Hidden Markov Models

Some useful extensions
Recall the simple gene finding HMM

N: non-coding

π_N = 1
π_C = 0

C: coding
Recall the simple gene finding HMM

- The gene is a substring of the DNA sequence of A,C,G,T’s
- The gene starts with a start-codon \texttt{atg}
- The gene ends with a stop-codon \texttt{taa}, \texttt{tag} or \texttt{tga}
- The number of nucleotides in a gene is a multiple of 3
- The gene does not contain internal start- or stop-codons

N: non-coding

\[ \pi_N = 1 \]
\[ \pi_C = 0 \]

How do we avoid internal start- or stop-codons?

\[ A: >0 \]
\[ C: >0 \]
\[ G: >0 \]
\[ T: >0 \]
Avoiding internal start- or stop-codons

Encode the emission of each legal codon as a sequence of states. Many states (60*3=180) and transitions (60*59=3540)!
Other ideas?
The probability of emitting $x_n$ depends also on $x_{n-1}$ and $x_{n-2}$.

The basic algorithms remain the same:

$$\alpha(z_n) = p(x_n|x_{n-1}, x_{n-2}, z_n) \sum_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1})$$

$$\omega(z_n) = p(x_n|x_{n-1}, x_{n-2}, z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n|z_{n-1})$$
Autoregressive HMMs

For each state, we just have to state the conditional probabilities. For a 4-letter DNA alphabet this corresponds to 4*16 emission prob.

The probability of emitting $x_n$ depends also on $x_{n-1}$ and $x_{n-2}$.

The basic algorithms remain the same:

$$\alpha(z_n) = p(x_n|x_{n-1}, x_{n-2}, z_n) \prod_{z_{n-1}} \alpha(z_{n-1}) p(z_n|z_{n-1})$$

$$\omega(z_n) = p(x_n|x_{n-1}, x_{n-2}, z_n) \max_{z_{n-1}} \omega(z_{n-1}) p(z_n|z_{n-1})$$
Adjusting our simple HMM

N: non-coding

C: coding

\[ \pi_N = 1 \]

\[ \pi_C = 0 \]
Emitting a variable number of symbols

Make it possible to emit a variable number of symbols depending on the state. Fx when being in state $z_n$ the model emits $d_n$ symbols, where $d_n$ is an integer $\geq 0$.

The basic algorithms can easily be reformulated, fx Viterbi:
Emitting a variable number of symbols

Make it possible to emit a variable number of symbols depending on the state. Fx when being in state $z_n$ the model emits $d_n$ symbols, where $d_n$ is an integer $\geq 0$.

The basic algorithms can easily be reformulated, fx Viterbi:

$$\omega(n, j) : \text{The probability of the most likely path generating the first } n \text{ symbols and ending in state } j.$$  

$$\omega(n, j) = \max_{i \to j} \omega(n - d_j, i)p(i \to j)p(x_n \ldots x_{n-d_j+1} | j)$$
Emitting a variable number of symbols

Make it possible to emit a variable number of symbols depending on the state. Fx when being in state $z_n$ the model emits $d_n$ symbols, where $d_n$ is an integer $\geq 0$.

The basic algorithms can easily be reformulated, fx Viterbi:

$$\omega(n, j) : \text{The probability of the most likely path generating the first } n \text{ symbols and ending in state } j.$$  
$$\omega(n, j) = \max_{i \rightarrow j} \omega(n - d_j, i)p(i \rightarrow j)p(x_n \ldots x_{n-d_j+1} | j)$$

Transition prob from state $i$ to $j$  
Emission prob of emitting $d_j$ symbols from state $j$. 
Emitting a variable number of symbols

Make it possible to emit a variable number of symbols depending on the state. Fx when being in state $z_n$ the model emits $d_n$ symbols, where $d_n$ is an integer $\geq 0$.

The basic algorithms can easily be reformulated, fx Viterbi:

$$\omega(n, j) = \max_{i \rightarrow j} \omega(n - d_j, i)p(i \rightarrow j)p(x_n \ldots x_{n-d_j+1} | j)$$

Transition prob from state $i$ to $j$

Emission prob of emitting $d_j$ symbols from state $j$.

Special case: If $d_j = 0$ then the state is called a silent state.
Adjusting our simple HMM

N: non-coding

A: >0
C: >0
G: >0
T: >0

ATG: 1
???: 0

TAA: >0
TAG: >0
TGA: >0
???: 0

π_N = 1
π_C = 0

C: coding

ATG: 0
TAA: 0
TAG: 0
TGA: 0
???: >0