Algorithmic Bioinformatics DD2450, spring 2010, Lecture 6

Lecturer Jens Lagergren
Several current and previous students
will be acknowledged in a separate document.

April 21, 2010

Chapter 1

Hidden Markov Models

Definition 1. *A Hidden Markov Model (HMM) is a 6-tuple $M = (\Sigma,\ Q,\ q_I,\ q_T,\ A,\ E)$ where:*

1. $\sum = \{\sigma_1,\ \sigma_2,\ \dots,\ \sigma_r\}$ *is an alphabet.*
2. $Q = \{q_0,\ q_1,\ \dots,\ q_i\}$ *is a set of states.*
3. q_I*, is the initial state.*
4. q_T*, is the terminal state.*
5. $A = \{a_{qq'} : q, q' \in Q\}$ *are transition probabilities such that*

$$\sum_{q'} a_{qq'} = 1, \quad \forall q \in Q$$

.

6. $E = \{e_q(\sigma) : q \in Q, \sigma \in \Sigma\}$ *are emission probabilities such that*

$$\sum_{\sigma \in \Sigma} e_q(\sigma) = 1, \quad \forall q \in Q$$

.

"Behaviour" of a HMM:

1. *Start in state $q = q_I$ (for this state no symbol is emitted).*
2. *"Transit" to state q' according to $a_{qq'}$.*
3. *In q', emit a symbol $\sigma \in \Sigma$ according to $e_{q'}(\sigma)$.*
4. *If $q' = q_T$ stop, otherwise set $q = q'$, then continue at (2).*

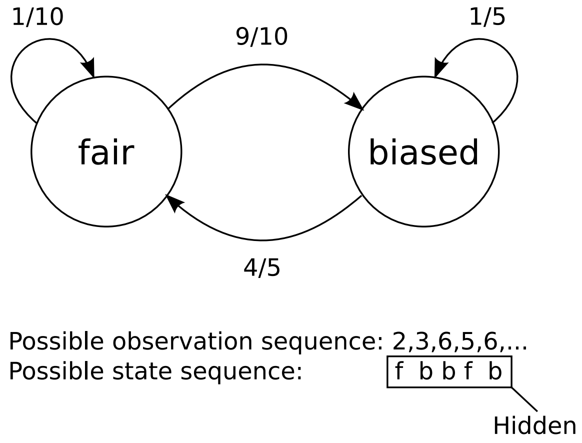

Figure 1.1: An example of an HMM for a casino that have two different dices one fair and one biased. The possible observation sequence is a sequence of observable outcomes of a dice. The possible state sequence is the state that the model was in for the different observations, in this case the different dice used. The state sequence is usually hidden.

The probability that a HMM emitted a certain sequence of its alphabet is:

$$Pr[x_1, x_2, \dots, x_n, \pi_0, \pi_1, \pi_2, \dots, \pi_n] = Pr[x_1, x_2, \dots, x_n, \pi_1, \pi_2, \dots, \pi_n] \quad (1.1)$$

where

$$\pi_0, \pi_1, \dots, \pi_n, \quad \pi_i \in Q$$

$$x_1, x_2, \dots, x_n, \quad x_i \in \Sigma$$

x_i is the symbol emitted in state π_i. Since we always know the initial state ($\pi_0 = q_I$), it presents no uncertainty, and therefore we need not consider π_0 in (1.1).

Observe that a Markov property is satisfied:

$$Pr[x_n, \pi_n | x_1, x_2, \dots, x_{n.1}, \pi_1, \dots, \pi_{n-1}] = Pr[x_n, \pi_n | \pi_{n-1}]$$

and also

$$Pr[x_n, \pi_n | \pi_{n-1}] = a_{\pi_{n-1}\pi_n} e_{\pi_n}(x_n) \quad (1.2)$$

Claim 1.

$$Pr[x_1, \dots, x_n, \pi_1, \dots, \pi_n] = \prod_{i=1}^{n} a_{\pi_{i-1}\pi_i} e_{\pi_i}(x_i)$$

Proof. We will do this proof by induction.
Base case:

$$Pr[x_1, \pi_1] = a_{\pi_0,\pi_1} e_{\pi_1}(x_1) = \prod_{i=1}^{1} a_{\pi_{i-1},\pi_i} e_{\pi_i}(x_i)$$

Induction assumption:

$$Pr[x_1, ..., x_{n-1}, \pi_1, ..., \pi_{n-1}] = \prod_{i=1}^{n-1} a_{\pi_{i-1}\pi_i} e_{\pi_i}(x_i)$$

It follows that

$$\begin{aligned} Pr[x_1, ..., x_n, \pi_1, ..., \pi_n] &= Pr[x_n, \pi_n | x_1, ..., x_{n-1}, \pi_1, ..., \pi_{n-1}] Pr[x_1, ..., x_{n-1}, \pi_1, ..., \pi_{n-1}] \\ &= Pr[x_n, \pi_n | \pi_{n-1}] Pr[x_1, ..., x_{n-1}, \pi_1, ..., \pi_{n-1}] \\ &= \{\text{Assumption and equation (1.2)}\} = e_{\pi_n}(x_n) a_{\pi_{n-1}\pi_n} \prod_{i=1}^{n-1} a_{\pi_{i-1}\pi_i} e_{\pi_i}(x_i) \\ &= \prod_{i=1}^{n} a_{\pi_{i-1}\pi_i} e_{\pi_i}(x_i) \end{aligned}$$

■

Application:
What can a HMM be used for in bioinformatics? One common application is to determine whether a given protein sequence $x = x_1, ..., x_n$ belongs to a certain family or not. Given a HMM F that represents the protein family and a threshold t (determines whether we include the sequence or not), we can answer the question by calculating $Pr[x_1, ..., x_n | F] > t$?

One way to calculate $Pr[x_1, ..., x_n]$ is to use $Pr[x_1, ..., x_n, \pi_1, ..., \pi_n]$ and sum over all $\pi_1, ..., \pi_n$ to get the marginal distribution

$$Pr[x_1, ..., x_n] = \sum_{\pi_1, ..., \pi_n \in Q} Pr[x_1, ..., x_n, \pi_1, ..., \pi_n]$$

However, this sum contains exponentially many terms $\mathcal{O}(|Q|^n)$.

One way to solve this is by using dynamic programming.

Claim 2. *$Pr[x_1, ..., x_n]$ can be computed in time $\mathcal{O}(|Q|^2 n)$.*

Proof. We use dynamic programming to prove this claim.

Let

$$f_\pi(i) = Pr[x_1, ..., x_i, \pi_i = \pi]$$

We now set up a recursion for $f_\pi(i)$

$$\begin{aligned}
f_\pi(i) &= Pr[x_1, ..., x_i, \pi_i = \pi] \\
&= \sum_{\pi' \in Q} Pr[x_1, ..., x_i, \pi_i = \pi | x_1, ..., x_{i-1}, \pi_{i-1} = \pi'] Pr[x_1, ..., x_{i-1}, \pi_{i-1} = \pi'] \\
&= \sum_{\pi' \in Q} Pr[x_i, \pi_i = \pi | \pi_{i-1} = \pi'] f_{\pi'}(i-1) \\
&= \sum_{\pi' \in Q} e_\pi(x_i) a_{\pi', \pi} f_{\pi'}(i-1) \\
&= e_\pi(x_i) \sum_{\pi' \in Q} a_{\pi', \pi} f_{\pi'}(i-1)
\end{aligned}$$

Base cases are

$$f_{q_I}(0) = 1$$

and

$$f_q(0) = 0, \quad \forall q \neq q_I \in Q$$

We have $\mathcal{O}(n|Q|)$ elements of f to calculate and each f takes $\mathcal{O}(|Q|)$ to calculate, therefore the time complexity is $\mathcal{O}(|Q|^2 n)$.
The final result will be in $f_{q_T}(n)$. ■

We now turn to the problem of finding a sequence of states that maximizes the probability of a given sequence of symbols, i.e. the sequence of states that is most likely to have generated the sequence of symbols, given that it was generated by the HMM. More precisely:

Problem
Given a HMM

$$M = (\Sigma,\ Q,\ q_I,\ q_T,\ A,\ E)$$

and a sequence of symbols

$$x_1, \ldots, x_n, \quad x_i \in \Sigma$$

find

$$\max_{\pi_1, \ldots, \pi_n \in Q} Pr[x_1, \ldots, x_n | \pi_1, \ldots, \pi_n]$$

and argmax i.e. a sequence of states that generates the maximum probability.

Claim 3. *This can be done in time $\mathcal{O}(|Q|^2 n)$ by dynamic programming (called Viterbi algorithm).*

Proof. :
Let

$$V_\pi(i) = \max_{\pi_1, \ldots, \pi_{i-1} \in Q} Pr[x_1, \ldots, x_i, \pi_1, \ldots, \pi_{i-1}, \pi_i = \pi]$$

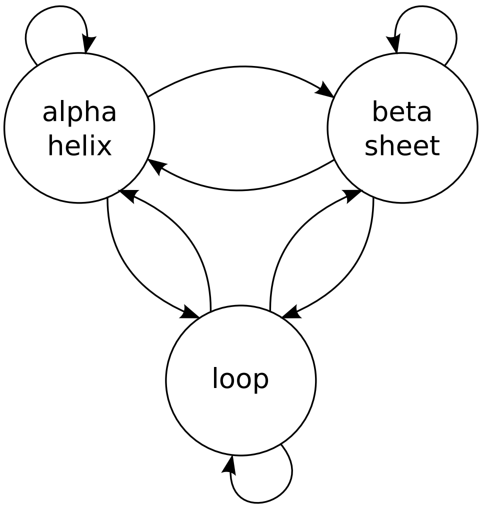

Figure 1.2: A possible HMM for the prediction of the secondary structure of a protein. The observed sequence is the actual protein sequence. The goal of this model would be to determine whether each of the amino acids in the protein sequence belongs to an α-helix, a β-sheet or a loop sequence (amino acids between structural components).

Recursion

$$\begin{aligned}
V_\pi(i) &= \max_{\pi_1,\ldots,\pi_{i-1}\in Q} Pr[x_1,\ldots,x_i,\pi_1,\ldots,\pi_{i-1},\pi_i=\pi] \\
&= \max_{\pi'} \max_{\pi_1,\ldots,\pi_{i-2}\in Q} Pr[x_1,\ldots,x_i,\pi_1,\ldots,\pi_{i-2},\pi_{i-1}=\pi',\pi_i=\pi] \\
&= \max_{\pi'} \max_{\pi_1,\ldots,\pi_{i-2}\in Q} Pr[x_i,\pi_i=\pi|\pi_{i-1}=\pi']Pr[x_1,\ldots,x_{i-1},\pi_1,\ldots,\pi_{i-2},\pi_{i-1}=\pi'] \\
&= e_\pi(x_i)\max_{\pi'} a_{\pi'\pi}V_{\pi'}(i-1)
\end{aligned}$$

Base cases

$$V_{q_I}(0)=1$$

and

$$V_q(0)=0, \quad \forall q \neq q_I \in Q$$

Each $V_q(i)$ takes time $\mathcal{O}(|Q|)$ and there are $\mathcal{O}(|Q|n)$ $V_q(i)$ to compute, so time is $\mathcal{O}(|Q|^2 n)$. ■