Hidden Markov Models and the Viterbi algorithm

The metaphor we used for a hidden Markov model was a set of urns labeled $1, \ldots, N$ behind a curtain. Each urn has balls of various colors (a total of M possible colors). A "genie" 1) chooses an urn according to some initial probability distribution, 2) chooses a ball from that urn, calls out its color, and replaces it in the urn, 3) picks a new urn according to some probability that depends only on the current urn, and continues steps 2,3, calling out a series of colors. The colors are the "observations" O and the series of urns correspond to the "hidden states" q to which we have no access. The initial probability weights in step 1 are given by w_i, the "emission" probabilities in step 2 are given by $e_i(a) = p(O = a|q = i)$, and the urn transitions in step 3 are given by an underlying set of Markov transitions characteried by a transition matrix T_{ij}. An HMM $H = \big(T_{ij}, e_i(a), w_i\big)$ is understood to have N hidden Markov states labelled by i $(1 \le i \le N)$, and M possible observables for each state, labelled by a $(1 \le a \le M)$. The state transition probabilies are $T_{ij} = p(q_{t+1} = j \,|\, q_t = i)$, $1 \le i, j \le N$ (where q_t is the hidden state at time t), the emission probability for the observable a from state i is $e_i(a) = p(O_t = a \,|\, q_t = i)$ (where O_t is the observation at time t), and the initial state probabilities are $w_i = p(q_1 = i)$.

Given a sequence of observations $O = O_1\, O_2 \cdots O_T$, and an HMM $H = \big(T_{ij}, e_i(a), w_i\big)$, we wish to find the maximum probability state path $Q = q_1\, q_2 \cdots q_T$. This can be done recursively using the Viterbi algorithm.

Let $v_i(t)$ be the probability of the most probable path ending in state i at time t, i.e.,

$$v_i(t) = \max_{q_1, q_2, \cdots q_{t-1}} P(q_1\, q_2 \cdots q_{t-1}, q_t = i, O_1\, O_2 \cdots O_t \,|\, H) \ ,$$

and let w_i be the initial probabilities of the states i at time $t = 1$.[†]

Then $v_j(t)$ can be calculated recursively using

$$v_j(t) = \max_{1 \le i \le N} \big[v_i(t-1)\, T_{ij}\big]\, e_j(O_t)$$

together with initialization

$$v_i(1) = w_i\, e_i(O_1) \qquad 1 \le i \le N$$

and termination

$$P^* = \max_{1 \le i \le N} [v_i(T)]$$

[†] Note this notation avoids the frightening greek letters δ, π, and λ used in the Rabiner notes, using instead v for Viterbi, w for weights, and H for hidden Markov model. The correspondence with the notation used in the Rabiner notes is $v_i(t) \leftrightarrow \delta_t(i)$, $e_i(a) \leftrightarrow b_i(a)$, $T_{ij} \leftrightarrow a_{ij}$, $w_i \leftrightarrow \pi_i$, $H \leftrightarrow \lambda$.

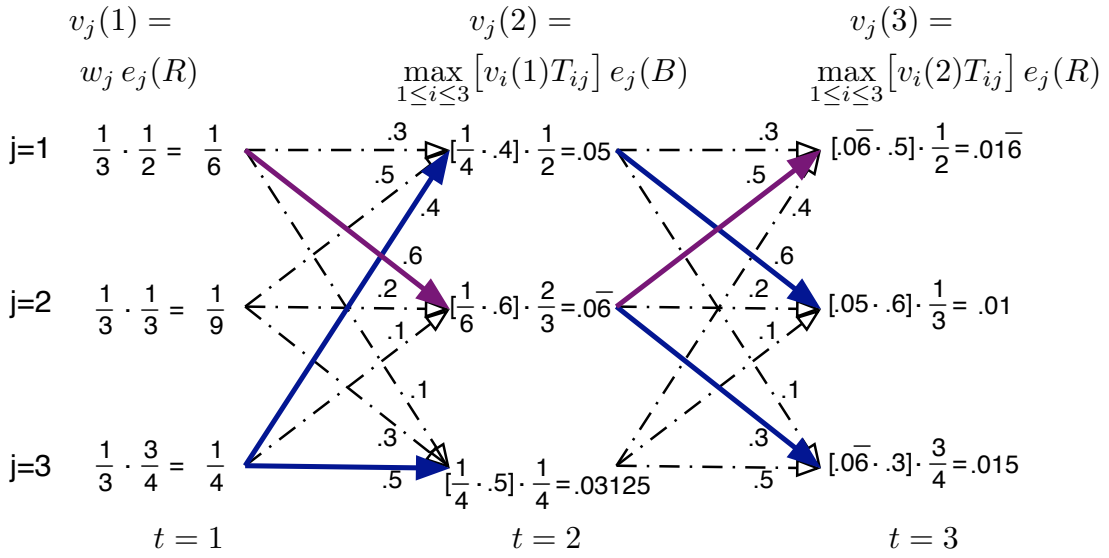
(i.e., at the end we choose the highest probability endpoint, and then we backtrack from there to find the highest probability path).

This algorithm has properties similar to the Dijkstra algorithm, discussed in the context of finding shortest length paths between two points. In that case, we needed to consider only the shortest length path from the start point to any intermediate point, since any longer path to the intermediate point would necessarily result in a longer total path from start to endpoint. In this case, we need to consider only the maximum probability path to any intermediate observation, since a lower probability path would necessarily result in a lower total probability path in the state space from initial to final observation. For a series of T observations, the total number of possible paths for an N state model is N^T (N possible states at each time), i.e., exponential in T, and this quickly grows prohibitively large. (For $T = 100$ observations in an $N = 5$ model, there would already be $5^{100} \approx 10^{70}$ possible paths.) The Viterbi algorithm instead finds the most probable path in computational time linear in the number of observations T.

Note that the maximally likely path is not the only possible optimality criterion, for example choosing the most likely state at any given time requires a different algorithm and can give a slightly different result (see last page of notes to follow). But the overall most likely path provided by the Viterbi algorithm provides an optimal state sequence for many purposes. One metaphor for describing it is the "lazy instructor" who gives an exam but does not want to solve the problems him or herself. So the exams are collected, and the instructor has a couple of methodologies to construct a solution sheet: either using the consensus of all students on each problem to determine the correct solutions, or finding a single student, known to be the best in the class, and using that student's answers as the solution sheet. The Viterbi algorithm corresponds to the latter methodology.

Some python code implementing the algorithm can be found here:
http://nbviewer.ipython.org/url/courses.cit.cornell.edu/info2950%5F2017sp/resources/viterbi.ipynb
and this notebook also redoes the example on the next page here, finding the Viterbi path for three observations in a simple three state model:

To illustrate this, consider a three state HMM, with R or B emitted by each state (e.g., three urns, each with red or blue balls) with emission probabilities $e_1(R) = 1/2$, $e_2(R) = 1/3$, and $e_3(R) = 3/4$ (and correspondingly $e_1(B) = 1/2$, $e_2(B) = 2/3$, and $e_3(B) = 1/4$), state transition matrix $T_{ij} = \begin{pmatrix} .3 & .6 & .1 \\ .5 & .2 & .3 \\ .4 & .1 & .5 \end{pmatrix}$, and initial state probabilities $w_i = 1/3$. Suppose we observe the sequence RBR, then we can find the "optimal" state sequence to explain this sequence of observations by running the Viterbi algorithm by hand:



In the first step, we initialize the probabilities at $t = 1$ to $v_j(t = 1) = w_j\, e_j(R)$ for each $j = 1, 2, 3$. These are given in the first column to the left, as 1/6, 1/9, 1/4, respectively.

In the second step, $t = 2$, we determine first $v_1(t = 2)$ by considering the three quantities $v_i(1)p_{i1}$ for $i = 1, 2, 3$. They are respectively $(1/6)\cdot.3$, $(1/9)\cdot.5$, and $(1/4)\cdot.4$. The third one is the largest, so according to the algorithm we set $v_1(2) = [(1/4)\cdot.4]\cdot(1/2) = .05$, and remember that the maximum probability path to state $j = 1$ at time $t = 2$ came from state $j = 3$ at time $t = 1$ (blue line). Similarly, to determine $v_2(2)$ we consider the three quantities $v_i(1)p_{i2}$ for $i = 1, 2, 3$, respectively $(1/6)\cdot.6$, $(1/9)\cdot.2$, $(1/4)\cdot.1$, and the first is the largest, so we set $v_2(2) = [(1/6)\cdot.6]\cdot(2/3) = .0\bar{6}$. Finally, to determine $v_3(2)$ we consider the three quantities $v_i(1)p_{i3}$ for $i = 1, 2, 3$, respectively $(1/6)\cdot.1$, $(1/9)\cdot.3$, $(1/4)\cdot.5$, and the third is the largest, so we set $v_3(2) = [(1/4)\cdot.5]\cdot(1/4) = .03125$.

In the third step, $t = 3$, we determine first $v_1(t = 3)$ by considering the three quantities $v_i(2)p_{i1}$ for $i = 1, 2, 3$. They are respectively $.05\cdot.3$, $.0\bar{6}\cdot.5$, and $.03125\cdot.4$. The second is the largest, so according to the algorithm we set $v_1(3) = [.0\bar{6}\cdot.5]\cdot(1/2) = .01\bar{6}$, and remember that the maximum probability path to state $j = 1$ at time $t = 3$ came from state $j = 2$ at time $t = 2$ (blue line). Similarly, to determine $v_2(3)$ we consider the three quantities $v_i(2)p_{i2}$ for $i = 1, 2, 3$, respectively $.05\cdot.6$, $.0\bar{6}\cdot.2$, $.03125\cdot.1$, and the first is the largest, so we set $v_2(3) = [.05\cdot.6]\cdot(1/3) = .01$. Finally, to determine $v_3(2)$ we consider the three quantities $v_i(1)p_{i3}$ for $i = 1, 2, 3$, respectively $.05\cdot.1$, $.0\bar{6}\cdot.3$, $.03125\cdot.5$, and the second is the largest, so we set $v_3(3) = [.0\bar{6}\cdot.3]\cdot(3/4) = .015$.

Since there are only three observations, we can now use the termination step to determine that the maximum probability for the observations $O = RBR$ is $P^* = .01\bar{6}$ with state path $Q = 1, 2, 1$ (purple lines).

In a general HMM, the individual probability of any state i at any time t is given by summing the probabilities for all possible state paths that pass through i at time t, i.e., those with $q_t = i$:

$$p(q_t = i) = \sum_{q_1,\ldots,q_{t-1},q_{t+1},\ldots,q_T} p(\{O\},\{q\}|H)$$

(where the summations are over all states from 1 to N at all times other than the t of interest). Note that the state j of the output of the Viterbi algorithm, i.e., the state at time t on the maximum likelihood path, is not always the individually most likely state at that time.

Consider the simplest possible HMM with $N = 2$ and $M = 2$, i.e., with two hidden Markov states 1,2 and state transition matrix T_{ij} $(1 \le i, j \le 2)$, two observables R, B per state with emission probabilities $e_i(a)$ $(a = R, B)$, and initial state probabilities w_i. Consider a sequence of just two observations RR. Let

$$P(i,j) = P(q_1 = i, q_2 = j, O_1 = R, O_2 = R \,|\, H)$$

be the probability of the state path i, j given the observation. Then we can calculate

$$P(i,j) = w_i \, e_i(R) \, T_{ij} \, e_j(R) \ .$$

There are only four possible state paths in this model (11,12,21,22), and the probability of state i at time $t = 1$ is $P(i,1) + P(i,2)$. As mentioned, the individually most probable state at any given time t is not necessarily on the maximally likely path, depending on the parameters of the model.

Here we'll choose parameters with p_{11} and $e_1(R)$ large enough to ensure that 11 will be the most probable state path, but with p_{21} and p_{22} comparable to one another so that the sum of the two paths starting at 2 will nonetheless outweigh the sum of the two paths starting at 1.

Consider the transition matrix $T_{ij} = \begin{pmatrix} .7 & .3 \\ .5 & .5 \end{pmatrix}$, and emission probabilities $e_1(R) = .8$, $e_2(R) = .9$, and initial state probabilities $w_1 = w_2 = 1/2$. Then the probabilities of the four paths are

$$P(1,1) = \frac{1}{2} \cdot .8 \cdot .7 \cdot .8 = .224$$

$$P(1,2) = \frac{1}{2} \cdot .8 \cdot .3 \cdot .9 = .108$$

$$P(2,1) = \frac{1}{2} \cdot .9 \cdot .5 \cdot .8 = .18$$

$$P(2,2) = \frac{1}{2} \cdot .9 \cdot .5 \cdot .9 = .2025$$

The most probable path, the result of the Viterbi algorithm in this case, is the first: state path 11. But the overall probability of state 1 at time 1 is $P(1,1) + P(1,2) = .332$ while the probability of state 2 at time 1 is $P(2,1) + P(2,2) = .3825$ and therefore state 2 is the individually most probable state at time 1, even though it is not the output of the Viterbi algorithm for time 1.