Practical Implementation of an Efficient Forward–Backward Algorithm for an Explicit-Duration Hidden Markov Model

Shun-Zheng Yu and Hisashi Kobayashi, Life Fellow, IEEE

Abstract—This correspondence addresses several practical problems in implementing a forward–backward (FB) algorithm for an explicit-duration hidden Markov model. First, the FB variables are redefined in terms of posterior probabilities to avoid possible underflows that may occur in practice. Then, a forward recursion is used that is symmetric to the backward one and can reduce the number of logic gates required to implement on a field-programmable gate-array (FPGA) chip.

Index Terms—Explicit-duration hidden Markov model (HMM), forward–backward (FB) algorithm, hidden Markov model (HMM), hidden semi-Markov model (HSMM), variable duration HMM.

I. INTRODUCTION

In the literature, a hidden Markov model (HMM) is commonly defined by a discrete-time finite-state homogeneous Markov chain observed through a finite set of transition densities indexed by the states of the Markov chain. As an extension of the HMM, an explicit-duration HMM or variable-duration HMM is traditionally defined by allowing the underlying process to be a semi-Markov chain with variable duration of each state that is associated with the number of observations produced while in the state. The explicit-duration HMM is also called “HMM with variable duration” [1], [2], “HMM with explicit duration” [3], and “hidden semi-Markov model” (HSMM) in the literature. The explicit-duration HMM has been applied to such problems as event recognition in videos [4], mobility tracking in cellular networks [5], [6], brain functional MRI sequence analysis [7], Internet traffic modeling [8], handwritten word recognition [9], [10], language identification [11], speech recognition [1]–[3], [12]–[14], and electrocardiograph (ECG) [15].

There are three approaches to the explicit-duration HMM. The first approach was proposed by Ferguson [1] (see also the survey paper by Rabiner [12]), who defines the forward–backward (FB) variables by the joint probabilities of the state ending at a certain time and a series of observations produced while in the state. The joint probability distribution of a sequence of observations up to that time. The joint probability distribution of the observations is required in the Ferguson algorithm and given in a product form, can be calculated more efficiently by a recursive method as suggested by Levinson [2] and further refined by Mitchell, Harper, and Jamieson [3]. The second approach is to transform a given explicit-duration HMM into an equivalent “super HMM” [13], [16]. In this case, however, the refined algorithms are still computationally too intensive in some applications. The third approach was proposed recently by the present authors [17], which defines the FB variables using the notion of a state together with its remaining sojourn (or residual life) time. It shows that it is the most efficient FB algorithm among all these three approaches. This makes the algorithm practical in many applications. The algorithm proposed in this correspondence will achieve the same computational efficiency. In addition to those FB algorithms of the three approaches for the explicit-duration HMM, the state duration distributions can be taken into account in the Viterbi algorithm as in [21]–[23], where the state duration distributions can be estimated efficiently for a left–right HMM.

It is well known, however, that the joint probabilities associated with observation sequence often decay exponentially as the sequence length increases. The implementation of the FB algorithms by programming in a real computer would suffer a severe underflow problem. A general heuristic method to solve this problem is to rescale the FB probabilities by multiplying a large factor whenever an underflow is likely to occur [14]. By replacing the joint probabilities with conditional ones, the FB algorithms for the standard HMM can automatically avoid the underflow problem [18], [19]. However, there exists no such method reported in the literature for the explicit-duration HMM that can automatically overcome the same problem. In this correspondence, we similarly redefine the FB variables using the notion of posterior probabilities. This results in essentially the same FB algorithms for the explicit-duration HMM that are robust against the underflow problem.

Furthermore, the existing algorithms for explicit-duration HMM are not practical for implementation in hardware because of the computational or logic complexity. By slightly modifying the forward recursion, we demonstrate in this correspondence that the forward and backward recursions can be made exactly symmetric to each other. Therefore, the forward logic modules can be utilized in the backward procedure as well and will considerably reduce the requirement for the silicon area on a chip.

The rest of the correspondence is organized as follows. Section II defines the model parameters. In Section III, we define various conditional probabilities for the predicted, filtered, and smoothed estimates of the state; then in Sections IV and V, the modified FB algorithm is provided for calculating those variables, and the re-estimation method is refined for training the model. Finally, Section VI concludes the correspondence.

II. DEFINITIONS

Let \( s_1, s_2, \ldots, s_{ht} \) be states of a semi-Markov chain, with the initial distribution \( \{ \pi_m \} \) and the transition probability matrix \( \{ a_{mn} \} \). Let \( q_t \) denote the state of the semi-Markov chain at time \( t \), \( t = 1, 2, \ldots, T \), and \( o_t \) the observable output with the conditional probability

\[
\hat{b}_m(v_k) \triangleq P(o_t = v_k|q_t = s_m)
\]

where \( \{ v_k \} \) is a set of \( K \) distinct values that may be assumed by the observation \( o_t \). Denote the observation sequence from time \( a \) to \( b \) as \( o_a^b \). We also assume that the duration of a given state is a discrete random variable, taking value \( d \) with probability \( p_m(d) \), where \( d \in \{1, 2, \ldots, D\} \).

Let \( \tau_i \) denote the remaining sojourn (or residual life) time of the current state \( q_t \). Then, if the pair process \( (q_t, \tau_i) \) takes on value \( (s_m, d) \) at, say, time \( t_0 \), then the semi-Markov chain will remain in the current state \( s_m \) until time \( t_0+d-1 \) and transit to another state at time \( t_0+d \), where \( d \geq 1 \). For brevity of notation, let \( \lambda \) stand for the complete set of model parameters. We evaluate the various probabilities \( \text{conditioned on the parameter set} \ \lambda \), but we often drop this conditioning on \( \lambda \) to simplify the notation.

III. PREDICTED, FILTERED, AND SMOOTHED PROBABILITIES

We define a variable

\[
\alpha_{q_t|m}(m, d) \triangleq P(q_t = s_m, \tau_t = d|o_t^d)
\]
where \( x = t - 1, t \) or \( T \). The above quantity is termed the “predicted,” “filtered,” or “smoothened” probability of \((q_t, \tau_t)\), depending on that the observation sequence is \( o_{t-1}^t, o_t^t, \) or \( o_t^T \).

Correspondingly, we denote the marginal probability distribution of \( q_t \) by summing over all \( d \)

\[
\gamma_{t|x}(m) \triangleq \sum_d \alpha_{t|x}(m, d)
\]

which represents the filtered, predicted, and smoothed conditional probability of state \( q_t = s_m \), given the observed sequences \( o_{t-1}^t, o_t^t, \) and \( o_t^T \), respectively.

We define the ratio of the filtered probability \( \alpha_{t|x}(m, d) \) over the predicted one \( \alpha_{t|-1}(m, d) \) by

\[
b_m^*(o_t) \triangleq \frac{\alpha_{t|x}(m, d)}{\alpha_{t|-1}(m, d)} = \frac{b_m(o_t)}{P(o_t | o_{t-1}^t)}
\]

for any \( d \), where in the derivation of the second expression the relation \( P(o_t | q_t = s_m, \tau_t = d, o_{t-1}^t) = P(o_t | q_t = s_m) \) is applied because of the Markov property. Obviously, this ratio approximates \( 1 \) when the model fits to the observations well.

We denote the one-step prediction of the observation, \( P(o_t | o_{t-1}^t) \), by \( r_t^{-1} \), with \( r_t^{-1} = P(o_t | o_{t-1}^t) \). It can be determined by

\[
r_t^{-1} \triangleq \frac{P(o_t | o_{t-1}^t)}{\sum_{m,d} \alpha_{t|-1}(m,d)b_m(o_t)}
\]

and

\[
= \frac{\sum_{m} \gamma_{t|-1}(m)b_m(o_t)}{\sum_{m} \gamma_{t|-1}(m)b_m(o_t)}
\]

The likelihood function of the observation sequence is given by

\[
P(o_{1:T}^T) = \left( \prod_{t=1}^{T} r_t^{-1} \right)^{-1}
\]

We further define a similar but different variable

\[
D_{t|x}(m, d) \triangleq P(\tau_{t-1} = 1, q_t = s_m, \tau_t = d | o_{t-1}^t)
\]

which represents the conditional probability that state \( s_m \) starts at time \( t \) and lasts for \( d \) time units (i.e., ends at \( t + d - 1 \)) given the observations \( o_{t}^T \), where \( x = t - 1, t + d - 1 \) or \( T \). As for a state transition from \( s_m \) to \( s_n \), we define

\[
T_{t|x}(m, n) \triangleq P(q_t = s_m, \tau_t = 1, q_{t+1} = s_n | o_{t-1}^t)
\]

and

\[
S_{t|x}(m) \triangleq P(\tau_t = 1, q_{t+1} = s_m | o_{t-1}^t) = \sum_n T_{t|x}(m, n)
\]

where for simplicity the subscript \( t \) in \( E_t(\cdot) \) and \( S_t(\cdot) \) denotes the condition \( o_{t}^T \). The corresponding smoothed probabilities are

\[
E_{t}|_{x}(m, n) = \sum_{T_{t|x}(m, n)} E_{t-1}(m) \quad \text{and} \quad S_{t}|_{x}(m) = \sum_{D_{t|x}(m, d)} D_{t|x}(m, d)
\]

To calculate the smoothed probabilities, we define the backward variable by the ratio of the smoothed probability \( \alpha_{t|x}(m, d) \) over the predicted one \( \alpha_{t|-1}(m, d) \), i.e.,

\[
\beta_{t}(m, d) \triangleq \frac{P(q_t = s_m, \tau_t = d | o_{t-1}^T)}{\sum_{m,d} \alpha_{t|-1}(m,d)b_m(o_t)}
\]

\[
= \frac{P(o_t | q_t = s_m, \tau_t = d)}{P(o_{t-1}^T | o_{t-1}^t)}
\]

with the initial value

\[
\beta_T(m, d) = b_{T}^*(o_T)
\]

for all \( d \), where the derivation is based on the following decomposition:

\[
P(q_{t}, \tau_t | o_{t-1}^T) = \frac{P(q_{t}, \tau_t | o_{t-1}^T)}{P(o_{t-1}^T | o_{t-1}^t)} \frac{P(o_{t-1}^T | o_{t-1}^t)}{P(o_{t-1}^T | o_{t-1}^t)}
\]

The similar decompositions will be implicitly applied to the derivations in the rest of the correspondence. For convenience in the backward recursion, we denote two variables that are symmetric to \( E_t(\cdot) \) and \( S_t(\cdot) \) by

\[
E_{t}^{*}(m) \triangleq \frac{P(o_{t} | q_t = s_m, \tau_t = 1)}{P(o_{t-1}^T | o_{t-1}^t)} \sum_{d} D_{t|x}(m, d)b_m(o_t)
\]

and

\[
S_{t}^{*}(m) \triangleq \frac{P(o_{t} | q_t = s_m, \tau_t = 1)}{P(o_{t-1}^T | o_{t-1}^t)} \sum_{m} E_{t-1}(m)\alpha_{t|-1}(m, 1)
\]

Therefore, the smoothed probability of state \( s_m \) with residual time \( d \) at \( t \) is given by

\[
\alpha_{t|x}(m, d) = \alpha_{t|-1}(m, d)\beta_t(m, d)
\]

The smoothed probability that a transition from state \( s_m \) to state \( s_n \) at \( t \) occurs is

\[
T_{t|x}(m, n) = E_{t-1}(m)\alpha_{t|-1}(m, 1)S_{t|x}(n)
\]

and the probability that state \( s_m \) is entered at \( t \) and lasts for \( d \) time units is

\[
D_{t|x}(m, d) = S_{t-1}(m)p_{m}(d)\beta_t(m, d)
\]

Therefore, to calculate various smoothed probabilities, we need to calculate the variables \( \alpha_{t|x}(m, d) \) and \( \beta_t(m, d) \) with the auxiliary variables \( E_{t}(m), S_{t}(m), E_{t}^{*}(m), \) and \( S_{t}^{*}(m) \). Now we derive the forward and backward recursion formulae for computing all these variables.

### IV. FORWARD–BACKWARD ALGORITHM

First, we derive the forward recursion formulae. Since state \((q_t, \tau_t) = (s_m, d)\) may transit from any other state (including self-transition) or continue the previous state, i.e., \((q_t, \tau_t = 1) = (s_m, d + 1)\), we readily obtain the following forward recursion formula:

\[
\alpha_{t|-1}(m, d) = S_{t-1}(m)p_{m}(d) + b_{T}^*(m)\alpha_{t-1|-2}(m, d + 1)
\]

with the initial value

\[
\alpha_{1|-1}(m, d) = \pi_{m}p_{m}(d)
\]

Now we derive the backward formulae. By examining all possible states that follow \((q_t, \tau_t) = (s_m, d)\), we see that when \( d = 1 \) the next state can be \((q_{t+1}, \tau_{t+1}) = (s_n, d')\) for any \( n \) and \( d' \geq 1 \), and when \( d > 1 \) it must be \((q_{t+1}, \tau_{t+1}) = (s_m, d - 1)\). Since for \( d > 1 \)

\[
P(q_{t} = s_m, \tau_t = d, q_{t+1} = s_n, \tau_{t+1} = d - 1 | o_{t-1}^T) = \frac{P(q_{t} = s_m, \tau_t = d | o_{t-1}^T)}{P(o_{t-1}^T | o_{t-1}^t)} \beta_{t+1}(m, d)P(o_{t} | q_{t} = s_m, \tau_t = d - 1 | o_{t-1}^T)
\]

with the initial value

\[
\beta_{T}(m, d) = b_{T}^*(o_T)
\]
we have the following backward recursion formula:
\[
\beta_t(m, d) = \begin{cases} 
S_{t-1}(m) b_m^*(o_t), & d = 1 \\
\beta_{t+1}(m, d - 1) b_m^*(o_t), & d > 1.
\end{cases} \tag{14}
\]

Thus, the FB algorithm using variables of \(\alpha_t | m, 1\) and \(\beta_t(m, d)\) can be summarized as follows:

**Forward-backward algorithm**

**The forward recursion**: for \(t = 1, \ldots, T\)
- Calculate \(\alpha_{0\cdot t-1}(m, d)\) by (12) (or (13)) when \(t = 1\);
- \(b_m^*(o_t)\) by (1), (3) and (2);
- \(E_t(m)\) by (5), and
- \(S_0(m)\) by (6).

The values \(\{b_m^*(o_t) : t = 1, \ldots, T, m = 1, \ldots, M\}\) or the factors \(\{r_t : t = 1, \ldots, T\}\) need to be stored for the backward computations.

**The backward recursion**: for \(t = T, \ldots, 1\)
- Calculate \(b_m^*(o_t)\) by (2);
- \(\beta_t(m, d)\) by (14) (or (7) when \(t = T\));
- \(E_T(m)\) by (8); and
- \(S_T(m)\) by (9).

The forward–backward recursions can be readily written using compact matrix notation. They can also be implemented in the logarithmic domain like the maximum a posteriori (MAP) and Viterbi algorithms used for turbo-decoding in digital communications, which transforms multiplications into additions and additions into maximization operations augmented with a small lookup table. When \(D = 1\), the FB algorithm becomes the standard HMM algorithm (or the Baum–Welch algorithm [12]). If we redefine \(b_m(o_t)\) as \(\sum_{s_t} P(x_t | q_{t-1} = s_t, q_t = m)p(o_t | x_t)\), where \(x_t\) is the transmitted symbol at time \(t\) when the system transits from state \(s_t\) to state \(s_n\), and \(o_t\) the received signal plus noise, then the FB algorithm with \(D = 1\) reduces to the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [24], which is often used to perform MAP estimation in turbo decoding in digital communications.

Obviously, updating the forward variables at every \(t\) requires \(O(MD + M^2)\) multiplications. Similarly, evaluation of the backward variables at each \(t\) also requires \(O(MD + M^2)\) multiplications. Hence, the total number of multiplications for evaluating the forward and backward variables is \(O\left((MD + M^2)T\right)\), where \(T\) is the total number of observations.

Once \(\alpha_{t | m - 1}(m, d), E_t(m), S_t(m), \beta_t(m, d), E'_t(m), \) and \(S'_t(m)\) are determined, all other predicted, filtered and smoothed probabilities can be determined.

In some applications, we may need a further reduction of the computational burden or memory capacity required for evaluating the variables. As a direct result of (10)–(14), \(\alpha_{t | m} \) can be rewritten as
\[
\alpha_{t | m} = \alpha_{t | m - 1} + \beta_{t | m} - S_{t | m} b_m(o_t).
\tag{15}
\]

where \(\alpha_{t | m - 1}(m, d) = E_{t | m - 1}S'_{t | m}\) based on (5), (10), and (14). It will reduce the computational amount by \(MDT\) multiplications in performing (10) and memory capacity by \(MDT\) units in storing the forward variable \(\alpha_{t | m - 1}(m, d)\). Similarly, summing (15) over \(d\) and using (5), (8), (11), and (14), \(\gamma_{t | m}\) can be rewritten as
\[
\gamma_{t | m} = \gamma_{t | m - 1} + E_{t | m - 1}S'_{t | m} - S_{t | m} E'_t(m).
\tag{16}
\]

It is apparent that a similar reduction is achieved because it avoids performing (1) and (10). We note that the subtractions performed in (15) and (16) may result in a negative value (e.g., \(-10^{-36}\)) due to a finite digit representation of real numbers in a computer. Therefore, in a program to implement the algorithm, let \(\alpha_{t | m} = 0\) if \(\alpha_{t | m} < 0\), and \(\gamma_{t | m} = 0\) if \(\gamma_{t | m} < 0\).

As an alternative to the forward variable \(\alpha_{t | m - 1}(m, d)\), we consider \(D_{t | d} \alpha_{t | m} (m, d)\). It is easy to derive that
\[
D_{t | d} \alpha_{t | m} = S_{t - d}(m)p_m(d) \sum_{i=1}^{d} b_m(o_i).
\tag{17}
\]

Hence, we define an alternative forward variable by
\[
\rho_t(m, d) = \frac{D_{t | d} \alpha_{t | m} (m, d)}{p_m(d)}
\tag{18}
\]

From (17), it can be seen that, for \(d > 1\)
\[
\rho_t(m, d - 1) = \frac{S_{t - d}(m)}{S_{t - d - 1}(m)} \prod_{i=1}^{d-1} b_m(o_i).
\tag{19}
\]

Therefore, we have the alternative forward recursion
\[
\rho_t(m, d) = \left\{ \begin{array}{ll}
S_{t - d}(m) b_m(o_t), & d = 1 \\
\rho_{t - 1}(m, d - 1) b_m(o_t), & d > 1
\end{array} \right.
\tag{20}
\]

with the initial values
\[
S_0(m) = \pi_m, \quad \rho_0(m, d) = 0
\tag{21}
\]

for all \(m\) and \(d\). From (12) and (20), \(\rho_t(m, d)\) and \(\alpha_{t | m - 1}(m, d)\) have the relationship
\[
\alpha_{t | m - 1}(m, d) b_m(o_t) = \sum_{i=1}^{d} \rho_t(m, i)p_m(d + i - 1).
\]

That is, \(\alpha_{t | m - 1}(m, d)\) can be determined by \(\rho_t(m, d)\), and vice versa. From (5), this relationship leads to
\[
E_t(m) = \sum_{d=1}^{D} \rho_t(m, d)p_m(d)
\tag{22}
\]

and, from (1) and (12)
\[
\gamma_{t | m} = S_{t - 1}(m) + \sum_{d=1}^{D} \rho_{t - 1}(m, d) \sum_{d'=d+1}^{D} p_m(d')
\tag{23}
\]

with initial value
\[
\gamma_1(m) = \pi_m
\]

where \(\sum_{d=d+1}^{D} p_m(d')\) is the tail distribution of the state duration \(d\). Hence, the factor \(r_{t | m}^{-1}\) given by (3) can be alternatively determined by \(\rho_{t - 1}(m, d)\). Therefore, we have the following.

**The alternative forward recursion**: for \(t = 1, \ldots, T\)
- calculate \(b_m(o_t)\) by (2), (3), and (23);
- \(\rho_t(m, d)\) by (20) (or (20) and (21) when \(t = 1\));
- \(E_t(m)\) by (22);
- \(S_t(m)\) by (6).

This alternative forward recursion is symmetric to the backward one except that in the backward recursion the factor \(r_{t | m}\) is given.

V. MAP ESTIMATION AND PARAMETER RE-ESTIMATION

Using the smoothed probabilities, \(\gamma_{t | m}, \{ D_{t | m}(m, d) \},\{ E_t(m, d) \},\{ T_t(m, n) \}\), we can make MAP estimates of the state of the system at time \(t\), including: i) state \(q_t\), ii) state \(q_t\) with a residual time \(d_t\), iii) state \(q_t\) with a dwell time \(d\), and iv) a transition from state \(q_t\) to \(q_{t+1}\), respectively, all conditioned on the observations
made in the interval \([1, T]\). For instance, the MAP estimate of the state that starts at \(t\) and lasts for a period given the observation \(a^t\) is given by
\[
\hat{q}_m = \arg \max_{(m,d)} P \left( s_m \text{ starts at } t, \tau_t = d \big| a^t \right)
\]
\[
= \arg \max_{(m,d)} D_{q,T}(m,d).
\] (24)

In the FB algorithm, we assume that the set of model parameters \(\lambda\) is given. In the case that \(\lambda\) is not given, we need to train the model for given observations \(\{a_t\}\), where \(\lambda\) is initially estimated and then re-estimated multiple times until the likelihood of the observations increases and converges to a certain value. In some other cases when the system is slowly varying (i.e., nonstationary), the model parameters \(\lambda\) may need to be updating adaptively. Such training and updating process is referred to as parameter re-estimation.

Similarly, the smoothed probabilities can be used in maximum-likelihood re-estimation of the model parameters
\[
\hat{\alpha}_{m,n} = \sum_{t=m}^{T} \frac{T_{q,T}(m,n)}{N_q} \quad (25)
\]
\[
\hat{\beta}_{m}(d) = \sum_{t=m}^{T} \frac{D_{q,T}(m,d)}{N_p} \quad (26)
\]
\[
\hat{\pi}_m = \frac{\gamma_{q,T}(m)}{N_x} \quad (27)
\]
\[
\hat{b}_{m}(v_k) = \sum_{t=1}^{T} \frac{\gamma_{q,T}(m) \delta(a,v_k)}{N_k} \quad (28)
\]
\[
\hat{b}_{m}(v_k) = \sum_{t=1}^{T} \frac{\gamma_{q,T}(m) \delta(a,v_k)}{N_k} \quad (29)
\]
where \(N_q, N_p, N_x,\) and \(N_k\) are the factors used for normalizing the estimated parameters such that \(\sum_{m} \hat{\alpha}_{m,n} = 1, \sum_{m} \hat{\beta}_{m}(d) = 1, \sum_{m} \hat{\pi}_m = 1,\) and \(\sum_{v_k} \hat{b}_{m}(v_k) = 1\) (for \(v_k\)) is the set of values that an observation \(a_t\) can take on, and \(\delta(a_t,v_k) = 1\) if \(a_t = v_k\), and zero otherwise. These refined re-estimation formulae can avoid evaluating \(\gamma_{q,T}(m)\) (or \(\alpha_{q,T}(m,d)\)), which requires a lot of memory units and multiplications in performing (10), or the subtractions in (15) and (16), as we mentioned in the last section.

In some applications, a parametric distribution may be required or preferred. The number of model parameters can be reduced using parametric distributions such as Gaussian, Poisson and gamma distributions [20]. Then instead of estimating \(b_{m}(v_k)\) and \(p_{m}(d)\) for each \(m, k,\) and \(d,\) one only needs to estimate a few parameters that specify the selected distribution functions. Ferguson [1] showed that the new parameters for the duration distribution \(p_{m}(d)\) and the observation distribution \(b_{m}(v_k)\) for a given state \(s_m\) can be found by maximizing \(\sum d \hat{p}_{m}(d) \log p_{m}(d) + \sum v \hat{b}_{m}(v_k) \log b_{m}(v_k)\) subject to the constraints \(\sum p_{m}(d) = 1\) and \(\sum \hat{b}_{m}(v_k) = 1\), where \(\hat{p}_{m}(d)\) is given by (26), and \(\hat{b}_{m}(v_k)\) by (28) or (29). For example, if the probability distribution \(b_{m}(v_k)\) is Poisson with mean \(\mu_{m}\), i.e., \(b_{m}(k) = \mu_{m}^k e^{-\mu_{m}} / k!\), then the parameter \(\mu_{m}\) can be re-estimated by \(\hat{\mu}_{m} = \sum \hat{b}_{m}(v_k) k\), or equivalently
\[
\hat{\mu}_{m} = \frac{\sum \gamma_{q,T}(m) \alpha_t}{N_k} = \frac{\sum \sum \gamma_{q,T}(m) \delta(a,v_k)}{N_k} \quad (30)
\]
where \(\sum \alpha_{t+i} \delta(a,v_k)\) is the number of arrivals within \(d\) seconds. It is straightforward to recursively calculate the above \(\hat{\mu}_{m}\) by using the relation \(\sum \alpha_{t+i} \delta(a,v_k) = \alpha_t + \sum \delta(a_{t+i} - 1) \delta(a_{t+i} + 1)\). For Gaussian, gamma and other distributions that belong to the exponential family, the maximum-likelihood parameters can be similarly obtained [20]. 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 [23].

The MAP estimation of states and the maximum-likelihood re-estimation of model parameters can be combined with the backward algorithm. Therefore, the backward variables \(\gamma_{q,T}(m), D_{q,T}(m,d),\) and \(\gamma_{q,T}(m)\) do not have to be stored for later use. Among the forward variables, only \(\xi_{q}(m)\) and \(S_{q}(m)\) (for all \(m\) and \(t\) need to be stored, since they are used in the parameter re-estimation. Hence, the storage requirement is \(O(MT)\).

Obviously, the number of computation steps required for evaluating \(\gamma_{q,T}(m), D_{q,T}(m,d)\) and \(D_{q,T}(m,d)\) are linearly proportional to the number of parameters. Hence, the computational complexity of the re-estimation algorithm is \(O(\lambda|T|)\), where \(\lambda = M^2 + M + M + D\) is the total number of model parameters. Recall that the integer \(K\) is the number of distinct values that the observation \(a_t\) can take on, i.e., the cardinality of the set \(\{v_k\}\).

VI. CONCLUSION

The redefined FB algorithm for the explicit-duration HMM can avoid the underflow problem in practical applications, without an increase in computational complexity compared with the most efficient FB algorithm proposed in [17]. The formulations of the various predicted, filtered, smoothed probabilities and the FB variables provide multiple choices to meet specific needs of an application. The general and symmetrical forms of the FB algorithm for the explicit-duration HMM provide different performance choices for its processing speed, memory usage, and implementation in software or hardware. Based on the specific preference of the application, re-estimation of the model parameters can be done using either nonparametric or parametric distributions. To validate the algorithm, we have implemented it in FPGA and applied it to an intrusion detection system to defend against Distributed Denial of Service (DDoS) attacks, which we do not present here in the interest of space but will be reported elsewhere.

REFERENCES


