Hidden semi-Markov models (HSMMs)

Kevin P. Murphy
www.ai.mit.edu/∼murphyk

20 November 2002

Abstract

A semi-Markov HMM is like an HMM except each state can emit a sequence of observations. In this paper, we provide a unified review of inference and learning in a variety of different kinds of semi-Markov HMMs, including explicit-duration HMMs, segment HMMs, and multigrams. We also discuss some important implementation details, such as avoiding underflow, which are often glossed over.

1 Segment models

1.1 Representation

A semi-Markov HMM (more properly called a hidden semi-Markov model, or HSMM) is like an HMM except each state can emit a sequence of observations. Let $Y(G_t)$ be the subsequence emitted by "generalized state" G_t. The "generalized state" usually contains both the automaton state, Q_t, and the length (duration) of the segment, L_t. We will define $Y(G_t)$ to be the subsequence $y_{t-l+1:t}$. After emitting a segment, the next state is G_{t_n}, where $t_n = t + L_t$. Similarly, denote the previous state by G_{t_p}. Let $Y(G_t^+)$ be all observations following G_t, and $Y(G_t^-)$ be all observations preceeding G_t, as in Figure 1.

Each segment $O_t(q,l) \stackrel{\text{def}}{=} P(Y(G_t)|Q_t = q, L_t = l)$ can be an arbitrary distribution. If $P(Y(G_t)|q,l) = \prod_{i=t-l+1}^{t} P(y_i|q)$, this is an explicit duration HMM [Fer80, Lev86, Rab89, MJ93, MHJ95]. If $P(Y(G_t)|q,l)$ is modelled by an HMM or state-space model (linear-dynamical system), this is called a segment model [GY93, ODK96]. In computational biology, $P(Y(G_t)|q,l)$ is often modelled by a weight matrix or higher-order Markov chain (see e.g., [BK97]). In this paper, we are agnostic about the form of $P(Y(G_t)|q,l)$.

It is possible to approximate a variable-duration HMM by adding extra states to a regular HMM (see [DEKM98, p69]), i.e., a mixture of geometric distributions. However, our main interest will be segment models, which are strict generalizations of variable-duration HMMs.

For the relationship between semi-Markov HMMs, pseudo-2D HMMs, hierarchical HMMs, etc., please see [Mur02]. (Essentially, with a pseudo-2D HMMs, we know the size of each segment ahead of time; an HHMM is a generalization of a segment model where each segment can have subsegments inside of it, each modelled by an HMM.)

We can represent a variable-duration HMMs as a DBN as shown in FIgure 2. We explicitly add Q_t^D, the remaining duration of state Q_t, to the state-space. (Even though Q_t is constant for a long period, we copy its value across every

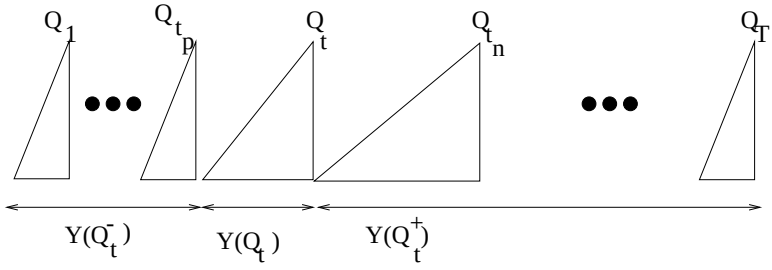

Figure 1: A segment model.

time slice, to ensure a regular structure.) When we first enter state i, Q_t^D is set to a value from from $q_i(\cdot)$; it then deterministically counts down to 0. When $Q_t^D = 0$, the state is free to change, and Q_t^D is set to the duration of the new state. We can encode this behavior by defining the CPDs as follows:

$$P(Q_t = j | Q_{t-1} = i, Q_{t-1}^D = d) = \begin{cases} \delta(i,j) & \text{if } d > 0 \text{ (remain in same state)} \\ A(i,j) & \text{if } d = 0 \text{ (transition)} \end{cases}$$

$$P(Q_t^D = d' | Q_{t-1}^D = d, Q_t = k) = \begin{cases} p_k(d') & \text{if } d = 0 \text{ (reset)} \\ \delta(d', d-1) & \text{if } d > 0 \text{ (decrement)} \end{cases}$$

Since we have expanded the state space, inference in a variable-duration HMM is slower than in a regular HMM. The naive approach to inference in this DBN takes $O(TD^2Q^2)$ time, where D is the maximum number of steps we can spend in any state, and Q is the number of HMM states. However, we can exploit the fact that the CPD for Q^D is deterministic to reduce this to $O(TDQ^2)$.

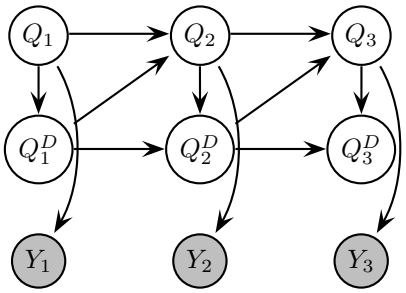

Figure 2: A variable-duration HMM. Q_t represents the state, and Q_t^D represents how long we have been in that state (duration).

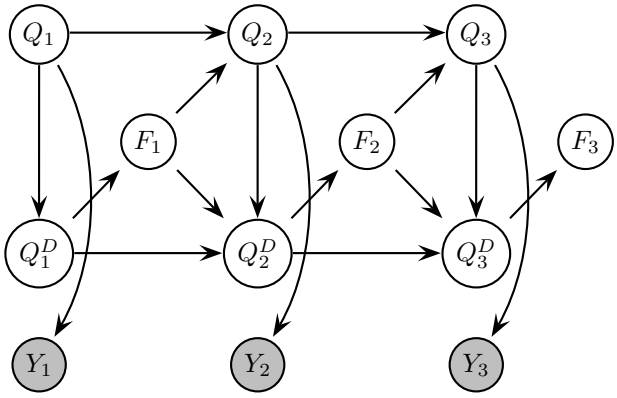

Figure 3: A variable-duration HMM with explicit finish nodes. Q_t represents the state, and Q_t^D represents how long we have been in that state (duration), and F_t is a binary indicator variable that turns to indicate that Q_t^D has finished.

To facilitate future generalizations, we introduce deterministic "finish" nodes, that "turn on" when the duration counter reaches 0. This is a signal that Q_t can change state, and that a new duration should be chosen. See Figure 3. We define the CPDs as follows:

$$
\begin{aligned}
P(Q_t = j | Q_{t-1} = i, F_{t-1} = f) &= \begin{cases} \delta(i,j) & \text{if } f = 0 \text{ (remain in same state)} \\ A(i,j) & \text{if } f = 1 \text{ (transition)} \end{cases} \\
P(Q_t^D = d' | Q_{t-1}^D = d, Q_t = k, F_{t-1} = 1) &= p_k(d') \\
P(Q_t^D = d' | Q_{t-1}^D = d, Q_t = k, F_{t-1} = 0) &= \begin{cases} \delta(d', d-1) & \text{if } d > 0 \\ \text{undefined} & \text{if } d = 0 \end{cases} \\
P(F_t = 1 | Q_t^D = d) &= \delta(d, 0)
\end{aligned}
$$

Note that $P(Q_t^D = d' | Q_{t-1}^D = d, Q_t = k, F_{t-1} = 0)$ is undefined if $d = 0$, since if $Q_{t-1}^D = 0$, then $F_{t-1} = 1$, by construction.

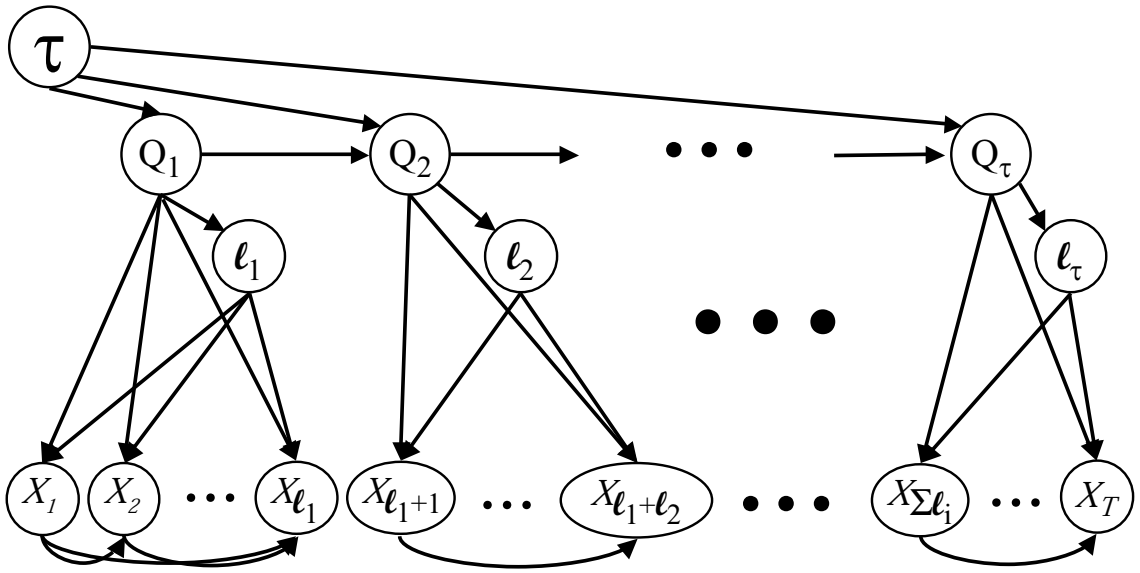

Figure 4: A schematic depiction of a segment model. The X_t nodes are observed, the rest are hidden. The X_t's within a segment need not be fully connected. Also, there may be dependencies between the observables in adjacent segments (not shown). This is not a valid DBN since the l_i's are random variables, and hence the structure is not fixed. Thanks to Jeff Bilmes for this Figure.

The basic idea of a segment model is that each HMM state can generate a sequence of observations, as in Figure 1, instead of just a single observation, The difference from a variable-duration HMM is that we do not assume the observations within each segment are conditionally independent; instead we can use an arbitrary model for their joint distribution. The likelihood is given by

$$
P(y_{1:T}) = \sum_{\tau} \sum_{q_{1:\tau}} \sum_{l_{1:\tau}} \prod_{i=1}^{l_i} P(\tau) P(q_i | q_{i-1}, \tau) P(l_i | q_i) P(y_{t_0(i):t_1(i)} | q_i, l_i)
$$

where τ is the number of segments, l_i is the length of the i'th segment (that satisfies the constraint $\sum_{i=1}^{\tau} l_i = T$), and $t_0(i) = \sum_{j=1}^{i-1} l_j + 1$ and $t_1(i) = t_0(i) + l_i - 1$ are the start and ending times of segment i.

A first attempt to represent this as a graphical model is shown in Figure 4. This is not a fully specified graphical model, since the L_i's are random variables, and hence the topology is variable. To specify a segment model as a DBN, we must make some assumptions about the form of $P(y_{t:t+l}|q, l)$. Let us consider a particular segment and renumber so that $t_0 = 1$ and $t_1 = l$. If we assume the observations are conditionally independent,

$$
P(y_{1:l} | Q_t = k, l) = \prod_{t=1}^{l} P(y_t | Q_t = k)
$$

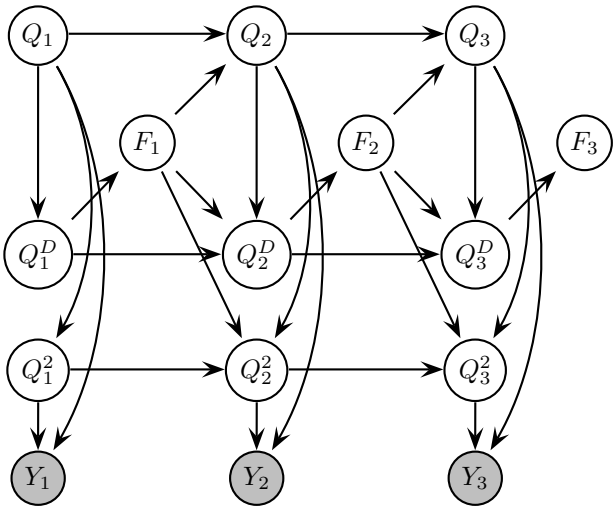

Figure 5: A segment model where each segment is modelled by an HMM.

we recover the variable-duration HMM in Figure 3. (If $p(l|q)$ is a geometric distribution, this becomes a regular HMM.) Note that the number of segments is equal to the number of transitions of the Q_t nodes, or equivalently, the number of times F_t turns on. By considering all possible assignments to Q_t and Q_t^D (and hence to F_t), we consider all possible segmentations of the data.

The next simplest segment model is to model each segment $P(y_{1:l}|Q_t = k, l)$ using an HMM, i.e.,

$$P(y_{1:l}|Q_t = k, l) = \sum_{q_{1:l}} \pi_k(q_1) P(y_1|Q_t = k, Q_1^2 = q_1) \prod_{\tau=2}^{l} A_k(q_{\tau-1}, q_\tau) P(y_t|Q_t = k, Q_\tau^2 = q_\tau)$$

where Q_t^2 represents the state of the HMM within this particular segment. We can model this as a DBN as shown in Figure 5. (Naturally we could allow arcs between the Y_t arcs, as in an autoregressive HMM.) The CPDs for Q_t, Q_t^D and F_t are the same as in Figure 3. The CPD for Y_t gets modified because it must condition on both Q_t and the state within the segment-level HMM, Q_t^2. The CPD for Q_t^2 is as follows:

$$P(Q_t^2 = j|Q_{t-1}^2 = i, Q_t = k, F_{t-1} = f) \quad = \quad \begin{cases} \pi_k^2(j) & \text{if } f = 0 \text{ (reset)} \\ A_k^2(i,j) & \text{if } f = 1 \text{ (transition)} \end{cases}$$

It is straightforward to use other models to define the segment likelihood $P(y_{1:l}|Q_t = k, l)$, e.g., a second-order Markov model, or a state-space model.

1.2 Forwards-backwards

We define the following quantities for generalized state variables, by analogy with a regular HMM:

$$\alpha_t(g) \quad \stackrel{\text{def}}{=} \quad P(G_t = g, Y(G_t), Y(G_t^-))$$
$$\beta_t(g) \quad \stackrel{\text{def}}{=} \quad P(Y(G_t^+)|G_t = g)$$

We can recursively compute α_t as follows. (Equation 9 of [ODK96].)

$$\begin{aligned}
\alpha_t(g) &= \sum_{g'} P(G_t = g, G_{t_p} = g', Y(G_t^-), Y(G_t)) \\
&= \sum_{g'} P(Y(G_t)|G_t = g, \cancel{G_{t_p} = g'}, \cancel{Y(G_t^-)}) P(G_t = g, G_{t_p} = g', Y(G_t^-)) \\
&= \sum_{g'} P(Y(G_t)|G_t = g) P(G_t = g|G_{t_p} = g', \cancel{Y(G_t^-)}) P(G_{t_p} = g', Y(G_t^-)) \\
&= \sum_{g'} P(Y(G_t)|G_t = g) P(G_t = g|G_{t_p} = g') P(G_{t_p} = g', Y(G_t^-)) \\
&= O_t(g) \sum_{g'} P(g|g') \alpha_{t_p}(g')
\end{aligned}$$

Similarly, we can recursively compute β_t as follows. (Equation 10 of [ODK96].)

$$\begin{aligned}
\beta_t(g) &= \sum_{g'} P(Y(G_{t_n}^+), Y(G_{t_n}), G_{t_n} = g'|G_t = g) \\
&= \sum_{g'} P(Y(G_{t_n}^+)|\cancel{Y(G_{t_n})}, G_{t_n} = g', \cancel{G_t = g}) P(Y(G_{t_n}), G_{t_n} = g'|G_t = g) \\
&= \sum_{g'} P(Y(G_{t_n}^+)|G_{t_n} = g') P(Y(G_{t_n})|G_{t_n} = g', \cancel{G_t = g}) P(G_{t_n} = g'|G_t = g) \\
&= \sum_{g'} \beta_{t_n}(g') O_{t_n}(g') P(g'|g)
\end{aligned}$$

These equations reduce to the regular HMM equations if $Y(G_t) = Y_t$, $t_p = t - 1$, etc.

The equations above cannot be implemented, since t_n and t_p are random variables. Hence we will rewrite them explicitely using the fact that $G_t = (Q_t, L_t)$. But first we will introduce an extra variable, that will allow us to be much more precise and concise than most papers that discuss semi-Markov HMMs. Let $F_t = 1$ if there is a segmentation boundary at t (F for finish). A segmentation boundary is usually interpreted to mean $Q_{t+1} \neq Q_t$, although one could imagine leaving state Q_t, and re-entering it at the next time step, resetting the duration. Then

$$\begin{aligned}
\alpha_t(q, l) &\stackrel{\text{def}}{=} P(Q_t = q, L_t = l, F_t = 1, y_{1:t}) \\
&= P(y_{t-l+1:t}|q, l) \sum_{q'} \sum_{l'} P(q, l|q', l') \alpha_{t-l}(q', l')
\end{aligned}$$

and

$$\begin{aligned}
\beta_t(q, l) &= P(y_{t+1:T}|Q_t = q, L_t = l, F_t = 1) \\
&= \sum_{q'} \sum_{l'} \beta_{t+l'}(q', l') P(y_{t+1:t+l'}|q', l') P(q', l'|q, l)
\end{aligned}$$

We can see that the forwards-backwards algorithm has complexity $O(TQ^2L^2)$. But if we make the standard assumption that

$$P(q, l|q', l') = P(q|q')P(l|q)$$

then we can reduce the complexity to $O(TQ^2L)$ as follows.[1]

We start with the backwards equation.

$$\beta_t(i, d') = \sum_{j} \sum_{d} \beta_{t+d}(j, d) P(y_{t+1:t+d}|j, d) P(j|i) P(d|j)$$

[1] [Rab89, p280] incorrectly states that the complexity of inference in a variable-duration HMM (a special case of a segment model) is $O(TQ^2L^2)$. However, by precomputing $P(y_{t-l+1:t}|Q, L)$, and substituting into Equation 68 of [Rab89], it is easy to see that the complexity is just $O(TQ^2L)$, as also pointed out in [MHJ95],

Since the RHS is independent of d', we simplify this to

$$\beta_t(i) = \sum_j P(j|i) \sum_d P(d|j)\beta_{t+d}(j)P(y_{t+1:t+d}|j,d)$$

This matches Equation 70 of [GY93].

$\beta_t(i)$ is the probability of future evidence given that we finish in state i at time t. For parameter estimation, it will be helpful to define a related quantity, $\beta^*_t(i)$, which is the probability of seeing future evidence given that we start in state i at $t+1$. Following [Rab89] we have:

$$\beta_t(i) \stackrel{\text{def}}{=} P(y_{t+1:T}|Q_t = i, F_t = 1) = \sum_j \beta^*_t(j)A(i,j)$$

$$\beta^*_t(i) \stackrel{\text{def}}{=} P(y_{t+1:T}|Q_{t+1} = i, F_t = 1) = \sum_{d=1}^{D} \beta_{t+d}(i)P(d|i)P(y_{t+1:t+d}|i,d)$$

The base case if $\beta_T(i) = 1$.

Now we turn to the forwards equation.

$$\begin{aligned}\alpha_t(j,d) &= P(y_{t-d+1:t}|j,d) \sum_i \sum_{d'} A(i,j)P(d|j)\alpha_{t-d}(i,d') \\ &= P(y_{t-d+1:t}|j,d)P(d|j) \sum_i A(i,j) \left(\sum_{d'} \alpha_{t-d}(i,d') \right)\end{aligned}$$

This matches Equation 69 of [GY93].

There is no need to include the duration in the state space. So if we define

$$\begin{aligned}\alpha_t(j) &\stackrel{\text{def}}{=} P(Q_t = j, F_t = 1, y_{1:t}) \\ &= \sum_d \alpha_t(j,d)\end{aligned}$$

then the above simplifies to

$$\alpha_t(j) = \sum_d P(y_{t-d+1:t}|j,d)P(d|j) \sum_i A(i,j)\alpha_{t-d}(i)$$

which matches Equation 68 of [Rab89]. If we define $\alpha^*_t(i)$ to be the probability of starting in i at $t+1$ (The correspondence with [GY93] is as follows: $\alpha_t(j,0) = \alpha^*_{t-1}(j)$.), then

$$\alpha_t(j) \stackrel{\text{def}}{=} P(y_{1:t}, Q_t = j, F_t = 1) = \sum_d P(y_{t-d+1:t}|j,d)P(d|j)\alpha^*_{t-d}(j)$$

$$\alpha^*_t(j) \stackrel{\text{def}}{=} P(y_{1:t}, Q_{t+1} = j, F_t = 1) = \sum_i \alpha_t(i)A(i,j)$$

The base case if $\alpha^*_0(j) = \pi(j)$, the probability of starting in state j.

1.3 Expected sufficient statistics

To compute the transition probability $P(Q_t|Q_{t_p}) = P(Q_t|Q_{t-1}, F_{t-1} = 1)$, we must count transitions that occur across segment boundaries, not just across neighboring time points. [Rab89, p280] shows how to do this. The maximum likelihood estimate of the transition matrix is

$$\hat{A}_{ij} \propto \sum_{t=1}^{T-1} \alpha_t(i)A_{ij}\beta^*_t(j)$$

The normalization constant is whatever is needed to make A_{ij} sum to one for each i, i.e., to make it a stochastic matrix:

$$\hat{A}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i) A_{ij} \beta_t^*(j)}{\sum_{j'} \sum_{t=1}^{T} \alpha_t(i) A_{ij'} \beta_t^*(j')}$$

In future, we will omit such normalization constants. The initial state distribution is

$$\hat{\pi}_i \propto P(Q_1 = i|y_{1:T}) \propto P(Q_1 = i)P(y_{1:T}|Q_1 = i, F_0 = 1) = \pi_i \beta_0^*(i)$$

We can estimate $P(Y_t = k|Q_t = i)$ for a discrete output variable-duration HMM as follows.

$$\hat{B}_{i,k} \propto \sum_{t:Y_t=k} P(Q_t = i|y_{1:T})$$

We can compute the numerator as follows (summing over all t for simplicity).

$$\sum_{t=1}^{T} P(Q_t = i|y_{1:T}) = \sum_t \sum_{\tau<t} [\gamma_\tau^*(i) - \gamma_\tau(i)]$$

where

$$\begin{aligned}
\gamma_t(i) &\stackrel{\text{def}}{=} P(Q_t = i, F_t = 1|y_{1:T}) \\
&\propto \alpha_t(i)\beta_t(i) \\
\gamma_t^*(i) &\stackrel{\text{def}}{=} P(Q_{t+1} = i, F_t = 1|y_{1:T}) \\
&\propto \alpha_t^*(i)\beta_t^*(i)
\end{aligned}$$

The reason this works is best explained by example. Consider a 2-state system; let us compute the expected amount of time spent in state 1, where the posterior probability of being in this state is shown in the second line below.

t	1	2	3	4	5	6	7	8	9
$P(Q_t = 1\|y_{1:T})$	0	1	1	1	0	0	1	1	0
$\gamma_t^*(1)$	1	0	0	0	0	1	0	0	0
$\gamma_t(1)$	0	0	0	1	0	0	0	1	0
$\gamma_t^*(1) - \gamma_t(1)$	1	0	0	-1	0	1	0	-1	0
$\sum_{\tau<t} \gamma_\tau^*(1) - \gamma_\tau(1)$	0	1	1	1	0	0	1	1	0

Hence $\sum_t \sum_{\tau<t} [\gamma_\tau^*(i) - \gamma_\tau(i)] = 3 + 2 = 5$.

Contrast this with the following, which counts the number of times we enter state i (2 in this example), as opposed to how long we spend in state i.

$$\begin{aligned}
\sum_{t=1}^{T} P(Q_t = i, y_{1:T}) &= \sum_t \left[\sum_d \alpha_{t-d}^*(i) P(d|i) P(y_{t-d+1:t}|i, d) \right] \beta_t(i) \\
&= \sum_t \left[\sum_d \alpha_t(i, d) \right] \beta_t(i) \\
&= \sum_t \alpha_t(i)\beta_t(i) \\
&= \sum_t P(Q_t = i, F_t = 1|y_{1:T})
\end{aligned}$$

We can estimate $P(d|i)$ for a non-parametric duration density [Fer80].

$$\hat{P}(d|i) \propto \sum_t P(Q_t = i, D_t = d, F_t = 1|y_{1:T}) = \sum_t \alpha_{t-d}^*(i) P(d|i) P(y_{t-d+1:t}|i, d) \beta_t(i)$$

Of course, a tabular representation of $P(d|i)$ might have too many parameters. [Lev86] shows how to fit a gamma distribution (the M step requires numerical techniques). [MHJ95] show how to fit $P(d|i)$ for any member of the exponential family.

1.4 Avoiding numerical underflow

As in an HMM, $\alpha_t(q) = P(Q_t = q, y_{1:t})$ rapidly underflows as t gets large. There are two standard approaches: use scaling or logs Scaling is much faster, but as we will see, not applicable to HSMMs. Furthemore, [DEKM98, p78] says scaling may not be enough to avoid underflow if there are many silent states.

1.4.1 Scaling in HMMs

For scaling, we compute $\hat{\alpha}_t(q) \stackrel{\text{def}}{=} P(Q_t = q|y_{1:t})$ as follows:

$$\begin{aligned}
\hat{\alpha}_t(q) &= \frac{P(y_t|Q_t = q, \cancel{y_{1:t-1}})P(Q_t = q|y_{1:t-1})}{P(y_t|y_{1:t-1})} \\
&= \frac{1}{c_t}P(y_t|Q_t = q)\sum_{q'} P(Q_t = q, Q_{t-1} = q'|y_{1:t-1}) \\
&= \frac{1}{c_t}P(y_t|Q_t = q)\sum_{q'} P(Q_t = q|Q_{t-1} = q', \cancel{y_{1:t-1}})P(Q_{t-1} = q'|y_{1:t-1}) \\
&= \frac{1}{c_t}O_t(q)\sum_{q'} P(q|q')\hat{\alpha}_{t-1}(q')
\end{aligned}$$

where

$$c_t \stackrel{\text{def}}{=} P(y_t|y_{1:t-1}) = \sum_q O_t(q)\sum_{q'} A(q', q)\hat{\alpha}_{t-1}(q')$$

Hence, by the chain rule,

$$\log P(y_{1:T}) = \log P(y_1)P(y_2|y_1)P(y_3|y_{1:2})\ldots = \sum_{t=1}^{T}\log c_t$$

Similarly, we will normalise β_t by dividing by $d_t = \sum_q \hat{\beta}_t(q)$ at each step. [Rab89] suggests dividing by c_t, but it does not matter which scale factor we use, since the normalizing constants will cancel when we normalize the posterior:

$$\begin{aligned}
\gamma_t(q) &= \frac{1}{P(y_{1:T})}P(Q_t = q, y_{1:T}) \\
&= \frac{1}{P(y_{1:T})}P(y_{t+1:T}|Q_t, \cancel{y_{1:t}})P(Q_t = q, y_{1:t}) \\
&= \frac{1}{P(y_{1:T})}(D_t\hat{\beta}_t(q))(C_t\hat{\alpha}_t(q)) \\
&= \frac{1}{Z_t}\hat{\beta}_t(q)\hat{\alpha}_t(q)
\end{aligned}$$

where $C_t = \prod_{i=1}^{t} c_t = P(y_{1:t})$, $D_t = \prod_{i=t+1}^{T} d_t$ and $Z_t = \sum_q \hat{\beta}_t(q)\hat{\alpha}_t(q)$.

1.4.2 Scaling in HSMMs

It is not clear how to apply the same scaling tricks to the semi-Markov case. To see the problem, consider

$$\begin{aligned}
\tilde{\alpha}_t(j, d) &\stackrel{\text{def}}{=} P(Q_t = j, D_t = d|y_{1:t-d}, y_{t-d+1:t}) \\
&= \frac{P(y_{t-d+1:t}|Q_t = j, D_t = d)P(Q_t = j, D_t = d|y_{1:t-d})}{P(y_{t-d+1:t}|y_{1:t-d})} \\
&= \frac{1}{c_{t,d}}B_{t,j,d}P(D_t = d|Q_t = j)\sum_i A_{i,j}\tilde{\alpha}_{t-d}(i)
\end{aligned}$$

where

$$c_{t,d} \stackrel{\text{def}}{=} P(y_{t-d+1:t}|y_{1:t-d}) = \sum_j \sum_{d=1}^{D} B_{t,j,d} P(D_t = d|Q_t = j) \sum_i A_{i,j} P(Q_{t-d} = i|y_{1:t-d})$$

and

$$\tilde{\alpha}_t(i) \stackrel{\text{def}}{=} \sum_d \tilde{\alpha}_t(i,d)$$

But this definition of $c_{t,d}$ is nonsensical: d is a free variable on the LHS, but is a bound dummy variable on the RHS (it is summed out). Indeed, experiments show this method doesn't work.

The basic problem is we know how to compute $P(Q_t = j, D_t = d, F_t = 1, y_{1:t})$, but not $P(Q_t = j, D_t = d, F_t = 0, y_{1:t})$, since if a segmentation boundary did not occur, we cannot compute the likelihood term for a partial segment. However, if the model defining the segment is first-order Markov (e.g., in a variable-duration HMM or when each segment is modelled by an HMM), then $F_t = 1$ means the observation comes from a new state, otherwise the old state. Hence scaling can be done, since the denominator is always of the form $P(y_t|y_{1:t-1})$, as in an HMM.

[Lev86, Sec.5] discusses how to scaling for the variable-duration HMM case, although it is not clear that it is correct. The easiest way to see how to do scaling is to consider the DBN in Figure 3. We have

$$P(Q_t = j, D_t = d, F_t = f|y_{1:t}) = \frac{P(y_t|Q_t = j, D_t = d, F_t = f, y_{1:t-1}) P(Q_t = j, D_t = d, F_t = f, y_{1:t-1})}{P(y_t|y_{1:t-1})}$$

In the variable-duration HMM case, $F_t = 1$ iff $D_t = 0$ (no duration left), so we can omit F_t, as shown in Figure 2; the sum over j and d is easy. In the hierarchical HMM case, we do not have D nodes, but $F_t = 1$ iff Q_t enters an end state. Hence we can compute the probability of $P(F_t = 1|y_{1:t})$ and $P(F_t = 0|y_{1:t})$, so we can sum over f and j.

1.4.3 Logs

For logs, define $\tilde{\alpha}_t(q) \stackrel{\text{def}}{=} \log \alpha_t(q)$. Then we use the following fact: to compute $\tilde{r} = \log(p + q)$ from $\tilde{p} = \log p$, $\tilde{q} = \log q$, we use

$$\begin{aligned}
\tilde{r} &= \log(e^{\tilde{p}} + e^{\tilde{q}}) \\
&= \log(e^{\tilde{p}}(1 + e^{\tilde{q}-\tilde{p}})) \\
&= \tilde{p} + \log(1 + e^{\tilde{q}-\tilde{p}}) \\
&= \tilde{p} + \log(e^{\tilde{p}-\tilde{p}} + e^{\tilde{q}-\tilde{p}})
\end{aligned}$$

We pull out the larger of $\tilde{p}$ and $\tilde{q}$; if $\tilde{p} > \tilde{q}$, then $\exp(\tilde{q} - \tilde{p})$ is very small, so we can use a small x approximation for $\log(1 + x)$.

1.4.4 Logs in HMMs

For example, the forwards equation for HMMs

$$\alpha_t(j) = B_t(j) \sum_i \alpha_{t-1}(i) A_{i,j}$$

becomes

$$\begin{aligned}
\tilde{\alpha}_t(j) &= \log B_{t,j} + \log \sum_i \alpha_{t-1}(i) A_{i,j} \\
&= \tilde{B}_t(j) + \log \sum \exp_i \log(\alpha_{t-1}(i) A_{i,j}) \\
&= \tilde{B}_t(j) + \text{logsumexp}_i(\tilde{\alpha}_{t-1}(i) + \tilde{A}_{i,j})
\end{aligned}$$

where $\text{logsumexp}_i c_i$ is a new primitive operator. (Note that if $P(Y_t|Q_t)$ is a Gaussian, we do not need to compute the $\exp$ term since we use $\tilde{B}_t(j)$.)

1.4.5 Logs in HSMMs

For HSMMs, the forward equation

$$\alpha_{t,j} = \sum_d B_{t,j,d} D_{j,d} \alpha^*_{t-d,j}$$

where

$$\alpha^*_{t,j} = \sum_i A_{i,j} \alpha_{t,i}$$

becomes

$$\tilde{\alpha}_{t,j} = \text{logsumexp}_d(\tilde{B}_{t,j,d} + \tilde{D}_{j,d} + \tilde{\alpha}^*_{t-d,j})$$

where

$$\tilde{\alpha}^*_{t,j} = \text{logsumexp}_i(\tilde{A}_{i,j} + \tilde{\alpha}_{t,i})$$

Similarly, the backward equation

$$\beta_{t,i} = \sum_j \beta^*_{t,j} A_{i,j}$$

where

$$\beta^*_{t,i} = \sum_d \beta_{t+d,j} B_{t+d,j,d}$$

becomes

$$\tilde{\beta}_{t,i} = \text{logsumexp}_j j \tilde{\beta}^*_{t,j} + \tilde{A}_{i,j}$$

where

$$\tilde{\beta}^*_{t,i} = \text{logsumexp}_d \tilde{\beta}_{t+d,j} + \tilde{B}_{t+d,j,d}$$

2 Multigrams

2.1 Representation

A multigram [DB95, DB97] is a special case of a segment model in which (1) the segments are independent, so the transition matrix is uniform and (2) each segment is a deterministic string. We will call the segment strings "words", and the collection of all such words a "lexicon"; note, however, that the words might in fact correspond to single symbols or whole phrases.

The likelihood of a sentence is given by

$$\alpha_t \stackrel{\text{def}}{=} P(y_{1:t}, F_t = 1) \stackrel{\text{def}}{=} \sum_n P(n) \sum_{w_{1:n} \stackrel{:}{=} y_{1:t}} \prod_{i=1}^n P(w_i)$$

Let us compare an n-gram with an n-multigram, which is a multigram whose lexicon is all strings of length 1 to n. Consider $n = 3$, alphabet $\{a, b, c, d\}$ and sequence $abcd$. The 3-gram gives it probability

$$P(abcd) = P(a)P(b|a)P(c|ab)P(d|bc)$$

whereas the 3-multigram gives it probability

$$\begin{aligned} P(abcd) &= P(a)P(bcd) + P(a)P(bc)P(d) + P(a)P(b)P(cd) + P(a)P(b)P(c)P(d) \\ &\quad + P(ab)P(cd) + P(ab)P(c)P(d) + P(abc)P(d) \end{aligned}$$

To keep the number of parameters tractable, not all possible strings need appear in the lexicon. This is a model selection (structure learning) issue.

A segment model also sums over segmentations, but uses an HMM to compute the probability of each segment (rather than storing this as a parameter):

$$\begin{aligned} P(abcd) &= \sum_{q_1,q_2} P(q_1)P(l_1=1|q_1)P(a|q_1) \\ &\quad \times P(q_2|q_1)P(l_2=3|q_2)P(bcd|q_2) \\ &+ \sum_{q_1,q_2,q_3} P(q_1)P(l_1=1|q_1)P(a|q_1) \\ &\quad \times P(q_2|q_1)P(l_2=2|q_2)P(bc|q_2) \\ &\quad \times P(q_3|q_2)P(l_3=1|q_3)P(d|q_3) \\ &+ \cdots \end{aligned}$$

2.2 Inference

In the multigram case, the equations simplify as follows.

$$\begin{aligned} \alpha_t(w) &\stackrel{\text{def}}{=} P(Q_t = w, F_t = 1, y_{1:t}) \\ &= P(y_{t-|w|+1:t}|w) \sum_{w'} P(w|w')\alpha_{t-|w|}(w') \\ &= P(y_{t-|w|+1:t}|w) \sum_{w'} \alpha_{t-|w|}(w') \\ &= O_t(w)\alpha_{t-|w|} \end{aligned}$$

where $P(w|w') \stackrel{\text{def}}{=} 1$, $O_t(w) \stackrel{\text{def}}{=} P(w)\delta(w, y_{t-|w|+1:t})$, and $\alpha_t \stackrel{\text{def}}{=} P(y_{1:t}, F_t = 1) = \sum_w \alpha_t(w)$. Hence

$$\begin{aligned} \alpha_t &= \sum_{w:w=y_{t-|w|+1:t}} P(w)\alpha_{t-|w|} \\ &= \sum_{l=1}^{\min\{W,t\}} \alpha_{t-l} \sum_{w:w=y_{t-l+1:t}} P(w) \end{aligned}$$

where W is the length of the longest word, and $\alpha_0 \stackrel{\text{def}}{=} 1$. This matches [dM96, p80]. Typically there will 0 or 1 words which match any given subsequence. Hence

$$\alpha_t = \sum_{l=1}^{\min\{W,t\}} \alpha_{t-l} P(y_{t-l+1:t})$$

Since there is no state, $\alpha_t = P(y_{1:t}, F_t = 1)$. It is not clear how to scale this, since it is a joint probability, and it seems hard to compute $P(y_{1:t}, F_t = 0)$. Hence we use logs:

$$\begin{aligned} \log \alpha_t &= \log \sum_l \exp(\log \alpha_{t-l} + \log P(y_{t-l+1:t})) \\ &= \text{logsumexp}_l(\tilde{\alpha}_{t-l} + \tilde{P}(y_{t-l+1:t})) \end{aligned}$$

In the backwards pass,

$$\begin{aligned} \beta_t(w) &= \sum_{w'} \beta_{t+|w'|}(w')P(y_{t+1:t+|w'|}|w')P(w'|w) \\ &= \sum_{w'} \beta_{t+|w'|}(w')O_{t+|w'|}(w') \end{aligned}$$

Note that $\beta_t(w)$ is independent of w, so we will write it as β_t; this is the probability of generating $y_{t+1:T}$ given that there is a segmentation boundary at t. Hence

$$\begin{aligned}
\beta_t &= \sum_w \beta_{t+|w|} O_{t+|w|}(w) \\
&= \sum_{w:w=y_{t+1:t+|w|}} \beta_{t+|w|} P(w) \\
&= \sum_{l=1}^{\min\{W,T-t\}} \beta_{t+l} P(y_{t+1:t+l})
\end{aligned}$$

In the log domain, this becomes

$$\log \beta_t = \text{logsumexp}_l(\tilde{\beta}_{t+l} + \tilde{P}(y_{t+1:t+l}))$$

The posterior is given by

$$\begin{aligned}
P(a \xrightarrow{w} b|y_{1:T}) &= P(Q_b = w, F_t = 1|y_{1:T}) \\
&= \frac{1}{P(y_{1:T})} \alpha_b(w) \beta_b(w) \\
&= \frac{1}{P(y_{1:T})} P(y_{1:a}) P(y_{a+1:b}|w) P(y_{b+1:T}) \\
&= \frac{1}{P(y_{1:T})} \alpha_a p(w) \beta_b \times \delta(w, y_{a:b})
\end{aligned}$$

where $a = b - |w|$ and $P(y_{1:T}) = \alpha_T$, which matches Equation 5.4 of [dM96, p.80]. In the log domain, if $w = y_{a:b}$, this becomes

$$\log P(a \xrightarrow{w} b|y_{1:T}) = \tilde{\alpha}_a + \tilde{p}(w) + \tilde{\beta}_b - \tilde{\alpha}_T$$

The two-slice posterior, assuming $w_1 = y_{a:b}$ and $w_2 = y_{b+1:c}$, is given by

$$P(a \xrightarrow{w_1,w_2} c|y_{1:T}) = \frac{1}{P(y_{1:T})} \alpha_a P(w_1) P(w_2) \beta_c$$

This matches Equation 5.7 of [dM96, p.82]. In the log domain, this becomes

$$\log P(a \xrightarrow{w_1,w_2} c|y_{1:T}) = \tilde{\alpha}_a + \tilde{P}(w_1) + \tilde{P}(w_2) + \tilde{\beta}_c - \tilde{\alpha}_T$$

References

[BK97] C. Burge and S. Karlin. Prediction of Complete Gene Structures in Human Genomic DNA. *J. Mol. Biol.*, (268):78–94, 1997.

[DB95] S. Deligne and F. Bimbot. Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams. In *Proc. ICASSP '95*, pages 169–172, Detroit, MI, 1995.

[DB97] S. Deligne and F. Bimbot. Inference of variable-length acoustic units for continuous speech recognition. In *Proc. ICASSP '97*, pages 1731–1734, Munich, Germany, 1997.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1998.

[dM96] C. de Marcken. *Unsupervised language acquisition*. PhD thesis, MIT AI lab, 1996.

[Fer80] J. D. Ferguson. Variable duration models for speech. In *Proc. Symp. on the Application of HMMs to Text and Speech*, pages 143–179, 1980.

[GY93] M.J.F. Gales and S.J. Young. The Theory of Segmental Hidden Markov Models. Technical Report CUED/F-INFENG/TR.133, Cambridge Univ. Eng. Dept., 1993.

[Lev86] S. E. Levinson. Coninuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):29–45, 1986.

[MHJ95] C. D. Mitchell, M. P. Harper, and L. H. Jamieson. On the Complexity of Explicit Duration HMMs. *IEEE Transactions on Speech and Audio Processing*, 3(3), May 1995.

[MJ93] C. D. Mitchell and L. H. Jamieson. Modeling duration in a hidden Markov model with the exponential family. In *Intl. Conf. on Acoustics, Speech and Signal Proc.*, pages 331–334, 1993.

[Mur02] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Dept. Computer Science, UC Berkeley, 2002.

[ODK96] M. Ostendorf, V. Digalakis, and O. Kimball. From HMMs to segment models: a unified view of stochastic modeling for speech recognition. *IEEE Trans. on Speech and Audio Processing*, 4(5):360–378, 1996.

[Rab89] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.