Hidden Markov Models

Namrata Vaswani, Iowa State University

April 24, 2014

1 Hidden Markov Model Definitions and Examples

Definitions:

1. A hidden Markov model (HMM) refers to a set of "hidden" states $X_0, X_1, \dots, X_t, \dots, X_T$ and a set of observations, $Y_1, \dots, Y_t, \dots, Y_T$ with the following joint PMF or PDF:
$$p(x_{0:T}, y_{1:T}) = [p(x_0)[\prod_{\tau=1}^{T} p(x_\tau|x_{\tau-1})]][[\prod_{\tau=1}^{T} p(y_\tau|x_\tau)]] \tag{1}$$

2. The sequence is an HMM if and only if

   (a) given X_t, X_{t+1} is independent of $X_{0:t-1}$ (past-X) and

   (b) given X_t, Y_t is independent of $X_{0:t-1}, X_{t+1:T}$ (all-X) and $Y_{0:t-1}$ (past-Y)

   This follows by writing out the expression for $p(x_{0:t}, y_{0:t})$ using chain rule and then using (1) and comparing coefficients. By chain rule,
$$p(x_{0:T}, y_{1:T}) = p(x_0) \prod_{\tau=1}^{T} p(x_\tau|x_{\tau-1}, x_{0:\tau-2}) \prod_{\tau=1}^{T} p(y_\tau|x_{0:T}, y_{1:\tau-1}) \tag{2}$$

   Compare this with (1). In both equations integrate over $y_{1:T}$ and cancel out $p(x_1|x_0)$ to get
$$\prod_{\tau=2}^{T} p(x_\tau|x_{\tau-1}, x_{0:\tau-2}) = \prod_{\tau=2}^{T} p(x_\tau|x_{\tau-1}).$$

   Now integrate also over $x_{3:T}$ on both sides to get $p(x_2|x_1, x_0) = p(x_2|x_1)$. Next, integrate over only $x_{4:T}$ and use this to conclude that $p(x_2|x_1, x_0)p(x_3|x_2, x_1, x_0) = p(x_2|x_1)p(x_3|x_2)$ and so $p(x_3|x_2, x_1, x_0) = p(x_3|x_2)$. Proceed in a similar fashion to conclude that $p(x_t|x_{0:t-1}) = p(x_t|x_{t-1})$ for each t, i.e. item (a) holds.

   At the end of the above, we conclude that
$$\prod_{\tau=1}^{T} p(y_\tau|x_{0:T}, y_{1:\tau-1}) = \prod_{\tau=1}^{T} p(y_\tau|x_\tau).$$

Integrate over $y_{2:T}$ to conclude that $p(y_1|x_{0:T}) = p(y_1|x_1)$. Use this and integrate over only $y_{3:T}$ to conclude that $p(y_2|x_{0:T}, y_1) = p(y_2|x_2)$. Proceed in a similar fashion to conclude that $p(y_t|x_{0:T}, y_{1:t-1}) = p(y_t|x_t)$ for each t, i.e. item (b) holds.

3. The sequence is an HMM if and only if

   (a) given X_t, X_{t+1} is independent of $X_{0:t-1}$ (past-X) and $Y_{0:t}$ (past-Y)

   (b) given X_t, Y_t is independent of $X_{0:t-1}$ (past-X) and $Y_{0:t-1}$ (past-Y)

   This also follows by writing out the expression for $p(x_{0:t}, y_{0:t})$ using chain rule and then using (1) and comparing coefficients.

   Either of the above can also be concluded by using results from the Graphical Models handout.

The following can be shown either using Theorem 2 of the Graphical Models handout or directly.

1. The joint PMF or PDF of the hidden states given by

$$p(x_{0:T}) = p(x_0) \prod_{\tau=1}^{T} p(x_\tau|x_{\tau-1}) \tag{3}$$

   This follows using (1) and integrating over $y_{1:T}$.

2. Given X_t, $X_{t+1:T}$ are conditionally independent (c.i.) of past-X ($X_{0:t-1}$) and of past-Y ($Y_{0:t}$).

3. Given X_t, $Y_{t:T}$ are c.i. of past-X ($X_{0:t-1}$) and of past-Y ($Y_{0:t-1}$).

4. Given X_{t-k}, $Y_{t:T}$ is c.i. of $Y_{0:t-k}$ and of $X_{0:t-k-1}$ for $k > 0$.

5. given X_t, X_{t+1} is c.i. of past-X ($X_{0:t-1}$) and of past-Y ($Y_{0:t}$), and

6. given X_t, Y_t is c.i. of all-X ($X_{0:t-1}, X_{t+1:T}$) and all-Y ($Y_{0:t-1}, Y_{0:t+1:T}$).

7. By reversing the Markov chain $\{X_t\}$, we can also claim that given X_t, X_{t-1} is c.i. of all future-X ($X_{t+1:T}$) and all future-Y ($Y_{t:T}$).

8. Given X_{t-k}, $Y_{t:T}$ is c.i. of $X_{0:t-k-1}$ and $Y_{0:t-k}$ for $k > 0$. If $k = 0$, replace $Y_{0:t-k}$ by $Y_{0:t-1}$

9. By reversing the Markov chain $\{X_t\}$, the opposite of 3 can also be shown for future.

10. Many more

Let us try to prove item 2. We get

$$\begin{aligned}
p(x_{t+1:T}|x_t, x_{0:t-1}, y_{0:t}) &= p(x_{t+1}|x_t, x_{0:t-1}, y_{0:t})p(x_{t+2:T}|x_{t+1}, x_{0:t}, y_{0:t}) \\
&= p(x_{t+1}|x_t)p(x_{t+2:T}|x_{t+1}, x_{0:t}, y_{0:t}) \\
&= p(x_{t+1}|x_t)p(x_{t+2}|x_{t+1}, x_{0:t}, y_{0:t})p(x_{t+3:T}|x_{t+2}, x_{0:t+1}, y_{0:t}) \\
&= p(x_{t+1}|x_t)p(x_{t+2}|x_{t+1})p(x_{t+3:T}|x_{t+2}, x_{0:t+1}, y_{0:t}) \qquad (4)
\end{aligned}$$

The first equality uses chain rule, the second uses (a) of definition 3, the third uses chain rule. The fourth uses (a) of definition 3 and the following fact with $X \equiv X_{t+2}$, $W \equiv X_{t+1}$, $Z \equiv Y_{0:t}$ and $Y \equiv Y_{t+1}$.

Fact 1 *X independent of $\{Z, Y\}$ implies that X independent of Z. Similarly given W, X c.i. of $\{Z, Y\}$ implies that given W, X c.i. of Z. The proof of this follows by writing $p(x, z, y|w) = p(x|w)p(z, y|w)$ and integrating over y.*

Proceeding in a similar fashion, we finally get

$$p(x_{t+1:T}|x_t, x_{0:t-1}, y_{0:t}) = \prod_{\tau=t+1}^{T} p(x_\tau|x_{\tau-1})$$

Using (a) of Definition 2 and Fact 1, $p(x_{t+1:T}|x_t) = \prod_{\tau=t+1}^{T} p(x_\tau|x_{\tau-1})$ and thus we get

$$p(x_{t+1:T}|x_t, x_{0:t-1}, y_{0:t}) = p(x_{t+1:T}|x_t)$$

i.e. the result follows.

The other conclusions given above can be proved similarly.

HMM Examples.

1. The state space model used for defining the Kalman filter was an example of an HMM with continuous states, X_t and continuous observations, Y_t.

2. X_t refers to today's weather which can take one of three possible values, {rainy, cloudy, sunny}. Y_t is a binary random variable which can take two possible values {class occurs, no class occurs}. It is natural to claim that today's weather depends only on yesterday's weather, i.e. given yesterday's weather, today's weather is c.i. of past weather or of whether class occurred yesterday or in the past or not. Also, the chance that class will occur today or not is governed only by today's weather (if it is sunny, it is more likely that the class will not occur!) and given today's weather, the chance is independent of all past or future weather and also of whether classes occurred in the past or in the future. This, of course models, an irresponsible professor who does not care about whether the material is covered or not!

3. Speech recognition, X_t: different phonems, Y_t: linear prediction coefficients (LPC's) of the AR model describing observed speech.

4. Gesture recognition, X_t: different gestures out of a set, Y_t: outer contour of the observed hand shape (for hand gestures)

5. In last two examples, X_t is discrete, Y_t is continuous, that is allowed too.

Causal Posterior Computation.

1. We refer to $p(x_t|y_{0:t})$ as the causal posterior. In real-time applications, there is a need to compute it recursively, for example, to be able to compute the causal MMSE or causal MAP estimate.

2. "Recursive computation" means use the causal posterior and $t-1$ and the current observation to compute the causal posterior at t.

3. Using Bayes' rule and HMM properties, the causal posterior satisfies

$$\begin{aligned}
p(x_t|y_{0:t}) &\propto p(x_t, y_t|y_{0:t-1}) \\
&= p(x_t|y_{0:t-1})p(y_t|x_t, y_{0:t-1}) \\
&= p(x_t|y_{0:t-1})p(y_t|x_t) \quad \text{(using HMM definition 3)} \\
&= p(y_t|x_t) \int p(x_t, x_{t-1}|y_{0:t-1})dx_{t-1} \\
&= p(y_t|x_t) \int p(x_{t-1}|y_{0:t-1})p(x_t|x_{t-1}, y_{0:t-1})dx_{t-1} \\
&= p(y_t|x_t) \int p(x_{t-1}|y_{0:t-1})p(x_t|x_{t-1})dx_{t-1} \quad \text{(using HMM definition 3)} \qquad (5)
\end{aligned}$$

4. The above recursion is another way to derive the Kalman filter recursion: the causal MMSE estimate, $\mathbb{E}[X_t|Y_{0:t}]$, is the expectation of X_t under the causal posterior. Since everything there is jointly Gaussian, the posteriors will also be Gaussian and hence completely specified by the mean and covariance. Kay's book does it this way.

5. The same rules apply for discrete states: just replace $\int$ by $\sum$.

2 Discrete-state HMM

We study the set of techniques developed for **discrete-state HMM's**. The material is based on Rabiner's tutorial (Proc. IEEE, February 1989).

Thus any X_t is a discrete random variable which takings one of N possible values, $i = 1, 2, \ldots N$. Y_t is either discrete or continuous.

2.1 Notation

A time-homogenous discrete state HMM is completely specified by

$$\begin{aligned}
\pi_i &\triangleq P(X_t = i) \\
a_{i,j} &\triangleq P(X_t = j | X_{t-1} = i) \\
b_j(y) &\triangleq P(Y_t = y | X_t = j) \quad \text{(if } Y_t \text{ is continuous this is replaced by the conditional PDF)}
\end{aligned} \tag{6}$$

The following notation is used in efficient computation of various quantities.

$$\begin{aligned}
\alpha_t(i) &\triangleq p(y_{0:t}, x_t = i) \\
\beta_t(i) &\triangleq p(y_{t+1:T} | x_t = i) \\
\gamma_t(i) &\triangleq p(x_t = i | y_{0:T}) \quad \text{(note this conditions on all observations)} \\
\xi_t(i,j) &\triangleq p(x_t = i, x_{t+1} = j | y_{0:T}) \quad \text{(note this conditions on all observations)}
\end{aligned} \tag{7}$$

2.2 Recursion for $\alpha_t, \beta_t, \gamma_t, \xi_t$

Consider α_t

$$\begin{aligned}
\alpha_t(i) &\triangleq p(y_{0:t}, x_t = i) \\
&= \sum_{j=1}^{N} p(y_{0:t}, x_t = i, x_{t-1} = j) \\
&= \sum_{j=1}^{N} p(y_{0:t-1}, x_{t-1} = j) p(x_t = i | x_{t-1} = j, y_{0:t-1}) p(y_t | x_t = i, x_{t-1} = j, y_{0:t-1}) \\
&= \sum_{j=1}^{N} p(y_{0:t-1}, x_{t-1} = j) p(x_t = i | x_{t-1} = j) p(y_t | x_t = i) \quad \text{(using HMM definition 3)} \\
&= \sum_{j=1}^{N} p(y_{0:t-1}, x_{t-1} = j) a_{ji} b_i(y_t) \\
&= b_i(y_t) \sum_{j=1}^{N} \alpha_{t-1}(j) a_{j,i}
\end{aligned} \tag{8}$$

Consider β_t

$$\begin{aligned}
\beta_t(i) &\triangleq p(y_{t+1:T}|x_t = i) \\
&= \sum_{j=1}^{N} p(y_{t+1:T}, x_{t+1} = j|x_t = i) \\
&= \sum_{j=1}^{N} p(x_{t+1} = j|x_t = i)p(y_{t+1}|x_{t+1} = j, x_t = i)p(y_{t+2:T}|x_{t+1} = j, x_t = i, y_{t+1}) \\
&= \sum_{j=1}^{N} p(x_{t+1} = j|x_t = i)p(y_{t+1}|x_{t+1} = j)p(y_{t+2:T}|x_{t+1} = j) \quad \text{(using HMM definition 3)} \\
&= \sum_{j=1}^{N} a_{i,j} b_j(y_{t+1})\beta_{t+1}(j) \qquad (9)
\end{aligned}$$

Consider γ_t. Using definitions of $\alpha_t(i)$ and $\beta_t(i)$, it is clear that $\gamma_t(i) \propto \alpha_t(i)\beta_t(i)$. Thus,

$$\gamma_t(i) = \frac{1}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \alpha_t(i)\beta_t(i) \qquad (10)$$

Consider ξ_t

$$\begin{aligned}
\xi_t(i,j) &\triangleq p(x_t = i, x_{t+1} = j|y_{0:T}) \\
&= \frac{1}{p(y_{0:T})} p(y_{0:T}, x_t = i, x_{t+1} = j) \\
&= \frac{1}{p(y_{0:T})} p(x_t = i, y_{0:t})p(x_{t+1} = j|x_t = i, y_{0:t})p(y_{t+1:T}|x_{t+1} = j, x_t = i, y_{0:t}) \\
&= \frac{1}{p(y_{0:T})} p(x_t = i, y_{0:t})p(x_{t+1} = j|x_t = i)p(y_{t+1:T}|x_{t+1} = j) \quad \text{(using HMM definition 3)} \\
&= \frac{1}{p(y_{0:T})} \alpha_t(i)a_{i,j}\beta_t(j) \\
&= \frac{1}{\sum_{i'}\sum_{j'} \alpha_t(i')a_{i',j'}\beta_t(j')} \alpha_t(i)a_{i,j}\beta_t(j) \qquad (11)
\end{aligned}$$

2.3 Computing $p(y_{0:T})$: Forward algorithm, Backward algorithm

Brute force computation of $p(y_{0:T})$ will require evaluating

$$p(y_{0:T}) = \sum_{x_{0:T}} p(x_0)[\prod_{t=1}^{T} p(x_t|x_{t-1})]p(y_0|x_0)[\prod_{t=1}^{T} p(y_t|x_t)] \qquad (12)$$

will require $O(N^T)$ computations.

2.3.1 Forward algorithm

A fast **and causal** way to compute $p(y_{0:T})$ is to go forward in time

$$p(y_{0:T}) = \sum_i \alpha_T(i) \tag{13}$$

The recursion for $\alpha_t(i)$ is given in (8). This takes $O(N^2T)$ computation only.

2.3.2 Backward Algorithm

Another $O(N^2T)$ way to compute $p(y_{0:T})$ is going backwards in time

$$p(y_{0:T}) = \sum_i \beta_0(i)\pi_i \tag{14}$$

The recursion for $\beta_t(i)$ is given in (9).

Typically one would use the forward algorithm to compute this, since that is also causal. There may be situations, e.g. if this is done offline and if observations are stored as last-in-first-out where one may need to use the backward algorithm.

2.4 EM algorithm for discrete-state HMM parameter estimation: Baum Welch algorithm

Let θ denote the set of parameters. In this case, θ includes all elements $\{a_{i,j}\}$, $\{\pi_i\}$ and the parameters of $b_i(y)$.

Assumption 1 *Assume for the discussion below that Y_t's are also discrete and take M possible values, $1, 2, \ldots M$. Thus, in $b_i(y)$, y can be $1, 2, \ldots M$.*

Then $\theta = \{\pi_i\}_{i=1,\ldots,N}, \{a_{i,j}\}_{i=1,\ldots N, j=1,\ldots N}, \{b_i(y)\}_{i=1\ldots N, y=1,\ldots M}$.

Let θ^k denote the parameter estimate at the k^{th} iteration. Recall that the EM algorithm computes

$$\begin{aligned} \theta^{k+1} &= \arg\max_\theta Q(\theta, \theta^k) \;\; s.t. \text{ constraints on } \theta \\ \text{where } \; Q(\theta, \theta^k) &\triangleq \mathbb{E}[\log p(y_{1:T}, X_{0:T}; \theta) | y_{1:T}; \theta^k] \end{aligned} \tag{15}$$

i.e. at each iteration EM maximizes the posterior expectation of the logarithm of the complete data likelihood (the posterior expectation is computed using the parameter estimates from the previous iteration). As discussed earlier (when talking about EM algorithm), under certain assumptions, this leads to maximization of the observed data likelihood, i.e. its solution converges to $\arg\max_\theta p(y_{0:T}; \theta)$.

Now for our HMM,

$$\log p(y_{0:T}, X_{0:T}; \theta) = \log \pi_{X_0} + \sum_{t=1}^{T} \log a_{X_{t-1},X_t} + \sum_{t=0}^{T} \log b_{X_t}(y_t) \tag{16}$$

Thus the first term is only a function of random variable X_0, the t^{th} entry of the second term is only a function of X_{t-1}, X_t and the t^{th} entry of the third term is only a function of X_t.

$$\begin{aligned}
& \mathbb{E}[\log p(y_{0:T}, X_{0:T}; \theta) | y_{0:T}; \theta^k] \\
= \; & \mathbb{E}[\log \pi_{X_0} | y_{0:T}; \theta^k] + \sum_{t=1}^{T} \mathbb{E}[\log a_{X_{t-1},X_t} | y_{0:T}; \theta^k] + \sum_{t=0}^{T} \mathbb{E}[\log b_{X_t}(y_t) | y_{0:T}; \theta^k] \\
= \; & \sum_i p(x_0 = i | y_{0:T}) \log \pi_i + \sum_{t=1}^{T} \sum_{i,j} p(x_{t-1} = i, x_t = j | y_{0:T}) \log a_{i,j} + \sum_{t=0}^{T} \sum_i p(x_t = i | y_{0:T}) \log b_i(y_t) \\
= \; & \sum_i \gamma_0^k(i) \log \pi_i + \sum_{t=1}^{T} \sum_{i,j} \xi_{t-1}^k(i,j) \log a_{i,j} + \sum_{t=0}^{T} \sum_i \gamma_t^k(i) \log b_i(y_t) \\
= \; & \sum_i \gamma_0^k(i) \log \pi_i + \sum_{i,j} \sum_{t=1}^{T} \xi_{t-1}^k(i,j) \log a_{i,j} + \sum_i \sum_{t=0}^{T} \gamma_t^k(i) \log b_i(y_t) \qquad (17)
\end{aligned}$$

where γ_t^k, ξ_t^k are computed using θ^k in the recursions given in (10) and (11).

We need to maximize the above subject to the constraints

$$\begin{aligned}
\sum_i \pi_i &= 1 \\
\sum_j a_{i,j} &= 1, \ \forall i = 1, \dots N \\
\sum_{y=1}^{M} b_i(y) &= 1, \ \forall i = 1, \dots N \qquad (18)
\end{aligned}$$

Using Lagrange multipliers, differentiating and solving, the final solutions are

$$\begin{aligned}
\pi_i^{k+1} &= \gamma_0^k(i) \\
a_{i,j}^{k+1} &= \frac{1}{\sum_{j'=1}^{N} \sum_{t=1}^{T} \xi_{t-1}^k(i,j')} (\sum_{t=1}^{T} \xi_{t-1}^k(i,j)) = \frac{1}{\sum_{t=1}^{T} \gamma_t(i)} (\sum_{t=1}^{T} \xi_{t-1}^k(i,j)) \\
b_i(m)^{k+1} &= \frac{1}{\sum_{t=0}^{T} \gamma_t^k(i)} \sum_{t=0}^{T} I(y_t = m) \gamma_t^k(i) \qquad (19)
\end{aligned}$$

where $I(A)$ is 1 if A occurs and 0 otherwise. Here γ_t^k, ξ_t^k are computed using the parameter estimates at iteration k in the recursions given earlier.

Thus, the stepwise EM algorithm is as follows. At iteration $k+1$,

1. Compute $\gamma_t^k(i)$ for all i for all t using (10) and parameter estimates from iteration k, θ^k.
2. Compute $\xi_t^k(i,j)$ for all i,j for all t using (11) and parameter estimates from iteration k, θ^k.
3. Compute parameter estimates at iteration $k+1$, θ^{k+1}, using (19.

Now, if Y_t's are not discrete, but are continuous r.v.'s with parameters of their PDF being governed by the current state, e.g. Y_t's can be scalar Gaussians with mean μ_i and variance σ_i^2 if the state $X_t = i$. In this case, their estimates can be computed as follows.

We need to maximize the following w.r.t. μ_i, σ_i^2.

$$\sum_i \sum_{t=0}^{T} \gamma_t^k(i) \log b_i(y_t) \quad = \quad \sum_i \sum_{t=0}^{T} \gamma_t^k(i) [-\log(\sqrt{2\pi})\sigma_i^2 - \frac{(y_t-\mu_i)^2}{2\sigma_i^2}] \tag{20}$$

Thus,

$$\begin{aligned} \mu_i^{k+1} &= \frac{1}{\sum_{t=0}^T \gamma_t^k(i)} \sum_{t=0}^{T} \gamma_t^k(i) y_t \\ \sigma_i^{2k+1} &= \frac{1}{\sum_{t=0}^T \gamma_t^k(i)} \sum_{t=0}^{T} \gamma_t^k(i)(y_t - \mu_i^{k+1})^2 \end{aligned} \tag{21}$$

2.5 General idea of Viterbi algorithm / dynamic programming

In dynamic programming / Viterbi algorithm, the goal is to find

$$\arg max_{q_{0:T}} f_T(q_{0:T}) \tag{22}$$

where $f_t(q_{0:t})$ at any t satisfies

$$f_t(q_{0:t}) \quad = \quad f_{t-1}(q_{0:t-1}) + h_t(q_{t-1}, q_t) + g_t(q_t) \tag{23}$$

Notice that $f_t(.)$ is a function only of the first $t+1$ variables. Typically path optimization problems are of this type.

Efficient solutions strategy: Let

$$\delta_t(i) \quad \triangleq \quad \max_{q_{0:t-1}} f_t(q_{0:t-1}, i) \tag{24}$$

Then, using (23),

$$\begin{aligned} \delta_t(i) &= \max_{q_{0:t-1}} [f_{t-1}(q_{0:t-1}) + h_t(q_{t-1}, i) + g_t(i)] \\ &= \max_{q_{t-1}} \max_{q_{0:t-2}} [f_{t-1}(q_{0:t-1}) + h_t(q_{t-1}, i) + g_t(i)] \\ &= g_t(i) + \max_{q_{t-1}} [h_t(q_{t-1}, i) + \max_{q_{0:t-2}} f_{t-1}(q_{0:t-1})] \\ &= g_t(i) + \max_{q_{t-1}} [h_t(q_{t-1}, i) + \delta_{t-1}(q_{t-1})] \end{aligned} \tag{25}$$

Also store the optimal path to get to q_t for each value of q_t. So if $q_t \in \{1, 2, \dots N\}$ then store the optimal path to get to $q_t = i$ for each i in the set. For the above problem, this can be done efficiently by only storing the optimal value of q_{t-1} that gets you to $q_t = i$ and doing this for each i at each t. Thus, at each t, for each $q_t = i$, we store

$$\psi_t(i) \triangleq \arg\max_{q_{t-1}} [h_t(q_{t-1}, i) + \delta_{t-1}(q_{t-1})] \tag{26}$$

To summarize the above idea, we do the following.

1. Initialize at $t = 0$ to $\delta_0(i) = f_0(i)$ for all $i = 1, 2 \dots N$

2. Starting at $t = 1$, at each t, compute the following for all $i = 1, 2 \dots N$

$$\delta_t(i) = g_t(i) + \max_{q_{t-1}} [h_t(q_{t-1}, i) + \delta_{t-1}(q_{t-1})] \tag{27}$$

3. Simultaneously, at each t, for each $i = 1, 2 \dots N$ also store the maximizer of the above, i.e. store

$$\psi_t(i) = \arg\max_{q_{t-1}} [h_t(q_{t-1}, i) + \delta_{t-1}(q_{t-1})] \tag{28}$$

4. At $t = T$, find the optimal cost and the optimal value of q_T as

$$\max_i \delta_T(i),$$
$$q_T^* = \arg\max_i \delta_T(i) \tag{29}$$

5. Backtrack using ψ_t to find the optimal state sequence, i.e. starting with $t = T - 1$, go backwards,

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \tag{30}$$

2.6 Posterior MAP (non-causal) computation: Viterbi algorithm

We would like to find the non-causal posterior MAP estimate is

$$x_{0:T}^* \triangleq \arg\max_{x_{0:T}} p(x_{0:T}|y_{0:T}) = \arg\max_{x_{0:T}} p(x_{0:T}, y_{0:T}) = \arg\max_{x_{0:T}} \log p(x_{0:T}, y_{0:T}) \tag{31}$$

Using notation from above,

$$f_t(x_{0:t}) := \log p(x_{0:t}, y_{0:t}) \tag{32}$$

Using the HMM definition, it is easy to see that

$$f_t(x_{0:t}) = f_{t-1}(x_{0:t-1}) + \log p(x_t|x_{t-1}) + \log p(y_t|x_t) \tag{33}$$

Thus,

$$h_t(x_{t-1}, x_t) := \log p(x_t|x_{t-1})$$
$$g_t(x_t) := \log p(y_t|x_t) \tag{34}$$

Thus, the final Viterbi algorithm is

1. Initialize at $t = 0$ to $\delta_0(i) = f_0(i) = \pi_i b_i(y_0)$ for all $i = 1, 2 \ldots N$

2. Starting at $t = 1$, at each t, compute the following for all $i = 1, 2 \ldots N$

$$\delta_t(i) = \log b_i(y_t) + \max_{x_{t-1}=1,2,\ldots N} [\log a_{x_{t-1},i} + \delta_{t-1}(x_{t-1})] \tag{35}$$

3. Simultaneously, at each t, for each $i = 1, 2 \ldots N$ also store the maximizer of the above, i.e. store

$$\psi_t(i) = \arg \max_{x_{t-1}=1,2,\ldots N} [\log a_{x_{t-1},i} + \delta_{t-1}(x_{t-1})] \tag{36}$$

4. At $t = T$, find the optimal cost and the optimal value of q_T as

$$\begin{aligned} &\max_i \delta_T(i), \\ x_T^* &= \arg \max_i \delta_T(i) \end{aligned} \tag{37}$$

5. Backtrack using ψ_t to find the optimal state sequence, i.e. starting with $t = T - 1$, go backwards,

$$z_t^* = \psi_{t+1}(x_{t+1}^*) \tag{38}$$

2.7 Direct derivation of Viterbi algorithm for HMMs

Let

$$\begin{aligned} \delta_t(x_t) &\triangleq \max_{x_{0:t-1}} p(x_{0:t}, y_{0:t}) \\ \psi_t(x_t) &\triangleq \arg \max_{1 \leq x_{t-1} \leq N} \delta_{t-1}(x_{t-1}) a_{x_{t-1},x_t} \end{aligned} \tag{39}$$

Recursion for δ_t

$$\begin{aligned} \delta_t(i) &\triangleq \max_{x_{0:t-1}} p(x_{0:t}, y_{0:t}) \\ &= \max_{x_{0:t-1}} p(x_{0:t-1}, y_{0:t-1}) p(x_t = i | x_{0:t-1}, y_{0:t-1}) p(y_t | x_t, x_{0:t-1}, y_{0:t-1}) \\ &= \max_{x_{0:t-1}} p(x_{0:t-1}, y_{0:t-1}) p(x_t = i | x_{t-1}) p(y_t | x_t = i) \quad \text{(using HMM definition 3)} \\ &= \max_{x_{0:t-1}} p(x_{0:t-1}, y_{0:t-1}) a_{x_{t-1},i} b_i(y_t) \\ &= b_i(y_t) \max_{x_{t-1}} (\max_{x_{0:t-2}} p(x_{0:t-1}, y_{0:t-1})) a_{x_{t-1},i} \\ &= b_i(y_t) \max_j \delta_{t-1}(j) a_{j,i} \end{aligned} \tag{40}$$

Also,

$$\psi_t(i) = \arg \max_j \delta_{t-1}(j) a_{j,i} \tag{41}$$

Thus,

$$\begin{aligned} x_T^* &= \arg \max_{1 \leq i \leq N} \delta_T(i) \\ x_t^* &= \psi_{t+1}(x_{t+1}^*), \ \forall \ t = T-1, T-2, \ldots 0 \end{aligned} \tag{42}$$