

LECTURE 11: PROFILE HMMs continued: Viterbi algorithmReview

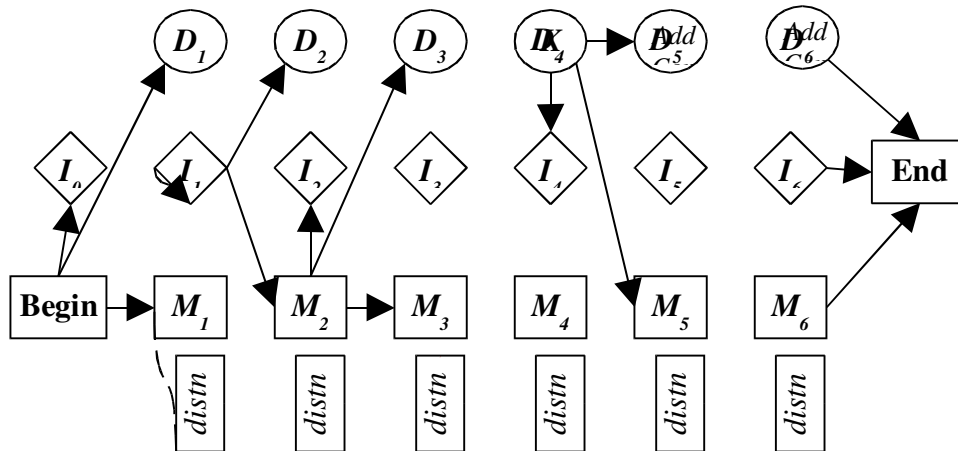
-Profile HMM

-Example: DNA Matching – fitting two DNA sequences to a common ancestor

Structure of Profile HMM

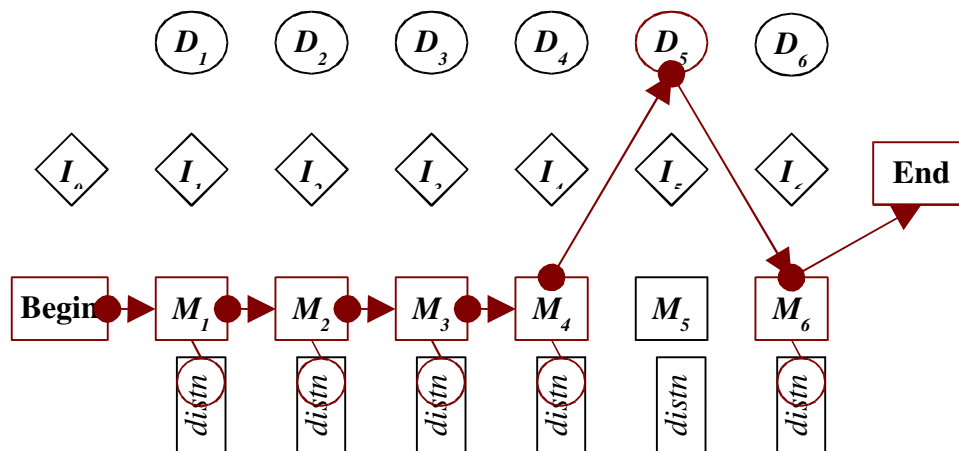
- Three types of states: Match States, Insert States, Delete States
- Match State $\square$
 - o Denoted as M_i or, pictorially, by squares
 - o One for each of the i output states of ancestor's DNA, $i = 1, \dots, K$
 - o Produces either Ancestor's output state (presumably with a high probability) or a Mutation (presumably less likely)
 - o Always a non-silent state
- Insert State $\diamond$
 - o Denoted I_i or, pictorially, by diamonds
 - o One after the Begin state and One above each of the Match states (i.e., $i = 0, \dots, K$)
 - o Produces an output state according to some probability distribution
 - o Always a non-silent state
 - o Only state that may transition back to itself
- Delete State $\bigcirc$
 - o Denoted D_i or, pictorially, by a circle
 - o One corresponding to each of the Match states
 - o Always a silent state
 - o Note: Doesn't "delete" previous output in chain, but rather provides mechanism to skip Match state of ancestor's DNA, i.e., "delete" refers to the ancestor Match state, not the previous state of current Markov chain
- Possible Transitions: For $i=1, \dots, K-1$,
 - o $M_i \rightarrow M_{i+1}, I_{i+1}, D_{i+1}$
 - o $I_i \rightarrow I_i, D_{i+1}, M_{i+1}$
 - o $D_i \rightarrow M_{i+1}, I_{i+1}, D_{i+1}$
 - o $\rightarrow$ Means "may transition to"
 - o BEGIN $\rightarrow I_0, D_1, M_1$
 - o $I_K \rightarrow I_K, \text{END}$
 - o $D_K, M_K \rightarrow I_K, \text{END}$
- Goal of HMM: Given Output Sequence
 - o Find the most probable path for HMM that yields the output sequence
 - o Note: There may be more than one path you can take to yield the given output sequence. However, since these paths will pass through a different ordered set of states for the HMM, they are considered "different" paths.
- Aligned Output Sequences:
 - o $_$ denote locations where HMM went through a Delete state
 - o Lower case output states indicate where HMM went through an Insert State
 - o Upper case letters indicate output where HMM went through a Match State

- . denote locations where one HMM went through an insert state but no such state was observed for another HMM. (ex. Below)

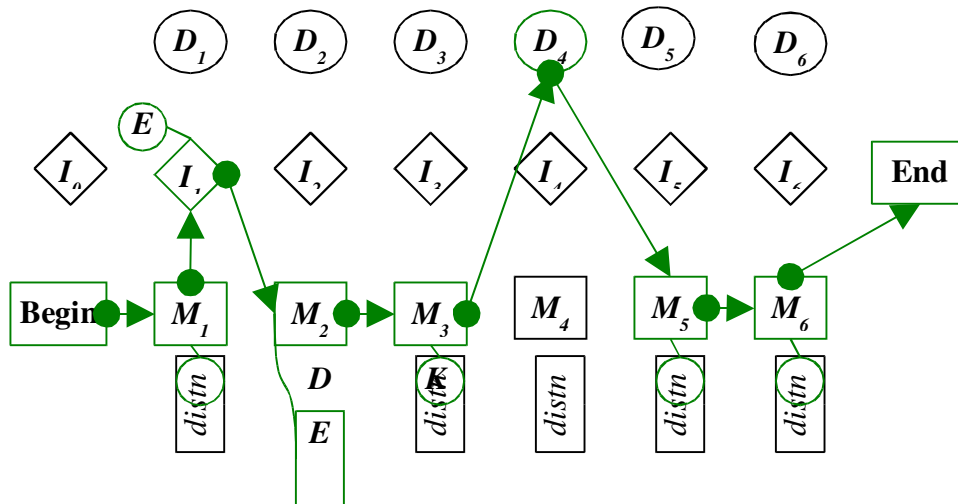


Possible Transitions in HMM for DNA Ancestor ADKTSN (not all are shown):

Possible Path of HMM to produce Output Sequence ADTKN:



Possible Path of HMM to produce Output Sequence AEEKSN:



- Example of Aligning Output Sequences:
 - Ancestor: ADKTSN
 - Outputs: ADKTN
AEEKSN
 - Representations of Outputs based on paths drawn above:
 - Ancestor: ADKTSN
 - Outputs: A.DKT_N
AeEK_SN
 - Note: “.” does not correspond to output from HMM or any state of the HMM, but simply serves as a placeholder for sequence alignment.
- Asides/Questions:
 - HMM does not behave like evolutionary process, rather creates the same output as an evolutionary process
 - AEK is not equivalent to A_eK because these are produced by different paths of HMM.

Viterbi Modification for Profile HMMs (sketch)

Given $\vec{o} = (o_0, o_1, \dots, o_{L-1})$ is an entire output sequence of length L . In the following, Q tracks all of the positions of the Markov chain between BEGIN and END, but it does not include BEGIN and END. A little care is needed in treating the beginning and end of the iteration below. We imagine that π gives the initial distribution of the HMM, which we regard as being concentrated on the states I_0, D_1, M_1 . The matrix $a = (a_{i,j})$ gives the transition probabilities between the match, insert and delete states (indexed by i and j) of the HMM, and the matrix $b = (b_{j,o})$ gives the probabilities of outputting each of the 20 amino acids (indexed by o) when the HMM is in a match or insert state (indexed by j). For convenience, we shall define $a_{i,end}$ as denoting the probability of making a transition from the states $i = D_K, I_K, M_K$ to END.

Define the objective function as:

$$V(O^*) = \max_q \Pr(Q = q, O = o^*)$$

$$= \max_{m=L} \max_{\substack{q \text{ has length } m \text{ and} \\ \text{has } L \text{ non-silent states}}} \Pr(Q = q, O = o^*)$$

where the second equality holds because the number of sites visited by the HMM chain must be greater than or equal to L .

- Note: The number of states m that the HMM chain goes through is equal to the length of the output + the number of Delete states visited
- $m - L = \#$ of silent states visited

Dynamic Programming (for $0 \leq n \leq L$)

Notation reminder:

$$o|_{n-1} = (o_0, o_1, \dots, o_{n-1}), \quad q \text{ of length } m: q = (q_0, q_1, \dots, q_{m-1}), \quad Q|_{m-1} = (Q_0, Q_1, \dots, Q_{m-1})$$

$$\delta_n(l) = \max_{\substack{m=n \\ q \text{ has length } m \\ n \text{ non-silent states} \\ q_{m-1} = l}} \max_{\substack{q \text{ has length } m \\ n \text{ non-silent states} \\ q_{m-1} = l}} \Pr(Q|_{m-1} = q, O|_{n-1} = o^*|_{n-1})$$

Initiation:

For $n=0$, $\delta_0(l)$ can be non-zero only if l is a silent state. For $l = D_1$, $\delta_0(l) = \pi_{D_1}$. For other delete states l , $\delta_0(l)$ can be determined in an iterative manner, similar to that for Part II below. For $n=1$ and $l = I_0$, $\delta_1(I_0) = \pi_{I_0} b_{I_0, o_0^*}$ and for $l = M_1$, $\delta_1(M_1) = \pi_{M_1} b_{M_1, o_0^*}$. Given these initial values, other values of $\delta_n(l)$ can be generated by using the iterative relations below.

Iteration (Heart of the Algorithm) Part I: ℓ not silent (match or insert state)

$$\begin{aligned}
 \delta_{n+1}(l) &= \max_{m | n \geq 1} \max_{\substack{q \text{ has length } m \\ n+1 \text{ non-silent states} \\ q_{m-1}=l}} \Pr(Q|_{m-1}=q, O|_n=o^*_n) \\
 &= \max_{m | n \geq 1} \max_{\substack{j=1 \\ n \text{ non-silent states} \\ q_{m-2}=j}}^N \max_{\substack{q \text{ has length } m-1 \\ n \text{ non-silent states} \\ q_{m-2}=j}} \Pr(Q_{m-1}=l, O_n=o^*_n | Q_{m-2}=j, Q|_{m-3}=q|_{m-3}, O|_{n-1}=o^*_{n-1}) \\
 &\quad \text{Add Conditioning} \\
 \delta_{n+1}(l) &= \max_{m | n \geq 1} \max_{\substack{j=1 \\ n \text{ non-silent states} \\ q_{m-2}=j}}^N \max_{\substack{q \text{ has length } m-1 \\ n \text{ non-silent states} \\ q_{m-2}=j}} a_{j,l} b_{l, o_n^*} \Pr(Q|_{m-2}=q|_{m-2}, O|_{n-1}=o^*_{n-1}) \\
 &\quad \text{Using Markov property of HMM} \\
 &= \max_{j=1}^N a_{j,l} b_{l, o_n^*} \max_{\substack{q \text{ has length } m-1 \\ n \text{ non-silent states} \\ q_{m-2}=j}} \max_{\substack{q \text{ has length } m-1 \\ n \text{ non-silent states} \\ q_{m-2}=j}} \Pr(Q|_{m-2}=q|_{m-2}, O|_{n-1}=o^*_{n-1}) \\
 &= \max_{j=1}^N a_{j,l} b_{l, o_n^*} \delta_n(j)
 \end{aligned}$$

Iteration, Part II: ℓ is silent (delete state).

Now, we can't break up the output sequence during conditioning:

$$\begin{aligned}
 \delta_{n+1}(l) &= \max_{m | n \geq 2} \max_{\substack{q \text{ has length } m \\ n+1 \text{ non-silent states} \\ q_{m-1}=l}} \Pr(Q|_{m-1}=q, O|_n=o^*_n) \\
 &= \max_{m | n \geq 2} \max_{\substack{j=1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}}^N \max_{\substack{q \text{ has length } m-1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}} \Pr(Q_{m-1}=l | Q_{m-2}=j) * \Pr(Q|_{m-2}=q|_{m-2}, O|_n=o^*_n) \\
 &= \max_{m | n \geq 2} \max_{\substack{j=1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}}^N \max_{\substack{q \text{ has length } m-1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}} a_{j,l} * \Pr(Q|_{m-2}=q|_{m-2}, O|_n=o^*_n) \\
 &= \max_{j=1}^N a_{j,l} \max_{\substack{m-1 | n \geq 1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}} \max_{\substack{q \text{ has length } m-1 \\ n+1 \text{ non-silent states} \\ q_{m-2}=j}} \Pr(Q|_{m-2}=q|_{m-2}, O|_n=o^*_n) \\
 &= \max_{j=1}^N a_{j,l} \delta_{n+1}(j)
 \end{aligned}$$

Hence, when ℓ is silent, our iterative logic seems to lead us right back to where we began!

Resolution

Number the states of the Markov Chain in increasing order so that the non-silent states have lower numbers than silent states and label silent states in left to right order

In particular, for ℓ silent,

$$a_{j,\ell} > 0 \text{ for } j \text{ silent only if } j < \ell$$

Then, for ℓ silent,

$$\delta_{n+1}(\ell) = \max_{j=1}^N a_{j,\ell} \delta_{n+1}(j) = \max \left\{ \max_{j \text{ non-silent}} a_{j,\ell} \delta_{n+1}(j), \max_{\substack{j \text{ silent} \\ j < \ell}} a_{j,\ell} \delta_{n+1}(j) \right\}.$$

Now, $\max_{j \text{ non-silent}} a_{j,\ell} \delta_{n+1}(j)$ can be written in terms of $\delta_n(p)$, $p = 1, \dots, N$ (by part I), and by computing $\delta_{n+1}(\ell)$ successively as ℓ ranges from lower numbered to higher numbered silent states, we avoid any loop in the iteration.

Finally,

$$V(O^*) = \max_{\ell = D_K, I_K, M_K} (\delta_L(\ell) a_{\ell, \text{end}})$$

PARAMETER ESTIMATION (for ordinary HMMs without silent states)

Best Situation – Training data is annotated, so you can see paths of the chain and know what output corresponds to what path. We'll discuss the situation of less information in the next lecture.

If we have training data (O^i, Q^i) , $i=1, \dots, m$, that are samples of output and associated sequence of HMM states (e.g. annotated CpG Island data), we can use a frequentist treatment to estimate $\lambda = (\pi, a, b)$:

For each state $j, k \in \{1, \dots, N\}$ of the Markov Chain, Compute:

$$A_{jk} = \# \text{ transitions from } j \text{ to } k \text{ in data} + r_{jk}$$

where r_{jk} is a “pseudo-count,” some small positive amount (to avoid zero values for A_{jk})

For each possible output, $\ell=1, \dots, M$ and state $j=1, \dots, N$, compute:

$$B_{jk} = \# \text{ outputs of } \ell \text{ from state } j \text{ in data} + s_j$$

where s_{jk} is a pseudo-count.

For each state j , let

$$\Pi_j = (\# \text{ samples } i \text{ where } Q_0^i = j) + t_j$$

where t_j is a pseudo-count.

Then, use the estimates:

$$a_{jk}^- = \frac{A_{j,k}}{\sum_{k'} A_{j,k'}}$$

$$b_j^- = \frac{B_{j,\ell}}{\sum_{\ell'} B_{j,\ell'}}$$

$$\bar{\pi}_j = \frac{\Pi_j}{\sum_{j'} \Pi_{j'}}$$