Statistical Computing

Hidden Markov Models for Bioinformatics

- Part III -

Uwe Menzel, 2011

uwe.menzel@matstat.org

www.matstat.org

Contents Part III

Decoding: The Viterbi algorithm

We have seen that a Hidden Markov Model consists of a state path $\{\pi_i\}$ which is not visible to the observer, and of visible symbols $\{x_i\}$ that have been emitted by the states with some probability $e_{\pi_i}(x_i)$.

A typical task connected with Hidden Markov Models is to identify the state path giving rise to the observed data. Uncovering the state path in Hidden Markov Models is often called decoding.

A common and relatively simple method to fulfill that task is the Viterbi algorithm. This algorithm attempts to find the most probable path given the observed data:

$$\pi^* = argmax_\pi \, P(x, \pi) = argmax_\pi \, P(\pi \mid x)$$

The algorithm is relatively simple because π^* can be calculated in a recursive manner, essentially reducing the computational load.

Decoding: finding hidden states from visible observations

The following **example** is taken from the book: **Durbin** et al. (Ed): Biological Sequence Analysis, Cambridge University Press, 1998

- Observed sequence (”emissions”):
  - C G C G
  - might have been generated by many state paths:
  - $C^+ G^+ C^+ G^+$
  - $C^- G^- C^- G^-$
  - $C^+ G^- C^+ G^-$
  - … and 13 more (we have $2^4 = 16$ possible state paths which can lead to this observation)
- How to find the ”best” state path ?
  - the best path π^* is the path that maximizes $P(x, \pi)$ →
  - $\pi^* = argmax_\pi \, P(x, \pi)$
  - practically not possible to calculate $P(x, \pi)$ for all possible paths…
  - → **Viterbi – algorithm** (”dynamic programming”)

Transition probabilities
(see table part II)

Observed sequence (”emissions”): **C G C G**

Some **transition probabilities** (based on the table presented in part II):

$p = 0.95$ (stay in ”+”) ; $q = 0.99$ (stay in ”-”)

$a_{C^+G^+} = 0.274 \cdot 0.95 = 0.26$

$a_{G^+C^+} = 0.339 \cdot 0.95 = 0.322$

$a_{C^-G^-} = 0.078 \cdot 0.99 = 0.0772$

$a_{G^-C^-} = 0.246 \cdot 0.99 = 0.2435$

$a_{C^+G^-} = (1 - 0.95)/4 = 0.0125$ small, switches from CpG island to non-island

$a_{G^-C^+} = (1 - 0.99)/4 = 0.0025$ small, switches from non-island to CpG island

$a_{\mathcal{B}C^+} = 0.13$ (just half the probability that a C occurs)

$a_{\mathcal{B}C^-} = 0.13$ (just half the probability that a C occurs)

Note: transitions from the begin state to state π_i will be denoted $a_{\mathcal{B}\pi_i}$ or $a_{0\pi_i}$.

Probability of the chain for some state paths:

Observed sequence:
C G C G

$$P(x,\pi) = a_{B\pi_1} \cdot \prod_{i=1}^{L} e_{\pi_i}(x_i) \cdot a_{\pi_i \pi_{i+1}}$$

$P(X = C,G,C,G \mid \pi = C^+,G^+,C^+,G^+)$ state path 1, completely in island

$= a_{BC^+} \cdot e_{C^+}(C) \cdot a_{C^+G^+} \cdot e_{G^+}(G) \cdot a_{G^+C^+} \cdot e_{C^+}(C) \cdot a_{C^+G^+} \cdot e_{G^+}(G) \cdot a_{G^+0}$

$= 0.13 \cdot 1 \cdot 0.26 \cdot 1 \cdot 0.322 \cdot 1 \cdot 0.26 \cdot 1 \cdot 1 = 2.83 \cdot 10^{-3}$

$P(X = C,G,C,G \mid \pi = C^-,G^-,C^-,G^-)$ state path 2, completely outside island

$= a_{BC^-} \cdot e_{C^-}(C) \cdot a_{C^-G^-} \cdot e_{G^-}(G) \cdot a_{G^-C^-} \cdot e_{C^-}(C) \cdot a_{C^-G^-} \cdot e_{G^-}(G) \cdot a_{G^-0}$

$= 0.13 \cdot 1 \cdot 0.0772 \cdot 1 \cdot 0.2435 \cdot 1 \cdot 0.0772 \cdot 1 \cdot 1 = 1.89 \cdot 10^{-4}$

$P(X = C,G,C,G \mid \pi = C^-,G^-,C^+,G^+)$ state path 3, border non-island → island

$= a_{BC^-} \cdot e_{C^-}(C) \cdot a_{C^-G^-} \cdot e_{G^-}(G) \cdot a_{G^-C^+} \cdot e_{C^+}(C) \cdot a_{C^+G^+} \cdot e_{G^+}(G) \cdot a_{G^+0}$

$= 0.13 \cdot 1 \cdot 0.0772 \cdot 1 \cdot 0.0025 \cdot 1 \cdot 0.26 \cdot 1 \cdot 1 = 6.52 \cdot 10^{-6}$

13 more state paths to consider to find the most probable one ! (?)

Viterbi algorithm

The task is to calculate the state path (π^*) which yields the highest probability for the chain, given the observed data:

$$\pi^* = argmax_\pi \, P(x, \pi) = argmax_\pi \, P(\pi \mid x)$$

A recursion to calculate π^* can be found in the following way:

Let us assume that we have the state path leading to the highest probability of the chain up to observation x_{i-1} with the constraint that the path ends in state k, i.e. we fix $\pi_{i-1} = k$. Let us call this probability $v_k(i-1)$:

$$v_k(i-1) = max_{\pi_1, \pi_2, \ldots, \pi_{i-2}} P(x_1, x_2, \ldots, x_{i-1}, \pi_1, \pi_2, \ldots, \pi_{i-2}, \pi_{i-1} = k)$$

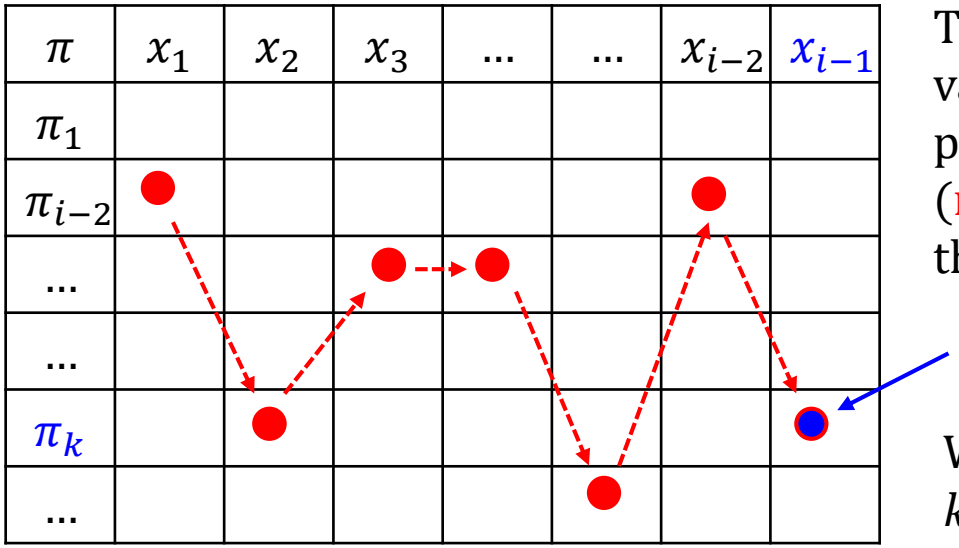
We calculate this probability for all states k, so that we have the most likely path yielding $x_1, x_2 \ldots x_{i-1}$ for all possible end states π_{i-1}. (That means that index k runs through all states, yielding $v_{A^+}(i-1), v_{C^+}(i-1), v_{G^+}(i-1)$, etc.).

Now, we can obtain the most probable path to state l at position i by finding the maximum of $v_k(i-1) \cdot a_{kl}$ w.r.t. index k. If we then multiply with the emmision probability $e_l(x_i)$, we have $v_l(i)$, the most probable path up to x_i ending in state l:

$$v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$$

Viterbi algorithm

$$v_k(i-1) = max_{\pi_1,\pi_2,\dots,\pi_{i-2}} P(x_1, x_2, \dots, x_{i-1}, \pi_1, \pi_2, \dots, \pi_{i-2}, \pi_{i-1} = k)$$



The state π_{i-1} behind observation x_{i-1} is fixed (blue), the path leading to that point (red) is chosen to maximize the probability up to x_{i-1}.

$\pi_{i-1} = k$

We calculate the optimal path for all k, i.e. for all cells in the last column.

The most probable path up to x_i ending in state l is then by recursion:

$$v_l(i) = e_l(x_i) \cdot max_k\{v_k(i-1) \cdot a_{kl}\}$$

which means that we succeeded to proceed one step in the recursive scheme.

Viterbi algorithm

Viterbi algorithm: the probabilities $v_l(i)$ can be calculated iteratively:

$$\boxed{v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}} \qquad i = 1, 2, \ldots, L$$

The recursion is initialised by setting the begin state:

$$v_0(0) = 1 \,;\, v_k(0) = 0 \text{ for } k \neq 0$$

By choosing the maximum of all $v_l(L)$ at the last position of the chain ($i = L$), we have identified the most probable chain. It must be recorded for each i which state switches to which one. This makes it then possible to backtrace the most probable state path leading to this chain. The example below demonstrates a complete recursion for a very short sequence, using the transition- and emission probabilities defined in part 2.

The following **example** is taken from the book: **Durbin** et al. (Ed): Biological Sequence Analysis, Cambridge University Press, 1998

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

$i = 1 \rightarrow v_l(1) = e_l(x_1) \cdot max_k \{v_k(0) \cdot a_{kl}\} = e_l(C) \cdot max_k \{v_k(0) \cdot a_{kl}\}$

- the indices k and l run through all states: $A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-$
- the begin state $\mathcal{B}$ must be included for $i = 1$: $v_{\mathcal{B}}(0) = 1$, all other $v_k(0) = 0$

k runs through all symbols (A^+, C^+, G^+, ...) to find the maximum of the product

$$l = A^+ \rightarrow v_{A^+}(1) = e_{A^+}(C) \cdot max_k\{v_k(0) \cdot a_{kA^+}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^+} = 0$$
$$l = C^+ \rightarrow v_{C^+}(1) = e_{C^+}(C) \cdot max_k\{v_k(0) \cdot a_{kC^+}\} = 1 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^+} = 1 \cdot 1 \cdot 0.13 = 0.13$$
$$l = G^+ \rightarrow v_{G^+}(1) = e_{G^+}(C) \cdot max_k\{v_k(0) \cdot a_{kG^+}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^+} = 0$$
$$l = T^+ \rightarrow v_{T^+}(1) = e_{T^+}(C) \cdot max_k\{v_k(0) \cdot a_{kT^+}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^+} = 0$$
$$l = A^- \rightarrow v_{A^-}(1) = e_{A^-}(C) \cdot max_k\{v_k(0) \cdot a_{kA^-}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^-} = 0$$
$$l = C^- \rightarrow v_{C^-}(1) = e_{C^-}(C) \cdot max_k\{v_k(0) \cdot a_{kC^-}\} = 1 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^-} = 1 \cdot 1 \cdot 0.13 = 0.13$$
$$l = G^- \rightarrow v_{G^-}(1) = e_{G^-}(C) \cdot max_k\{v_k(0) \cdot a_{kG^-}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^-} = 0$$
$$l = T^- \rightarrow v_{T^-}(1) = e_{T^-}(C) \cdot max_k\{v_k(0) \cdot a_{kT^-}\} = 0 \cdot v_{\mathcal{B}}(0) \cdot a_{\mathcal{B}C^-} = 0$$

The begin state switches to C^+ or C^-, with equal probability. It is important to remember which transitions have been made going from $i-1$ to i.

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

The obtained results can be arranged in a table:

state l	start: $v_l(0)$	$v_l(1)$
$\mathcal{B}$	1	0
A^+	0	0
C^+	0	0.13
G^+	0	0
T^+	0	0
A^-	0	0
C^-	0	0.13
G^-	0	0
T^-	0	0

Observed sequence:
C G C G

$x_0 = \mathcal{B}$ (begin state)
$x_1 = C$
$x_2 = G$
$x_3 = C$
$x_4 = G$

It is important to remember what transition was made at each iteration.

arrows indicate the transition leading to $v_l(i)$

Example from: Durbin et al. (ed.), **Biological sequence analysis**, Chapter 3

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

$i = 2 \rightarrow v_l(2) = e_l(x_2) \cdot max_k \{v_k(1) \cdot a_{kl}\} = e_l(G) \cdot max_k \{v_k(1) \cdot a_{kl}\}$

- the indices k and l run through all states: $A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-$
- we had: $v_{C^+}(1) = 0.13$ and $v_{C^-}(1) = 0.13$; all other $v_l(1) = 0$

k runs through all symbols (A^+, C^+, G^+, ...) to find the maximum of the product

$l = A^+ \rightarrow v_{A^+}(2) = e_{A^+}(G) \cdot max_k\{v_k(1) \cdot a_{kA^+}\} = 0 \cdot max_k\{\dots\} = 0$
$l = C^+ \rightarrow v_{C^+}(2) = e_{C^+}(G) \cdot max_k\{v_k(1) \cdot a_{kC^+}\} = 0 \cdot max_k\{\dots\} = 0$
$l = G^+ \rightarrow v_{G^+}(2) = e_{G^+}(G) \cdot max_k\{v_k(1) \cdot a_{kG^+}\} = 1 \cdot 0.0338 = 0.0338$ (see next page)
$l = T^+ \rightarrow v_{T^+}(2) = e_{T^+}(G) \cdot max_k\{v_k(1) \cdot a_{kT^+}\} = 0 \cdot max_k\{\dots\} = 0$
$l = A^- \rightarrow v_{A^-}(2) = e_{A^-}(G) \cdot max_k\{v_k(1) \cdot a_{kA^-}\} = 0 \cdot max_k\{\dots\} = 0$
$l = C^- \rightarrow v_{C^-}(2) = e_{C^-}(G) \cdot max_k\{v_k(1) \cdot a_{kC^-}\} = 0 \cdot max_k\{\dots\} = 0$
$l = G^- \rightarrow v_{G^-}(2) = e_{G^-}(G) \cdot max_k\{v_k(1) \cdot a_{kG^-}\} = 1 \cdot 0.01 = 0.01$ (see next page)
$l = T^- \rightarrow v_{T^-}(2) = e_{T^-}(G) \cdot max_k\{v_k(1) \cdot a_{kT^-}\} = 0 \cdot max_k\{\dots\} = 0$

These results can be recorded in the next column of the table →

Auxiliary calculation: $max_k\{v_k(1) \cdot a_{kG^+}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had: $v_{C^+}(1) = 0.13$ and $v_{C^-}(1) = 0.13$; all other $v_l(1) = 0$

k	$v_k(1) \cdot a_{kG^+}$	$v_k(1) \cdot a_{kG^+}$
A^+	$0 \cdot a_{A^+G^+}$	0
C^+	$0.13 \cdot a_{C^+G^+}$	$0.13 \cdot 0.26 = 0.0338$
G^+	$0 \cdot a_{G^+G^+}$	0
T^+	$0 \cdot a_{T^+G^+}$	0
A^-	$0 \cdot a_{A^-G^+}$	0
C^-	$0.13 \cdot a_{C^-G^+}$	$0.13 \cdot 0.0025 = 0.000325$
G^-	$0 \cdot a_{G^-G^+}$	0
T^-	$0 \cdot a_{T^-G^+}$	0

maximum = transition C^+G^+

$\Rightarrow$ $max_k\{v_k(1) \cdot a_{kG^+}\} = 0.0338$, by switching from C^+ to G^+

Auxiliary calculation: $max_k\{v_k(1) \cdot a_{kG^-}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had: $v_{C^+}(1) = 0.13$ and $v_{C^-}(1) = 0.13$; all other $v_l(1) = 0$

k	$v_k(1) \cdot a_{kG^-}$	$v_k(1) \cdot a_{kG^-}$
A^+	$0 \cdot a_{A^+G^-}$	0
C^+	$0.13 \cdot a_{C^+G^-}$	$0.13 \cdot 0.0125 = 0.001625$
G^+	$0 \cdot a_{G^+G^-}$	0
T^+	$0 \cdot a_{T^+G^-}$	0
A^-	$0 \cdot a_{A^-G^-}$	0
C^-	$0.13 \cdot a_{C^-G^-}$	$0.13 \cdot 0.077 = 0.01001$
G^-	$0 \cdot a_{G^-G^-}$	0
T^-	$0 \cdot a_{T^-G^-}$	0

maximum = transition C^-G^-

$\Rightarrow$ $max_k\{v_k(1) \cdot a_{kG^-}\} = 0.01001$, by switching from C^- to G^-

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

We calculated $v_{G^+}(2) = 0.0338$ and $v_{G^-}(2) = 0.01001$; all other $v_l(2) = 0$

state l	start: $v_l(0)$	$v_l(1)$	$v_l(2)$
$\mathcal{B}$	1	0	
A^+	0	0	
C^+	0	0.13	
G^+	0	0	0.0338
T^+	0	0	
A^-	0	0	
C^-	0	0.13	
G^-	0	0	0.01001
T^-	0	0	

Observed sequence:
C G C G

$x_0 = \mathcal{B}$ (begin state)
$x_1 = C$
$x_2 = G$
$x_3 = C$
$x_4 = G$

arrows indicate the transition leading to maximum $v_l(i)$

Example from: Durbin et al. (ed.), **Biological sequence analysis**, Chapter 3

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

$i = 3 \rightarrow v_l(3) = e_l(x_3) \cdot max_k \{v_k(2) \cdot a_{kl}\} = e_l(C) \cdot max_k \{v_k(2) \cdot a_{kl}\}$

- the indices k and l run through all states: $A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-$
- we had $v_{G^+}(2) = 0.0338$ and $v_{G^-}(2) = 0.01001$; all other $v_l(2) = 0$

k runs through all symbols ($A^+, C^+, G^+, \ldots$) to find the maximum of the product

$l = A^+ \rightarrow v_{A^+}(3) = e_{A^+}(C) \cdot max_k\{v_k(2) \cdot a_{kA^+}\} = 0 \cdot max_k\{\ldots\} = 0$
$l = C^+ \rightarrow v_{C^+}(3) = e_{C^+}(C) \cdot max_k\{v_k(2) \cdot a_{kC^+}\} = 1 \cdot 0.011 = 0.0108836$ (next page)
$l = G^+ \rightarrow v_{G^+}(3) = e_{G^+}(C) \cdot max_k\{v_k(2) \cdot a_{kG^+}\} = 0 \cdot max_k\{\ldots\} = 0$
$l = T^+ \rightarrow v_{T^+}(3) = e_{T^+}(C) \cdot max_k\{v_k(2) \cdot a_{kT^+}\} = 0 \cdot max_k\{\ldots\} = 0$
$l = A^- \rightarrow v_{A^-}(3) = e_{A^-}(C) \cdot max_k\{v_k(2) \cdot a_{kA^-}\} = 0 \cdot max_k\{\ldots\} = 0$
$l = C^- \rightarrow v_{C^-}(3) = e_{C^-}(C) \cdot max_k\{v_k(2) \cdot a_{kC^-}\} = 1 \cdot 0.00244 = 0.00244244$ (next page)
$l = G^- \rightarrow v_{G^-}(3) = e_{G^-}(C) \cdot max_k\{v_k(2) \cdot a_{kG^-}\} = 0 \cdot max_k\{\ldots\} = 0$
$l = T^- \rightarrow v_{T^-}(3) = e_{T^-}(C) \cdot max_k\{v_k(2) \cdot a_{kT^-}\} = 0 \cdot max_k\{\ldots\} = 0$

These results can be recorded in the next column of the table →

Auxiliary calculation: $max_k\{v_k(2) \cdot a_{kC^+}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had $v_{G^+}(2) = 0.0338$ and $v_{G^-}(2) = 0.01001$; all other $v_l(2) = 0$

k	$v_k(2) \cdot a_{kC^+}$	$v_k(2) \cdot a_{kC^+}$
A^+	$0 \cdot a_{A^+C^+}$	0
C^+	$0 \cdot a_{C^+C^+}$	0
G^+	$0.034 \cdot a_{G^+C^+}$	$0.0338 \cdot 0.322 = 0.0108836$
T^+	$0 \cdot a_{T^+C^+}$	0
A^-	$0 \cdot a_{A^-C^+}$	0
C^-	$0 \cdot a_{C^-C^+}$	0
G^-	$0.01 \cdot a_{G^-C^+}$	$0.01 \cdot 0.0025 = 0.000025$
T^-	$0 \cdot a_{T^-C^+}$	0

maximum = transition G^+C^+

$\Rightarrow$ $max_k\{v_k(2) \cdot a_{kC^+}\} = 0.0108836$, by transition from G^+ to C^+

Auxiliary calculation: $max_k\{v_k(2) \cdot a_{kC^-}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had $v_{G^+}(2) = 0.034$ and $v_{G^-}(2) = 0.01$; all other $v_l(2) = 0$

k	$v_k(2) \cdot a_{kC^-}$	$v_k(2) \cdot a_{kC^-}$
A^+	$0 \cdot a_{A^+C^-}$	0
C^+	$0 \cdot a_{C^+C^-}$	0
G^+	$0.034 \cdot a_{G^+C^-}$	$0.034 \cdot 0.0125 = 0.000425$
T^+	$0 \cdot a_{T^+C^-}$	0
A^-	$0 \cdot a_{A^-C^-}$	0
C^-	$0 \cdot a_{C^-C^-}$	0
G^-	$0.01001 \cdot a_{G^-C^-}$	$0.01001 \cdot 0.244 = 0.00244244$
T^-	$0 \cdot a_{T^-C^-}$	0

maximum = transition G^-C^-

$\Rightarrow$ $max_k\{v_k(2) \cdot a_{kC^-}\} = 0.00244244$, by switching from G^- to C^-

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

We had $v_{C^+}(3) = 0.0108836$ and $v_{C^-}(3) = 0.00244244$; all other $v_l(3) = 0$

state l	start: $v_l(0)$	$v_l(1)$	$v_l(2)$	$v_l(3)$
$\mathcal{B}$	1	0	0	0
A^+	0	0	0	0
C^+	0	0.13	0	0.0109
G^+	0	0	0.034	0
T^+	0	0	0	0
A^-	0	0	0	0
C^-	0	0.13	0	0.00244
G^-	0	0	0.01	0
T^-	0	0	0	0

Observed sequence:
C G C G

$x_0 = \mathcal{B}$ (begin state)
$x_1 = C$
$x_2 = G$
$x_3 = C$
$x_4 = G$

arrows indicate the transition leading to maximum $v_l(i)$

Example from: Durbin et al. (ed.), **Biological sequence analysis**, Chapter 3

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

$$i = 4 \rightarrow v_l(4) = e_l(x_4) \cdot max_k \{v_k(3) \cdot a_{kl}\} = e_l(G) \cdot max_k \{v_k(3) \cdot a_{kl}\}$$

- the indices k and l run through all states: $A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-$
- we had $v_{C^+}(3) = 0.0108836$ and $v_{C^-}(3) = 0.00244244$; all other $v_l(3) = 0$

k runs through all symbols ($A^+, C^+, G^+, \ldots$) to find the maximum of the product

$$l = A^+ \rightarrow v_{A^+}(4) = e_{A^+}(G) \cdot max_k\{v_k(3) \cdot a_{kA^+}\} = 0 \cdot max_k\{\ldots\} = 0$$
$$l = C^+ \rightarrow v_{C^+}(4) = e_{C^+}(G) \cdot max_k\{v_k(3) \cdot a_{kC^+}\} = 0 \cdot max_k\{\ldots\} = 0$$
$$l = G^+ \rightarrow v_{G^+}(4) = e_{G^+}(G) \cdot max_k\{v_k(3) \cdot a_{kG^+}\} = 1 \cdot 0.0028297 = 0.0028297 \text{ (next page)}$$
$$l = T^+ \rightarrow v_{T^+}(4) = e_{T^+}(G) \cdot max_k\{v_k(3) \cdot a_{kT^+}\} = 0 \cdot max_k\{\ldots\} = 0$$
$$l = A^- \rightarrow v_{A^-}(4) = e_{A^-}(G) \cdot max_k\{v_k(3) \cdot a_{kA^-}\} = 0 \cdot max_k\{\ldots\} = 0$$
$$l = C^- \rightarrow v_{C^-}(4) = e_{C^-}(G) \cdot max_k\{v_k(3) \cdot a_{kC^-}\} = 0 \cdot max_k\{\ldots\} = 0$$
$$l = G^- \rightarrow v_{G^-}(4) = e_{G^-}(G) \cdot max_k\{v_k(3) \cdot a_{kG^-}\} = 1 \cdot 1.88068 \cdot 10^{-4} = 1.88068 \cdot 10^{-4}$$
$$l = T^- \rightarrow v_{T^-}(4) = e_{T^-}(G) \cdot max_k\{v_k(3) \cdot a_{kT^-}\} = 0 \cdot max_k\{\ldots\} = 0$$

These results can be recorded in the next column of the table →

Auxiliary calculation: $max_k\{\nu_k(3) \cdot a_{kG^+}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had $\nu_{C^+}(3) = 0.0108836$ and $\nu_{C^-}(3) = 0.00244244$; all other $\nu_l(3) = 0$

k	$\nu_k(3) \cdot a_{kG^+}$	$\nu_k(3) \cdot a_{kG^+}$
A^+	$0 \cdot a_{A^+G^+}$	0
C^+	$0.011 \cdot a_{C^+G^+}$	$0.0108836 \cdot 0.26 = 0.002829736$
G^+	$0 \cdot a_{G^+G^+}$	0
T^+	$0 \cdot a_{T^+G^+}$	0
A^-	$0 \cdot a_{A^-G^+}$	0
C^-	$0.00244 \cdot a_{C^-G^+}$	$0.00244244 \cdot 0.0025 = 6 \cdot 10^{-6}$
G^-	$0 \cdot a_{G^-G^+}$	0
T^-	$0 \cdot a_{T^-G^+}$	0

maximum = transition C^+G^+

$\Rightarrow$ $max_k\{\nu_k(3) \cdot a_{kG^+}\} = 0.002829736$, by switching from C^+ to G^+

Auxiliary calculation: $max_k\{\nu_k(3) \cdot a_{kG^-}\}$

$k = \{A^+, C^+, G^+, T^+, A^-, C^-, G^-, T^-\}$

we had $\nu_{C^+}(3) = 0.0108836$ and $\nu_{C^-}(3) = 0.00244244$; all other $\nu_l(3) = 0$

k	$\nu_k(3) \cdot a_{kG^-}$	$\nu_k(3) \cdot a_{kG^-}$
A^+	$0 \cdot a_{A^+G^-}$	0
C^+	$0.011 \cdot a_{C^+G^-}$	$0.011 \cdot 0.0125 = 1.375 \cdot 10^{-4}$
G^+	$0 \cdot a_{G^+G^-}$	0
T^+	$0 \cdot a_{T^+G^-}$	0
A^-	$0 \cdot a_{A^-G^-}$	0
C^-	$0.00244 \cdot a_{C^-G^-}$	$0.00244244 \cdot 0.077$ $= 1.880679 \cdot 10^{-4}$
G^-	$0 \cdot a_{G^-G^-}$	0
T^-	$0 \cdot a_{T^-G^-}$	0

maximum = transition C^-G^-

$\Rightarrow$ $max_k\{\nu_k(3) \cdot a_{kG^-}\} = 1.88068 \cdot 10^{-4}$, by switching from C^- to G^-

Recursion: $v_l(i) = e_l(x_i) \cdot max_k \{v_k(i-1) \cdot a_{kl}\}$
Observed sequence: **C G C G**
$x_0 = \mathcal{B}$ (begin state) ; $x_1 = C$; $x_2 = G$; $x_3 = C$; $x_4 = G$

We had $v_{G^+}(4) = 0.0028297$ and $v_{G^-}(4) = 1.8807 \cdot 10^{-4}$; all other $v_l(4) = 0$

state l	start: $v_l(0)$	$v_l(1)$	$v_l(2)$	$v_l(3)$	$v_l(4)$
$\mathcal{B}$	1	0	0	0	0
A^+	0	0	0	0	0
C^+	0	0.13	0	0.011	0
G^+	0	0	0.034	0	0.00283
T^+	0	0	0	0	0
A^-	0	0	0	0	0
C^-	0	0.13	0	0.00244	0
G^-	0	0	0.01	0	$1.88 \cdot 10^{-4}$
T^-	0	0	0	0	0

Observed:
C G C G

$x_0 = \mathcal{B}$ (begin)
$x_1 = C$
$x_2 = G$
$x_3 = C$
$x_4 = G$

Viterby algorithm

state l	start: $\nu_l(0)$	$\nu_l(1)$	$\nu_l(2)$	$\nu_l(3)$	$\nu_l(4)$
$\mathcal{B}$	1	0	0	0	0
A^+	0	0	0	0	0
C^+	0	0.13	0	0.011	0
G^+	0	0	0.034	0	**0.00283**
T^+	0	0	0	0	0
A^-	0	0	0	0	0
C^-	0	0.13	0	0.00244	0
G^-	0	0	0.01	0	$1.88 \cdot 10^{-4}$
T^-	0	0	0	0	0

Observed:
C G C G

$x_0 = \mathcal{B}$ (begin)
$x_1 = C$
$x_2 = G$
$x_3 = C$
$x_4 = G$

The largest $\nu_l(4)$, 0.00283, is the probability of the most probable chain. Backtracking the state path leading to that chain identifies the path yielding to maximum $P(x, \pi)$. Here, we find the path $\mathcal{B}\ C^+G^+C^+G^+$, only including "+"-states → we found that the sequence originates from a CpG island.

Viterby algorithm

C:\Users\Uwe\Desktop\TALKS_POSTERS\LECTURES\HMM-Talk am HKI\HMM_Viterbi_CGCG.R - R Editor

```
library(HMM)

states = c("A+", "C+", "G+", "T+", "A-", "C-", "G-", "T-")
symbols = c("A", "C", "G", "T")

trans_prob = get(load("trans_prob_HMM.RData"))
emission_prob = get(load("emission_prob_HMM.RData"))

start_probs = c(0.12, 0.13, 0.13, 0.12, 0.12, 0.13, 0.13, 0.12)
names(start_probs) = c("BA+", "BC+", "BG+", "BT+", "BA-", "BC-", "BG-", "BT-")
start_probs
#  BA+  BC+  BG+  BT+  BA-  BC-  BG-  BT-
# 0.12 0.13 0.13 0.12 0.12 0.13 0.13 0.12

hmm = initHMM(states, symbols, startProbs = start_probs,
     transProbs = trans_prob, emissionProbs = emission_prob)

observation = c("C", "G", "C", "G")    # observed data

vit = viterbi(hmm, observation)        # decoding
vit  # "C+" "G+" "C+" "G+"
```

CpG island example

same result

- the files trans_prob_HMM.Rdata and emission_prob_HMM.Rdata are linked on matstat.org

Viterby algorithm

C:\Users\Uwe\Desktop\TALKS_POSTERS\LECTURES\HMM-Talk am HKI\HMM_Viterbi2_CGCG.R - R Editor

```
library(HMM)

states = c("A+", "C+", "G+", "T+", "A-", "C-", "G-", "T-")
symbols = c("A", "C", "G", "T")

trans_prob = get(load("trans_prob_HMM.RData"))
emission_prob = get(load("emission_prob_HMM.RData"))
start_probs = c(0.12, 0.13, 0.13, 0.12, 0.12, 0.13, 0.13, 0.12)
names(start_probs) = c("BA+", "BC+", "BG+", "BT+", "BA-", "BC-", "BG-", "BT-")

hmm = initHMM(states, symbols, startProbs = start_probs,
     transProbs = trans_prob, emissionProbs = emission_prob)

observation = c("C", "G", "C", "G")    # observed data

source("viterbi2.R")  # slightly changed function

vit2 = viterbi2(hmm, observation)  # decoding
exp(vit2$vmatrix)
# states    1       2          3            4
#    A+ 0.00 0.00000 0.00000000 0.0000000000
#    C+ 0.13 0.00000 0.01088360 0.0000000000
#    G+ 0.00 0.03380 0.00000000 0.0028297360
#    T+ 0.00 0.00000 0.00000000 0.0000000000
#    A- 0.00 0.00000 0.00000000 0.0000000000
#    C- 0.13 0.00000 0.00244244 0.0000000000
#    G- 0.00 0.01001 0.00000000 0.0001880679
#    T- 0.00 0.00000 0.00000000 0.0000000000
```

a slightly changed function ”viterbi2.R” shows also the details of the calculation

Uwe Menzel, 2011

Viterby algorithm

C:\Users\Uwe\Desktop\TALKS_POSTERS\LECTURES\HMM-Talk am HKI\HMM_Viterbi.R - R Editor

Dishonest Casino

```
library(HMM)

states = c("Fair", "Loaded")
symbols = 1:6

trans_prob = matrix(c(0.95, 0.05, 0.1, 0.9), nrow = 2, byrow = TRUE)

emission_prob_fair = rep(1/6, 6)
emission_prob_loaded = c(rep(0.1, 5), 0.5)
emission_prob = rbind(emission_prob_fair, emission_prob_loaded)
nrow(emission_prob) == length(states) # IMPORTANT!, must be TRUE

hmm = initHMM(states, symbols, transProbs = trans_prob,
              emissionProbs = emission_prob)

## simulate observation:
fair_part = sample(1:6, 500, replace = TRUE, prob = rep(1/6, 6))
loaded_part = sample(1:6, 500, replace = TRUE, prob = c(rep(0.1, 5), 0.5))
observation = c(fair_part, loaded_part)

## Decoding, Viterbi:
viterbi = viterbi(hmm, observation)
viterbi
```

Analyzing a Hidden Markov Model

This section explains how to use functions in the Statistics Toolbox to analyze hidden Markov models. For illustration, the section uses the example described in Example of a Hidden Markov Model. The section shows how to recover information about the model, assuming that you do not know some of the model's parameters. The section covers the following topics:

MATLAB: Casino example!

Setting Up the Model and Generating Data

This section shows how to set up a hidden Markov model and use it to generate data. First, create the transition and emission matrices by entering the following commands.

```
TRANS = [.9 .1; .05 .95;];

EMIS = [1/6, 1/6, 1/6, 1/6, 1/6, 1/6;...
7/12, 1/12, 1/12, 1/12, 1/12, 1/12];
```

Next, generate a random sequence of emissions from the model, seq, of length 1000, using the function hmmgenerate. You can also return the corresponding random sequence of states in the model as the second output, states.

```
[seq, states] = hmmgenerate(1000, TRANS, EMIS);
```

Note In generating the sequences seq and states, hmmgenerate begins with the model in state $i_0 = 1$ at step 0. The model then makes a transition to state i_1 at step 1, and returns i_1 as the first entry in states.

Computing the Most Likely Sequence of States

Suppose you know the transition and emission matrices, TRANS and EMIS. If you observe a sequence, seq, of emissions, how can you compute the most likely sequence of states that generated the sequence? The function `hmmviterbi` uses the Viterbi algorithm to compute the most likely sequence of states that the model would go through to generate the given sequence of emissions.

```
likelystates = hmmviterbi(seq, TRANS, EMIS);
```

Viterby algorithm

`likelystates` is a sequence of the same length as seq.

To test the accuracy of `hmmviterbi`, you can compute the percentage of the time that the actual sequence `states` agrees with the sequence `likelystates`.

```
sum(states==likelystates)/1000

ans =

0.8200
```

MATLAB: Casino example!

This shows that the most likely sequence of states agrees with the actual sequence 82% of the time. Note that your results might differ if you run the same commands, because the sequence seq is random.

> **Note** The states at the beginning of the sequence returned by `hmmviterbi` are less reliable because of the computational delay in the Viterbi algorithm.

Analyzing a Hidden Markov Model

Estimating the Transition and Emission Matrices

Viterby algorithm: Remarks

- Result: we have found that the whole sequence CGCG is within a CpG island
- Method works for arbitrary long sequence and might then switch between long stretches of + and – states, i.e. between CpG islands and other genomic sequence
- For long sequences, it is suggested to calculate the log-probability instead of probability, in order to avoid underflow during computation. Products are replaced by sums when doing so.

Trellis-Diagramm

	$i = 0$ B	$i = 1$ x_1	$i = 2$ x_2	$i = 3$ x_3	$i = 4$ x_4	$i = 5$ ...
B	1	-	-	-	-	-
π_1	0	●	●	●	●	●
π_2	0	●	●	●	●	●
π_3	0	●	●	●	●	●
π_4	0	●	●	●	●	●
π_5	0	●	●	●	●	●