Hidden Markov Models

Andrew W. Moore
Professor
School of Computer Science
Carnegie Mellon University

www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599
A Markov System

Has $N$ states, called $s_1$, $s_2$ .. $s_N$

There are discrete timesteps, $t=0$, $t=1$, ...

$N = 3$

$t=0$
A Markov System

Has $N$ states, called $s_1, s_2 \ldots s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the $t$’th timestep the system is in exactly one of the available states. Call it $q_t$

Note: $q_t \in \{s_1, s_2 \ldots s_N\}$

$N = 3$
$t=0$
$q_t = q_0 = s_3$
A Markov System

Has $N$ states, called $s_1, s_2, \ldots, s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the $t$'th timestep the system is in exactly one of the available states. Call it $q_t$

Note: $q_t \in \{s_1, s_2, \ldots, s_N\}$

Between each timestep, the next state is chosen randomly.

$N = 3$
$t=1$
$q_t = q_1 = s_2$
A Markov System

Has $N$ states, called $s_1, s_2 \ldots s_N$

There are discrete timesteps, $t=0, t=1, \ldots$

On the $t$'th timestep the system is in exactly one of the available states. Call it $q_t$

Note: $q_t \in \{s_1, s_2 \ldots s_N\}$

Between each timestep, the next state is chosen randomly.

The current state determines the probability distribution for the next state.
A Markov System

Has $N$ states, called $s_1$, $s_2$ .. $s_N$

There are discrete timesteps, $t=0$, $t=1$, ...

On the $t$’th timestep the system is in exactly one of the available states. Call it $q_t$

Note: $q_t \in \{s_1, s_2 .. s_N\}$

Between each timestep, the next state is chosen randomly.

The current state determines the probability distribution for the next state.

Often notated with arcs between states

$N = 3$

t=1

$q_t = q_1 = s_2$

P($q_{t+1} = s_1 | q_t = s_2$) = 1/2
P($q_{t+1} = s_2 | q_t = s_2$) = 1/2
P($q_{t+1} = s_3 | q_t = s_2$) = 0

P($q_{t+1} = s_1 | q_t = s_1$) = 0
P($q_{t+1} = s_2 | q_t = s_1$) = 0
P($q_{t+1} = s_3 | q_t = s_1$) = 1

P($q_{t+1} = s_1 | q_t = s_3$) = 1/3
P($q_{t+1} = s_2 | q_t = s_3$) = 2/3
P($q_{t+1} = s_3 | q_t = s_3$) = 0
What is $P(q_t = s)$? Slow, stupid answer

Step 1: Work out how to compute $P(Q)$ for any path $Q$

$= q_1 q_2 q_3 \ldots q_t$

Given we know the start state $q_1$ (i.e. $P(q_1) = 1$)

$P(q_1 q_2 \ldots q_t) = P(q_1 q_2 \ldots q_{t-1}) P(q_t | q_1 q_2 \ldots q_{t-1})$

$= P(q_1 q_2 \ldots q_{t-1}) P(q_t | q_{t-1})$ \hspace{1cm} **WHY?**

$= P(q_2 | q_1) P(q_3 | q_2) \ldots P(q_t | q_{t-1})$

Step 2: Use this knowledge to get $P(q_t = s)$

$$P(q_t = s) = \sum_{Q \in \text{Paths of length } t \text{ that end in } s} P(Q)$$

**Computation is exponential in t**
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define
  
  \[ p_t(i) = \text{Prob. state is } s_i \text{ at time } t \]
  
  \[ = P(q_t = s_i) \]

- Easy to do inductive definition

\[ \forall i \quad p_0(i) = \]

\[ \forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = \]
What is $P(q_t = s)$? Clever answer

• For each state $s_i$, define
  \[ p_t(i) = \text{Prob. state is } s_i \text{ at time } t \]
  \[ = P(q_t = s_i) \]

• Easy to do inductive definition
  \[ \forall i \quad p_0(i) = \begin{cases} 
  1 & \text{if } s_i \text{ is the start state} \\
  0 & \text{otherwise} 
\end{cases} \]

  \[ \forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = \]
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define
  
  $p_t(i) = \text{Prob. state is } s_i \text{ at time } t$
  
  $= P(q_t = s_i)$

- Easy to do inductive definition

  $\forall i \quad p_0(i) = \begin{cases} 
  1 & \text{if } s_i \text{ is the start state} \\
  0 & \text{otherwise}
  \end{cases}$

  $\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = 
  \sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) =$
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define
  
  $$p_t(i) = \text{Prob. state is } s_i \text{ at time } t$$
  
  $$= P(q_t = s_i)$$

- Easy to do inductive definition

  $$\forall i \quad p_0(i) = \begin{cases} 
  1 & \text{if } s_i \text{ is the start state} \\
  0 & \text{otherwise} 
  \end{cases}$$

  $$\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) =$$

  $$\sum_{i=1}^{N} P(q_{t+1} = s_j \wedge q_t = s_i) =$$

  $$\sum_{i=1}^{N} P(q_{t+1} = s_j \mid q_t = s_i) P(q_t = s_i) = \sum_{i=1}^{N} a_{ij} p_t(i)$$

  Remember,

  $$a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$$
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define
  
  $p_t(i) = \text{Prob. state is } s_i \text{ at time } t$
  
  $= P(q_t = s_i)$

- Easy to do inductive definition

  $\forall i \quad p_0(i) = \begin{cases} 
  1 & \text{if } s_i \text{ is the start state} \\
  0 & \text{otherwise}
  \end{cases}$

  $\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) =$

  $\sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) =$

  $\sum_{i=1}^{N} P(q_{t+1} = s_j \mid q_t = s_i) P(q_t = s_i) = \sum_{i=1}^{N} a_{ij} p_t(i)$

- Computation is simple.
- Just fill in this table in this order:

<table>
<thead>
<tr>
<th>$t$</th>
<th>$p_t(1)$</th>
<th>$p_t(2)$</th>
<th>\ldots</th>
<th>$p_t(N)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>\ldots</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t_{\text{final}}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is $P(q_t = s)$? Clever answer

- For each state $s_i$, define $p_t(i) = \text{Prob. state is } s_i \text{ at time } t = P(q_t = s_i)$
- Easy to do inductive definition

\[
\forall i \quad p_0(i) = \begin{cases} 
1 & \text{if } s_i \text{ is the start state} \\
0 & \text{otherwise}
\end{cases}
\]

\[
\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = 
\sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) = 
\sum_{i=1}^{N} P(q_{t+1} = s_j \mid q_t = s_i) P(q_t = s_i) = 
\sum_{i=1}^{N} a_{ij} p_t(i)
\]

- Cost of computing $P_t(i)$ for all states $S_i$ is now $O(t N^2)$
- The stupid way was $O(N^t)$
- This was a simple example
- It was meant to warm you up to this trick, called *Dynamic Programming*, because HMMs do many tricks like this.
Hidden State

- The previous example tried to estimate $P(q_t = s_i)$ unconditionally (using no observed evidence).
- Suppose we can observe something that’s affected by the true state.
- Example: **Proximity sensors**. (tell us the contents of the 8 adjacent squares)

True state $q_t$

What the robot sees: Observation $O_t$
Noisy Hidden State

- Example: **Noisy Proximity sensors.** (unreliably tell us the contents of the 8 adjacent squares)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>$R_0$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>H</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

True state $q_t$

<table>
<thead>
<tr>
<th>W</th>
<th>W</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathbb{R}$</td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Uncorrupted Observation

<table>
<thead>
<tr>
<th>W</th>
<th>W</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$H$</td>
<td>$H$</td>
<td></td>
</tr>
</tbody>
</table>

What the robot sees: Observation $O_t$
Noisy Hidden State

- Example: **Noisy Proximity sensors.** (unreliably tell us the contents of the 8 adjacent squares)

True state $q_t$

$O_t$ is noisily determined depending on the current state.

Assume that $O_t$ is conditionally independent of $\{q_{t-1}, q_{t-2}, \ldots q_1, q_0, O_{t-1}, O_{t-2}, \ldots O_1, O_0\}$ given $q_t$.

In other words:

$P(O_t = X | q_t = s_i) =$

$P(O_t = X | q_t = s_i, any \ earlier \ history) =$
Hidden Markov Models

Our robot with noisy sensors is a good example of an HMM

• **Question 1: State Estimation**
  
  What is $P(q_T=S_i \mid O_1O_2\ldots O_T)$?

  It will turn out that a new cute D.P. trick will get this for us.

• **Question 2: Most Probable Path**
  
  Given $O_1O_2\ldots O_T$, what is the most probable path that I took?

  And what is that probability?

  Yet another famous D.P. trick, the VITERBI algorithm, gets this.

• **Question 3: Learning HMMs:**
  
  Given $O_1O_2\ldots O_T$, what is the maximum likelihood HMM that could have produced this string of observations?

  Very very useful. Uses the E.M. Algorithm
Basic Operations in HMMs

For an observation sequence \( O = O_1 \ldots O_T \), the three basic HMM operations are:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation:</td>
<td>Forward-Backward</td>
<td>( O(TN^2) )</td>
</tr>
<tr>
<td>Calculating ( P(q_t=S_i \mid O_1O_2\ldots O_t) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inference:</td>
<td>Viterbi Decoding</td>
<td>( O(TN^2) )</td>
</tr>
<tr>
<td>Computing ( Q^* = \text{argmax}_Q P(Q \mid O) )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learning:</td>
<td>Baum-Welch (EM)</td>
<td>( O(TN^2) )</td>
</tr>
<tr>
<td>Computing ( \lambda^* = \text{argmax}_\lambda P(O \mid \lambda) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( T = \# \text{ timesteps, } N = \# \text{ states} \)
HMM Notation
(from Rabiner’s Survey)

The states are labeled $S_1, S_2, \ldots, S_N$.

For a particular trial:

Let $T$ be the number of observations
${\mathbf{T}}$ is also the number of states passed through
$O = O_1 O_2 \ldots O_T$ is the sequence of observations
$Q = q_1 q_2 \ldots q_T$ is the notation for a path of states

$\lambda = \langle N, M, \{\pi_i\}, \{a_{ij}\}, \{b_i(j)\} \rangle$ is the specification of an HMM
HMM Formal Definition

An HMM, $\lambda$, is a 5-tuple consisting of

- $N$ the number of states
- $M$ the number of possible observations
- $\{\pi_1, \pi_2, \ldots, \pi_N\}$ The starting state probabilities
  \[ P(q_0 = S_i) = \pi_i \]

- The state transition probabilities
  \[ P(q_{t+1} = S_j | q_t = S_i) = a_{ij} \]

- The observation probabilities
  \[ P(O_t = k | q_t = S_i) = b_i(k) \]
Here’s an HMM

Start randomly in state 1 or 2
Choose one of the output symbols in each state at random.

N = 3  
M = 3

\( \pi_1 = 1/2 \)  \( \pi_2 = 1/2 \)  \( \pi_3 = 0 \)

\( a_{11} = 0 \)  \( a_{12} = 1/3 \)  \( a_{13} = 2/3 \)
\( a_{12} = 1/3 \)  \( a_{22} = 0 \)  \( a_{13} = 2/3 \)
\( a_{13} = 1/3 \)  \( a_{32} = 1/3 \)  \( a_{13} = 1/3 \)

\( b_1 (X) = 1/2 \)  \( b_1 (Y) = 1/2 \)  \( b_1 (Z) = 0 \)
\( b_2 (X) = 0 \)  \( b_2 (Y) = 1/2 \)  \( b_2 (Z) = 1/2 \)
\( b_3 (X) = 1/2 \)  \( b_3 (Y) = 0 \)  \( b_3 (Z) = 1/2 \)

Caution: this slide contains errors
Here’s an HMM

Start randomly in state 1 or 2
Choose one of the output symbols in each state at random.
Let’s generate a sequence of observations:

\( N = 3 \)
\( M = 3 \)
\( \pi_1 = \frac{1}{2} \)
\( \pi_2 = \frac{1}{2} \)
\( \pi_3 = 0 \)

\( a_{11} = 0 \)
\( a_{12} = \frac{1}{3} \)
\( a_{13} = \frac{1}{3} \)
\( a_{21} = \frac{2}{3} \)
\( a_{22} = 0 \)
\( a_{23} = \frac{2}{3} \)
\( a_{31} = \frac{1}{3} \)
\( a_{32} = \frac{1}{3} \)
\( a_{33} = \frac{1}{3} \)

\( b_1 (X) = \frac{1}{2} \)
\( b_1 (Y) = \frac{1}{2} \)
\( b_1 (Z) = 0 \)
\( b_2 (X) = 0 \)
\( b_2 (Y) = \frac{1}{2} \)
\( b_2 (Z) = \frac{1}{2} \)
\( b_3 (X) = \frac{1}{2} \)
\( b_3 (Y) = 0 \)
\( b_3 (Z) = \frac{1}{2} \)

\[ q_0 = \_ \quad O_0 = \_ \]
\[ q_1 = \_ \quad O_1 = \_ \]
\[ q_2 = \_ \quad O_2 = \_ \]
Here’s an HMM

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let’s generate a sequence of observations:

50-50 choice between X and Y

N = 3
M = 3
\( \pi_1 = \frac{1}{2} \) \( \pi_2 = \frac{1}{2} \) \( \pi_3 = 0 \)

\( a_{11} = 0 \) \( a_{12} = \frac{1}{3} \) \( a_{13} = \frac{2}{3} \)
\( a_{12} = \frac{1}{3} \) \( a_{22} = 0 \) \( a_{13} = \frac{2}{3} \)
\( a_{13} = \frac{1}{3} \) \( a_{32} = \frac{1}{3} \) \( a_{13} = \frac{1}{3} \)

\( b_1 (X) = \frac{1}{2} \) \( b_1 (Y) = \frac{1}{2} \) \( b_1 (Z) = 0 \)
\( b_2 (X) = 0 \) \( b_2 (Y) = \frac{1}{2} \) \( b_2 (Z) = \frac{1}{2} \)
\( b_3 (X) = \frac{1}{2} \) \( b_3 (Y) = 0 \) \( b_3 (Z) = \frac{1}{2} \)
Here’s an HMM

Start randomly in state 1 or 2
Choose one of the output symbols in each state at random.
Let’s generate a sequence of observations:

\[ N = 3 \]
\[ M = 3 \]
\[ \pi_1 = \frac{1}{2} \quad \pi_2 = \frac{1}{2} \quad \pi_3 = 0 \]

\[ a_{11} = 0 \quad a_{12} = \frac{1}{3} \quad a_{13} = \frac{2}{3} \]
\[ a_{12} = \frac{1}{3} \quad a_{22} = 0 \quad a_{13} = \frac{2}{3} \]
\[ a_{13} = \frac{1}{3} \quad a_{32} = \frac{1}{3} \quad a_{13} = \frac{1}{3} \]

\[ b_1 (X) = \frac{1}{2} \quad b_1 (Y) = \frac{1}{2} \quad b_1 (Z) = 0 \]
\[ b_2 (X) = 0 \quad b_2 (Y) = \frac{1}{2} \quad b_2 (Z) = \frac{1}{2} \]
\[ b_3 (X) = \frac{1}{2} \quad b_3 (Y) = 0 \quad b_3 (Z) = \frac{1}{2} \]
Here’s an HMM

Start randomly in state 1 or 2
Choose one of the output symbols in each state at random.

Let’s generate a sequence of observations:

50-50 choice between Z and X

N = 3
M = 3
π₁ = ½
π₂ = ½
π₃ = 0

a₁₁ = 0
a₁₂ = ½
a₁₃ = ½

a₁₂ = ½
a₂₂ = 0
a₃₂ = ½

a₁₃ = ½
a₁₃ = ½
a₁₃ = ½

b₁ (X) = ½
b₁ (Y) = ½
b₁ (Z) = 0

b₂ (X) = 0
b₂ (Y) = ½
b₂ (Z) = ½

b₃ (X) = ½
b₃ (Y) = 0
b₃ (Z) = ½

| q₀ = S₁ | O₀ = X |
| q₁ = S₃ | O₁ =   |
| q₂ =    | O₂ =   |
Here’s an HMM

Start randomly in state 1 or 2
Choose one of the output symbols in each state at random.
Let’s generate a sequence of observations:

Each of the three next states is equally likely

\[ q_0 = S_1 \quad O_0 = X \]
\[ q_1 = S_3 \quad O_1 = X \]

\[ q_2 = \_ \_ \_ \quad O_2 = \_ \_ \_ \]

N = 3
M = 3
\( \pi_1 = \frac{1}{2} \quad \pi_2 = \frac{1}{2} \quad \pi_3 = 0 \)

\( a_{11} = 0 \quad a_{12} = \frac{1}{3} \quad a_{13} = \frac{2}{3} \)
\( a_{12} = \frac{1}{3} \quad a_{22} = 0 \quad a_{13} = \frac{2}{3} \)
\( a_{13} = \frac{1}{3} \quad a_{32} = \frac{1}{3} \quad a_{13} = \frac{1}{3} \)

\( b_1 (X) = \frac{1}{2} \quad b_1 (Y) = \frac{1}{2} \quad b_1 (Z) = 0 \)
\( b_2 (X) = 0 \quad b_2 (Y) = \frac{1}{2} \quad b_2 (Z) = \frac{1}{2} \)
\( b_3 (X) = \frac{1}{2} \quad b_3 (Y) = 0 \quad b_3 (Z) = \frac{1}{2} \)
Here’s an HMM

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let’s generate a sequence of observations:

N = 3
M = 3
π₁ = ½
π₂ = ½
π₃ = 0

a₁₁ = 0
a₁₂ = ¼
a₁₃ = ¼

a₁₂ = ¾
a₂₂ = 0
a₂₃ = ¼

a₁₃ = ¾
a₃₂ = ¼
a₃₃ = ¾

b₁ (X) = ½
b₁ (Y) = ½
b₁ (Z) = 0

b₂ (X) = 0
b₂ (Y) = ½
b₂ (Z) = ½

b₃ (X) = ½
b₃ (Y) = 0
b₃ (Z) = ½

50-50 choice between Z and X

<table>
<thead>
<tr>
<th>q₀</th>
<th>S₁</th>
<th>O₀</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>q₁</td>
<td>S₃</td>
<td>O₁</td>
<td>X</td>
</tr>
<tr>
<td>q₂</td>
<td>S₃</td>
<td>O₂</td>
<td></td>
</tr>
</tbody>
</table>
Here’s an HMM

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let’s generate a sequence of observations:

\( N = 3 \)
\( M = 3 \)
\( \pi_1 = \frac{1}{2} \)
\( \pi_2 = \frac{1}{2} \)
\( \pi_3 = 0 \)

\( a_{11} = 0 \)
\( a_{12} = \frac{1}{3} \)
\( a_{13} = \frac{1}{3} \)
\( a_{21} = \frac{1}{3} \)
\( a_{22} = 0 \)
\( a_{23} = \frac{1}{3} \)
\( a_{31} = \frac{1}{3} \)
\( a_{32} = \frac{1}{3} \)
\( a_{33} = 0 \)

\( b_1 (X) = \frac{1}{2} \)
\( b_1 (Y) = \frac{1}{2} \)
\( b_1 (Z) = 0 \)
\( b_2 (X) = 0 \)
\( b_2 (Y) = \frac{1}{2} \)
\( b_2 (Z) = \frac{1}{2} \)
\( b_3 (X) = \frac{1}{2} \)
\( b_3 (Y) = 0 \)
\( b_3 (Z) = \frac{1}{2} \)

| \( q_0 \) |   | \( q_1 \) |   | \( q_2 \) |
|---|---|---|---|
| \( S_1 \) | \( O_0 = X \) | \( S_3 \) | \( O_1 = X \) | \( S_3 \) | \( O_2 = Z \) |
State Estimation

Start randomly in state 1 or 2

Choose one of the output symbols in each state at random.

Let’s generate a sequence of observations:

This is what the observer has to work with...

|   | \(q_0\) |   | \(O_0\) |   | \(q_1\) |   | \(O_1\) |   | \(q_2\) |   | \(O_2\) |
|---|---|---|---|---|---|---|---|---|---|---|
|   |   | \(\text{?}\) |   | \(X\) |   | \(\text{?}\) |   | \(X\) |   | \(\text{?}\) |   | \(Z\) |

\(N = 3\)  
\(M = 3\)  
\(\pi_1 = \frac{1}{2}\) \(\pi_2 = \frac{1}{2}\) \(\pi_3 = 0\)

\(a_{11} = 0\)  
\(a_{12} = \frac{1}{3}\)  
\(a_{13} = \frac{1}{3}\)  
\(a_{22} = 0\)  
\(a_{32} = \frac{1}{3}\)  
\(a_{13} = \frac{1}{3}\)

\(b_1 (X) = \frac{1}{2}\)  
\(b_2 (X) = 0\)  
\(b_3 (X) = \frac{1}{2}\)  
\(b_1 (Y) = \frac{1}{2}\)  
\(b_2 (Y) = \frac{1}{2}\)  
\(b_3 (Y) = 0\)  
\(b_1 (Z) = 0\)  
\(b_2 (Z) = \frac{1}{2}\)  
\(b_3 (Z) = \frac{1}{2}\)
Prob. of a series of observations

What is \( P(O) = P(O_1, O_2, O_3) = P(O_1 = X \land O_2 = X \land O_3 = Z) \)?

Slow, stupid way:

\[
P(O) = \sum_{Q \in \text{Paths of length 3}} P(O \land Q) \\
= \sum_{Q \in \text{Paths of length 3}} P(O | Q) P(Q)
\]

How do we compute \( P(Q) \) for an arbitrary path \( Q \)?

How do we compute \( P(O|Q) \) for an arbitrary path \( Q \)?
Prob. of a series of observations

What is \( P(O) = P(O_1 \, O_2 \, O_3) = P(O_1 = X \, O_2 = X \, O_3 = Z) \)?

Slow, stupid way:

\[
P(O) = \sum_{Q \in \text{Paths of length 3}} P(O \cap Q)
= \sum_{Q \in \text{Paths of length 3}} P(O|Q)P(Q)
\]

How do we compute \( P(Q) \) for an arbitrary path \( Q \)?

How do we compute \( P(O|Q) \) for an arbitrary path \( Q \)?

\[
P(Q) = P(q_1, q_2, q_3)
= P(q_1) \, P(q_2, q_3|q_1) \quad \text{(chain rule)}
= P(q_1) \, P(q_2|q_1) \, P(q_3|q_2, q_1) \quad \text{(chain)}
= P(q_1) \, P(q_2|q_1) \, P(q_3|q_2) \quad \text{(why?)}
\]

Example in the case \( Q = S_1 \, S_3 \, S_3 \): \( P(O) = P(O_1 = X \, O_2 = X \, O_3 = Z) \):

\[
\frac{1}{2} * \frac{2}{3} * \frac{1}{3} = \frac{1}{9}
\]
Prob. of a series of observations

What is $P(O) = P(O_1 O_2 O_3) = P(O_1 = X \land O_2 = X \land O_3 = Z)$?

Slow, stupid way:

$$P(O) = \sum_{Q \in \text{Paths of length 3}} P(O \land Q)$$

$$= \sum_{Q \in \text{Paths of length 3}} P(O|Q)P(Q)$$

How do we compute $P(Q)$ for an arbitrary path $Q$?

How do we compute $P(O|Q)$ for an arbitrary path $Q$?

Example in the case $Q = S_1 S_3 S_3$:

$$P(O|Q) = P(O_1 \land q_1)P(O_2 \land q_2)P(O_3 \land q_3)$$

$$= P(O_1|q_1)P(O_2|q_2)P(O_3|q_3)$$

$$= P(X|S_1)P(X|S_3)P(Z|S_3)$$

$$= 1/2 \times 1/2 \times 1/2 = 1/8$$
Prob. of a series of observations

What is \( P(O) = P(O_1, O_2, O_3) = P(O_1 = X \land O_2 = X \land O_3 = Z) \)?

Slow, stupid way:

\[
P(O) = \sum_{Q \in \text{Paths of length } 3} P(O \land Q)
= \sum_{Q \in \text{Paths of length } 3} P(O|Q)P(Q)
\]

How do we compute \( P(Q) \) for an arbitrary path \( Q \)?

How do we compute \( P(O|Q) \) for an arbitrary path \( Q \)?

A sequence of 20 observations would need \( 3^{20} = 3.5 \) billion \( P(O|Q) \) computations

So let’s be smarter…
The Prob. of a given series of observations, non-exponential-cost-style

Given observations $O_1 \ O_2 \ ... \ O_T$

Define

$$\alpha_t(i) = P(O_1 \ O_2 \ ... \ O_t \ \land \ q_t = S_i \ | \ \lambda) \quad \text{where } 1 \leq t \leq T$$

$\alpha_t(i) =$ Probability that, in a random trial,

- We’d have seen the first t observations
- We’d have ended up in $S_i$ as the t’th state visited.

In our example, what is $\alpha_2(3)$ ?
$\alpha_t(i)$: easy to define recursively

$\alpha_t(i) = P(O_1 \ O_2 \ldots \ O_T \land q_t = S_i \mid \lambda)$ \hspace{1em} ($\alpha_t(i)$ can be defined stupidly by considering all paths length “t”. How?)

$$\alpha_1(i) = P(O_1 \land q_1 = S_i)$$
$$= P(q_1 = S_i)P(O_1 \mid q_1 = S_i)$$
$$= \text{what?}$$

$$\alpha_{t+1}(j) = P(O_1O_2\ldots O_tO_{t+1} \land q_{t+1} = S_j)$$
$$= \text{what?}$$
\( \alpha_t(i) \): easy to define recursively

\[ \alpha_t(i) = P(O_1 O_2 \ldots O_T \land q_t = S_i \mid \lambda) \]  

(\( \alpha_t(i) \) can be defined stupidly by considering all paths length “\( t \). How?)

\[ \alpha_1(i) = P(O_1 \land q_1 = S_i) \]
\[ = P(q_1 = S_i) P(O_1 \mid q_1 = S_i) \]
\[ = \text{what?} \]

\[ \alpha_{t+1}(j) = P(O_1 O_2 \ldots O_t O_{t+1} \land q_{t+1} = S_j) \]
\[ = \sum_{i=1}^{N} P(O_1 O_2 \ldots O_t \land q_t = S_i \land O_{t+1} \land q_{t+1} = S_j) \]
\[ = \sum_{i=1}^{N} P(O_{t+1}, q_{t+1} = S_j \mid O_1 O_2 \ldots O_t \land q_t = S_i) P(O_1 O_2 \ldots O_t \land q_t = S_i) \]
\[ = \sum_{i} P(O_{t+1}, q_{t+1} = S_j \mid q_t = S_i) \alpha_t(i) \]
\[ = \sum_{i} P(q_{t+1} = S_j \mid q_t = S_i) P(O_{t+1} \mid q_{t+1} = S_j) \alpha_t(i) \]
\[ = \sum_{i} a_{ij} b_{j}(O_{t+1}) \alpha_t(i) \]
in our example

\[ \alpha_i (i) = P(O_1 O_2 \ldots O_t \land q_t = S_i | \lambda) \]

\[ \alpha_1 (i) = b_i (O_1) \pi_i \]

\[ \alpha_{t+1} (j) = \sum_i a_{ij} b_j (O_{t+1}) \alpha_t (i) \]

---

WE SAW \( O_1 O_2 O_3 = X X Z \)

\[ \alpha_1 (1) = \frac{1}{4} \quad \alpha_1 (2) = 0 \quad \alpha_1 (3) = 0 \]

\[ \alpha_2 (1) = 0 \quad \alpha_2 (2) = 0 \quad \alpha_2 (3) = \frac{1}{12} \]

\[ \alpha_3 (1) = 0 \quad \alpha_3 (2) = \frac{1}{72} \quad \alpha_3 (3) = \frac{1}{72} \]
Easy Question

We can cheaply compute

$$\alpha_t(i) = P(O_1 O_2 \ldots O_t \land q_t = S_i)$$

(How) can we cheaply compute

$$P(O_1 O_2 \ldots O_t)$$

(How) can we cheaply compute

$$P(q_t = S_i | O_1 O_2 \ldots O_t)$$
Easy Question

We can cheaply compute

\[ \alpha_t(i) = P(O_1 O_2 \ldots O_t \land q_t = S_i) \]

(How) can we cheaply compute

\[ P(O_1 O_2 \ldots O_t) ? \]

\[ \sum_{i=1}^{N} \alpha_t(i) \]

(How) can we cheaply compute

\[ P(q_t = S_i | O_1 O_2 \ldots O_t) \]

\[ \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} \]
Most probable path given observations

What's most probable path given $O_1O_2...O_T$, i.e.

What is $\underset{Q}{\text{argmax}} \ P(Q|O_1O_2...O_T)$?

Slow, stupid answer:

$$\underset{Q}{\text{argmax}} \ P(Q|O_1O_2...O_T)$$

$$= \underset{Q}{\text{argmax}} \ \frac{P(O_1O_2...O_T|Q)P(Q)}{P(O_1O_2...O_T)}$$

$$= \underset{Q}{\text{argmax}} \ P(O_1O_2...O_T|Q)P(Q)$$
Efficient MPP computation

We’re going to compute the following variables:

$$\delta_t(i) = \max_{q_1 q_2 \ldots q_{t-1}} P(q_1 q_2 \ldots q_t = S_i \land O_1 \ldots O_t)$$

= The Probability of the path of Length t-1 with the maximum chance of doing all these things:

...OCCURING

and

...ENDING UP IN STATE $S_i$

and

...PRODUCING OUTPUT $O_1 \ldots O_t$

DEFINE:  $mpp_t(i) = \text{that path}$

So:  $\delta_t(i) = \text{Prob}(mpp_t(i))$
The Viterbi Algorithm

\[
\delta_t(i) = q_1 q_2 \cdots q_{t-1} P(q_1 q_2 \cdots q_{t-1} \land q_t = S_i \land O_1 O_2 \cdots O_t)
\]

\[
\arg\max mpp_t(i) = q_1 q_2 \cdots q_{t-1} P(q_1 q_2 \cdots q_{t-1} \land q_t = S_i \land O_1 O_2 \cdots O_t)
\]

\[
\max \delta_1(i) = \text{one choice } P(q_1 = S_i \land O_1)
\]

\[
= P(q_1 = S_i)P(O_1|q_1 = S_i)
\]

\[
= \pi_i b_i(O_1)
\]

Now, suppose we have all the \(\delta_t(i)\)'s and \(mpp_t(i)\)'s for all \(i\).

**HOW TO GET** \(\delta_{t+1}(j)\) and \(mpp_{t+1}(j)\)?

\(mpp_t(1)\) → \(S_1\) with \(\text{Prob} = \delta_t(1)\)

\(mpp_t(2)\) → \(S_2\) with \(\text{Prob} = \delta_t(2)\)

\(\vdots\)

\(mpp_t(N)\) → \(S_N\) with \(\text{Prob} = \delta_t(N)\)

\(q_t\) → \(S_j\) with \(\text{Prob} = \delta_t(N)\)
The most probable path with last two states $S_i, S_j$ is the most probable path to $S_i$, followed by transition $S_i \rightarrow S_j$. 
The Viterbi Algorithm

The most probable path with last two states $S_i$, $S_j$ is the most probable path to $S_j$, followed by transition $S_i \rightarrow S_j$.

What is the prob of that path?

$$\delta_t(i) \times P(S_i \rightarrow S_j \land O_{t+1} | \lambda)$$

$$= \delta_t(i) \ a_{ij} \ b_j (O_{t+1})$$

SO The most probable path to $S_j$ has $S_{i^*}$ as its penultimate state where $i^* = \text{argmax}_i \delta_t(i) \ a_{ij} \ b_j (O_{t+1})$.
The Viterbi Algorithm

What is the prob of that path?
\[ \delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1}) \]
\[ = \delta_t(i) a_{ij} b_j(O_{t+1}) \]
SO The most probable is 
\[ S_i^* \] as its penultimate state
where \( i^* = \text{argmax} \delta_t(i) a_{ij} b_j(O_{t+1}) \)

Summary:
\[ \delta_{t+1}(j) = \delta_t(i^*) a_{ij} b_j(O_{t+1}) \]
\[ \text{mpp}_{t+1}(j) = \text{mpp}_{t+1}(i^*)S_{i^*} \] with \( i^* \) defined to the left
What’s Viterbi used for?

Classic Example

Speech recognition:

Signal → words

HMM → observable is signal

→ Hidden state is part of word formation

What is the most probable word given this signal?

UTTERLY GROSS SIMPLIFICATION

In practice: many levels of inference; not one big jump.