be updated online with time or with the increase of observation sequence length. Therefore, the re-
estimation algorithms based on the forward–backward algorithms become unsuitable. Instead, one should use the online
algorithms for the real-time update of the model parameters, which are usually done by maximizing the likelihood functions.
Sometimes, to reduce the computation amount required in the re-estimation of the HSMM parameters, non-maximum
likelihood estimator is used. For instance, the Viterbi algorithms or the forward–backward algorithms for the HMM are used
to estimate the state duration distributions for the HSMMs. The method is splitting the states into segments.
The HSMMs have been applied in thirty areas. More and more papers are published in the literature. The number of
papers published in each five years, which are cited in this survey, is listed in the following table:

<table>
<thead>
<tr>
<th>Years</th>
<th>No. of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980–1984</td>
<td>3</td>
</tr>
<tr>
<td>1985–1989</td>
<td>16</td>
</tr>
<tr>
<td>1990–1994</td>
<td>29</td>
</tr>
<tr>
<td>1995–1999</td>
<td>35</td>
</tr>
<tr>
<td>2000–2004</td>
<td>38</td>
</tr>
<tr>
<td>2005–2008</td>
<td>77</td>
</tr>
</tbody>
</table>

We note that there are about one hundred papers (mainly in speech recognition) that use the HSMMs but do not
contribute to the theory or algorithm of the HSMMs are not cited in this paper.

Acknowledgements

We would like to acknowledge support for this project from the State Key Program of NSFC-Guangdong Joint Funds
(U0735002) and The National High Technology Research and Development Program of China (2007AA01Z449).

References

[8] Z. Aydin, Y. Altunbasak, M. Borodovsky, Protein secondary structure prediction for a single-sequence using hidden semi-Markov models, BMC Bioin-
[16] R. Bippus, V. Margner, Script recognition using inhomogeneous P2DHMM and hierarchical search space reduction, in: Proceedings of the Fifth Inter-
1547–1550.
[18] A. Bonafonte, J. Vidal, A. Nogueiras, Duration modeling with expanded HMM applied to speech recognition, in: Fourth International Conference on
2192–2209.
240–242.
...


