Hidden Markov Models and Gene Prediction

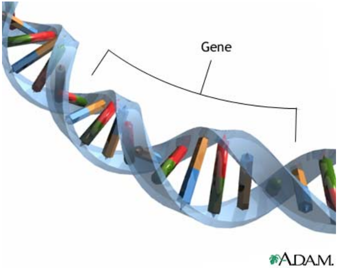
Brandon Malone

Much of this material is adapted from Chapter 15 in Russell - Norvig
Many of the images were taken from the Internet

January 30, 2014

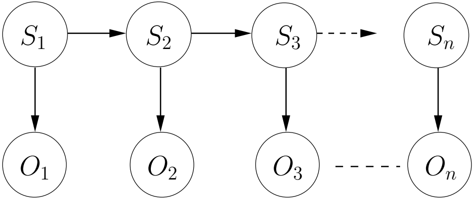
Gene Prediction

Suppose we have a long DNA sequence.



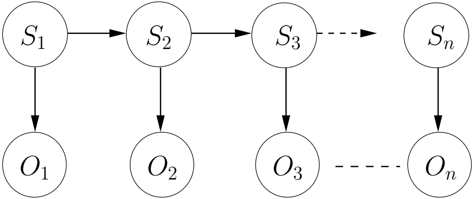
We are interested in parts of the sequence that may be genes. How can we (automatically) tell which parts may be genes?

Hidden Markov Model



What are the conditional independencies asserted by this structure?

Hidden Markov Model

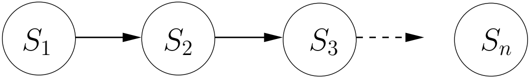


What are the conditional independencies asserted by this structure?

All of the **obervations** (O_ts) are independent, given the **state** (S_ts).

A particular state S_{t+1} is independent of all previous states given its immediate successor S_t.

Observable Markov models

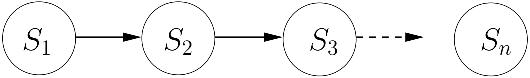


Each variable S_t corresponds to the state of the world at "time" t*.

For a **stationary first-order Markov process**, the state of the world at time $t+1$ depends only upon the state at time t.

* In our running example, "time" will actually be the position in the DNA sequence.

Observable Markov models



Each variable S_t corresponds to the state of the world at “time” t^*.

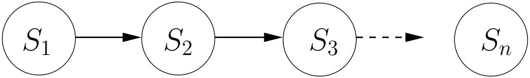
For a **stationary first-order Markov process**, the state of the world at time $t+1$ depends only upon the state at time t.

$$I(S_{t+1}, S_t, \{S_0, S_1, \ldots, S_{t-1}\})$$

* In our running example, “time” will actually be the position in the DNA sequence.

Observable Markov models - Simple class example

Given the following Markov process, calculate the probability of the follwing sequence of states: *true*, *true*, *false*, *true*, *true*.



S_0	Θ_{S_0}
T	.5
F	.5

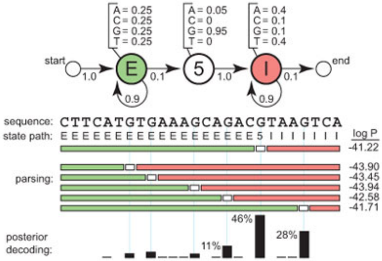
S_{t-1}	S_t	$\Theta_{S_t \mid S_{t-1}}$
T	T	.7
T	F	.3
F	T	.3
F	F	.7

Hidden Markov models

We (often) cannot directly observe if a piece of DNA is a gene or not.

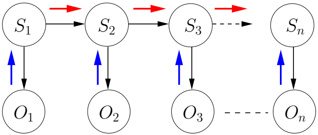
We *can* observe the DNA sequence, though.

So, given the DNA sequence, we would like to label each base as "Genic" or "Intergenic".



Inference algorithms

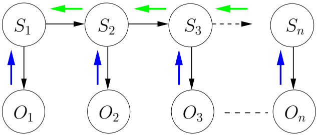
We will discuss four inference algorithms. All of the algorithms are based on the notion of **messsage passing**.



- **The forward algorithm** predicts the state in the future given current observations.
- The backward algorithm updates predictions about states in the past given more recent observations.
- The forward-backward efficiently calculates the posterior probabilities of all states given observations.
- The Viterbi algorithm calculates the most likely sequence of states to generate the observations.

Inference algorithms

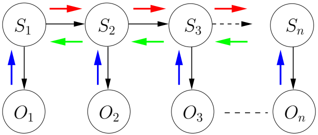
We will discuss four inference algorithms. All of the algorithms are based on the notion of **messsage passing**.



- The forward algorithm predicts the state in the future given current observations.
- **The backward algorithm** updates predictions about states in the past given more recent observations.
- The forward-backward efficiently calculates the posterior probabilities of all states given observations.
- The Viterbi algorithm calculates the most likely sequence of states to generate the observations.

Inference algorithms

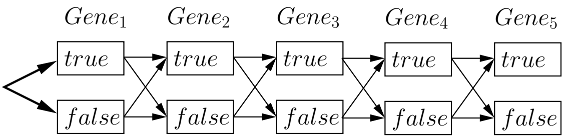
We will discuss four inference algorithms. All of the algorithms are based on the notion of **messsage passing**.



- The forward algorithm predicts the state in the future given current observations.
- The backward algorithm updates predictions about states in the past given more recent observations.
- **The forward-backward algorithm** efficiently calculates the posterior probabilities of all states given observations.
- The Viterbi algorithm calculates the most likely sequence of states to generate the observations.

Inference algorithms

We will discuss four inference algorithms. All of the algorithms are based on the notion of **messsage passing**.



- The forward algorithm predicts the state in the future given current observations.
- The backward algorithm updates predictions about states in the past given more recent observations.
- The forward-backward efficiently calculates the posterior probabilities of all states given observations.
- **The Viterbi algorithm** calculates the most likely sequence of states to generate the observations.

The forward algorithm

Problem: Given the observations up to time $t+1$, what is the posterior probability of S_{t+1}?

$$
\begin{aligned}
P(S_{t+1}|O_1,\ldots,O_{t+1}) &= \frac{P(S_{t+1},O_1,\ldots,O_t,O_{t+1})}{P(O_1,\ldots,O_{t+1})} \\
&= \frac{P(O_{t+1}|S_{t+1},O_1,\ldots O_t)P(S_{t+1}|O_1,\ldots,O_t)P(O_1,\ldots O_t)}{P(O_1,\ldots,O_{t+1})} \\
&= \frac{P(O_1,\ldots O_t)}{P(O_1,\ldots,O_{t+1})}P(O_{t+1}|S_{t+1},O_1,\ldots O_t)P(S_{t+1}|O_1,\ldots,O_t) \\
&= \frac{P(O_1,\ldots O_t)}{P(O_1,\ldots,O_{t+1})}P(O_{t+1}|S_{t+1})P(S_{t+1}|O_1,\ldots,O_t) \\
&= \frac{P(O_1,\ldots O_t)}{P(O_1,\ldots,O_{t+1})}P(O_{t+1}|S_{t+1})\sum_{S_t=s_t}P(S_{t+1}|S_t,O_1,\ldots,O_t)P(S_t=s_t|O_1,\ldots,O_t) \\
&= \frac{P(O_1,\ldots O_t)}{P(O_1,\ldots,O_{t+1})}P(O_{t+1}|S_{t+1})\sum_{S_t=s_t}P(S_{t+1}|S_t)P(S_t=s_t|O_1,\ldots,O_t)
\end{aligned}
$$

The forward algorithm

Problem: Given the observations up to time $t+1$, what is the posterior probability of S_{t+1}?

$$P(\text{next state}|\text{observations so far, next observation})$$
$$\propto P(\text{next observation}|\text{next state}) \sum_{\text{current state}} P(\text{next state}|\text{current state}) P(\text{current state}|\text{observations so far})$$

$$P(S_{t+1}|O_1, O_2, \ldots, O_{t+1}) \propto P(O_{t+1}|S_{t+1}) \sum_{S_t = s_t} P(S_{t+1}|S_t) P(S_t = s_t|O_1, \ldots, O_t)$$

The forward algorithm

Problem: Given the observations up to time $t+1$, what is the posterior probability of S_{t+1}?

$$P(\text{next state}|\text{observations so far, next observation})$$
$$\propto P(\text{next observation}|\text{next state}) \sum_{\text{current state}} P(\text{next state}|\text{current state}) P(\text{current state}|\text{observations so far})$$
$$\mathbf{P}(\mathbf{S_{t+1}}|\mathbf{O_1}, \mathbf{O_2}, \ldots, \mathbf{O_{t+1}}) \propto P(O_{t+1}|S_{t+1}) \sum_{S_t = s_t} P(S_{t+1}|S_t) \mathbf{P}(\mathbf{S_t} = \mathbf{s_t}|\mathbf{O_1}, \ldots, \mathbf{O_t})$$

We recursively calculate $P(S_t = s_t|O_1, \ldots, O_t)$, starting with $t = 1$

We will refer to $P(S_t|O_1, \ldots O_t)$ as $\texttt{forward}(t)$.

The backward algorithm

Problem: Given the observations up to time t, what is the posterior probability of $S_1, \ldots, S_t$?

$$
\begin{aligned}
P(S_k | O_1, \ldots, O_t) &= \frac{P(S_k, O_1, \ldots, O_t)}{P(O_1, \ldots, O_t)} \\
&= \frac{P(O_1, \ldots, O_k) P(S_k | O_1, \ldots, O_k) P(O_{k+1}, \ldots, O_t | S_k, O_1, \ldots, O_k)}{P(O_1, \ldots, O_t)} \\
&\propto P(S_k | O_1, \ldots, O_k) P(O_{k+1}, \ldots, O_t | S_k, O_1, \ldots, O_k) \\
&\propto \texttt{forward}(t) P(O_{k+1}, \ldots, O_t | S_k)
\end{aligned}
$$

The backward algorithm

Problem: Given the observations up to time t, what is the posterior probability of $S_1, \ldots, S_t$?

$$
\begin{aligned}
P(O_{k+1}, \ldots, O_t | S_k) &= \frac{P(O_{k+1}, \ldots, O_t, S_k)}{P(S_k)} \\
&= \frac{\sum_{S_{k+1}=s_{k+1}} P(O_{k+1}, \ldots, O_t, S_k, S_{k+1} = s_{k+1})}{P(S_k)} \\
&= \frac{\sum_{S_{k+1}=s_{k+1}} P(O_{k+1}, \ldots, O_t | S_k, S_{k+1} = s_{k+1}) P(S_{k+1} = s_{k+1} | S_k) P(S_k)}{P(S_k)} \\
&= \sum_{S_{k+1}=s_{k+1}} P(O_{k+1}, \ldots, O_t | S_k, S_{k+1} = s_{k+1}) P(S_{k+1} = s_{k_1} | S_k) \\
&= \sum_{S_{k+1}=s_{k+1}} P(O_{k+1}, \ldots, O_t | S_{k+1} = s_{k+1}) P(S_{k+1} = s_{k+1} | S_k) \\
&= \sum_{S_{k+1}=s_{k+1}} P(O_{k+1} | S_{k+1}) P(O_{k+2}, \ldots, O_t | S_{k+1} = s_{k+1}) P(S_{k+1} = s_{k+1} | S_k)
\end{aligned}
$$

The backward algorithm

Problem: Given the observations up to time t, what is the posterior probability of $S_1, \ldots, S_t$?

$$P(\text{remaining observations}|\text{current state})$$
$$= \sum_{\text{next state}} P(\text{next state}|\text{current state}) P(\text{next observation}|\text{next state}) P(\text{further observations}|\text{next state})$$

$$P(O_{k+1}, \ldots, O_t | S_k) = \sum_{S_{k+1}=s_{k+1}} P(S_{k+1} = s_{k+1}|S_k) P(O_{k+1}|S_{k+1}) P(O_{k+2}, \ldots, O_t | S_{k+1} = s_{k+1})$$

The backward algorithm

Problem: Given the observations up to time t, what is the posterior probability of $S_1, \ldots, S_t$?

$$P(\text{remaining observations}|\text{current state})$$
$$= \sum_{\text{next state}} P(\text{next state}|\text{current state})P(\text{next observation}|\text{next state})P(\text{further observations}|\text{next state})$$

$$\mathbf{P(O_{k+1}, \ldots, O_t|S_k)} = \sum_{S_{k+1}=s_{k+1}} P(S_{k+1} = s_{k+1}|S_k)P(O_{k+1}|S_{k+1})\mathbf{P(O_{k+2}, \ldots, O_t|S_{k+1} = s_{k+1})}$$

We recursively calculate $P(O_{k+2}, \ldots, O_t|S_{k+1} = s_{k+1})$, starting with $k = t$.

We will refer to $P(O_{k+2}, \ldots, O_t|S_{k+1} = s_{k+1})$ as backward(t).

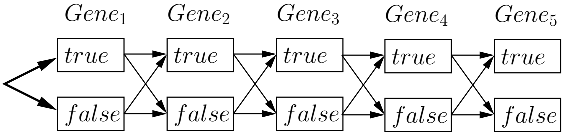
The forward-backward algorithm

procedure FORWARDBACKWARD(observations $O_1 \ldots O_t$, prior $P(S_0)$)
  $forward_0 \leftarrow P(S_0)$
  for i in 1 to t **do**
    $forward_i \leftarrow \texttt{forward}(forward_{i-1}, O_i)$
  end for
  $b \leftarrow \overrightarrow{1}$
  for i in t downto 1 **do**
    $smoothed_i \leftarrow \texttt{normalize}(forward_i \times b)$
    $b \leftarrow \texttt{backward}(b, O_i)$
  end for
  return b
end procedure

The Viterbi algorithm

Problem: Given the observations up to time t, what is the *most likely instantiation* of $S_1, \ldots, S_t$?

We can think about this as a path-finding problem.



The probability of a state is the probability of the most likely path to that state.

The Viterbi algorithm

Problem: Given the observations up to time t, what is the *most likely instantiation* of $S_1, \dots, S_t$?

We can think about this as a path-finding problem.

$$\max_{\text{path so far}} P(\text{path so far, next state in path} | \text{observations so far, next observation})$$

$$\propto P(\text{next observation} | \text{next state}) \left\{ \max_{\text{current state}} P(\text{next state} | \text{current state}) \right.$$

$$\left. \max_{\text{previous states}} P(\text{previous states, current state} | \text{observations so far}) \right\}$$

$$\max_{s_1 \dots s_t} P(s_1 \dots s_t, S_{t+1} | O_1 \dots O_{t+1})$$

$$\propto P(O_{t+1} | S_{t+1}) \left\{ \max_{s_t} P(S_{t+1} | s_t) \left\{ \max_{s_1 \dots s_{t-1}} P(s_1 \dots s_{t-1}, s_t | O_1 \dots O_t) \right\} \right\}$$

The Viterbi algorithm

Problem: Given the observations up to time t, what is the *most likely instantiation* of $S_1, \ldots, S_t$?

We can think about this as a path-finding problem.

$$\max_{\text{path so far}} P(\text{path so far, next state in path}|\text{observations so far, next observation})$$

$$\propto P(\text{next observation}|\text{next state}) \left\{ \max_{\text{current state}} P(\text{next state}|\text{current state}) \right.$$

$$\left. \max_{\text{previous states}} P(\text{previous states, current state}|\text{observations so far} \right\}$$

$$\max_{\mathbf{s_1 \ldots s_t}} \mathbf{P(s_1 \ldots s_t, S_{t+1}|O_1 \ldots O_{t+1})}$$

$$\propto P(O_{t+1}|S_{t+1}) \left\{ \max_{s_t} P(S_{t+1}|s_t) \left\{ \max_{\mathbf{s_1 \ldots s_{t-1}}} \mathbf{P(s_1 \ldots s_{t-1}, s_t|O_1 \ldots O_t)} \right\} \right\}$$

Recap

During this section, we discussed

- Stationary, first-order Markov processes
- Hidden Markov models (HMMs)
- Prediction in HMMs with the forward algorithm
- Posterior probability calcuations with the backward algorithm
- Efficient calculations with the forward-backward algorithm
- Identifying the most likely instantiation of the state variables with the Viterbi algorithm

Next in probabilistic models

- The belief propagation algorithm for efficient inference in polytree networks