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

$N = 3$
$t=0$
$q_0 = q_t = s_3$

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.
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.

Often notated with arcs between states
Markov Property

$q_{t+1}$ is conditionally independent of \{ $q_{t-1}$, $q_{t-2}$, ... $q_1$, $q_0$ \} given $q_t$.

In other words:

\[
P(q_{t+1} = s_j | q_t = s_i) = P(q_{t+1} = s_j | q_t = s_i, \text{any earlier history})
\]

Question: what would be the best Bayes Net structure to represent the Joint Distribution of ( $q_0$, $q_1$, $q_2$, $q_3$, $q_4$ )?

Answer:
Markov Property

$q_{t+1}$ is conditionally independent of $\{q_{t-1}, q_{t-2}, \ldots, q_1, q_0\}$ given $q_t$.

In other words:

$$P(q_{t+1} = s_j | q_t = s_i) = P(q_{t+1} = s_j | q_t = s_i, \text{any earlier history})$$

Question: what would be the best Bayes Net structure to represent the Joint Distribution of $(q_0, q_1, q_2, q_3, q_4)$?

$p(q_{t+1} = s_i | q_t = s_j) = 1/3$

$p(q_{t+1} = s_2 | q_t = s_3) = 2/3$

$p(q_{t+1} = s_3 | q_t = s_3) = 0$

Notation:

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$$

A Blind Robot

A human and a robot wander around randomly on a grid…

STATE $q =$ Location of Robot, Location of Human

Note: $N$ (num. states) = 18 * 18 = 324
Dynamics of System

Each timestep the human moves randomly to an adjacent cell. And Robot also moves randomly to an adjacent cell.

Typical Questions:

- “What’s the expected time until the human is crushed like a bug?”
- “What’s the probability that the robot will hit the left wall before it hits the human?”
- “What’s the probability Robot crushes human on next time step?”

Example Question

“It’s currently time t, and human remains uncrushed. What’s the probability of crushing occurring at time t + 1?”

If robot is blind:
- We can compute this in advance.

If robot is omnipotent:
- (I.E. If robot knows state at time t), can compute directly.

If robot has some sensors, but incomplete state information …
- Hidden Markov Models are applicable!

We’ll do this first

Too Easy. We won’t do this

Main Body of Lecture
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. $P(q_1)=1$)

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

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

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 $P(q_t = s)$

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

Computation is exponential in $t$

\[
\sum_{Q: \text{Paths of length } t \text{ that end in } s} P(Q)
\]

What is $P(q_t = s)$? Clever answer

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

  = $P(q_t = s_i)$

- Easy to do inductive definition

  $\forall i \ \ p_0(i) =$

  $\forall j \ \ p_{t+1}(j) = P(q_{t+1} = s_j) =$
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
  \[ p_t(i) = \text{Prob. state is } s_j \text{ at time } t \]
  \[ = P(q_t = s_j) \]
- Easy to do inductive definition

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

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

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
  \[ p_t(i) = \text{Prob. state is } s_j \text{ at time } t \]
  \[ = P(q_t = s_j) \]
- 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_0(1) )</th>
<th>( p_0(2) )</th>
<th>( \ldots )</th>
<th>( p_0(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>\ldots</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
  \[
  \forall i \quad p_0(i) = \begin{cases} 
  1 & \text{if } s_i \text{ is the start state} \\
  0 & \text{otherwise}
  \end{cases}
  \]
  \[
  \forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = 
  \sum_{i=1}^{N} P(q_{t+1} = s_j \land q_t = s_i) = 
  \sum_{i=1}^{N} P(q_{t+1} = s_j \mid q_t = s_i) P(q_t = s_i) = 
  \sum_{i=1}^{N} p_t(i)
  \]
- 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

“It’s currently time $t$, and human remains uncrushed. What’s the probability of crushing occurring at time $t + 1$?”

If robot is blind:
We can compute this in advance.

If robot is omnipotent:
(I.E. If robot knows state at time $t$), can compute directly.

If robot has some sensors, but incomplete state information …
Hidden Markov Models are applicable!

We’ll do this first
Too Easy. We won’t do this
Main Body of Lecture
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)

![Diagram of Hidden State example]

Noisy Hidden State

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

![Diagram of Noisy Hidden State example]
Noisy Hidden State

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

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R₂</td>
<td>H</td>
<td>2</td>
</tr>
</tbody>
</table>

True state qₜ

Oₜ is noisily determined depending on the current state.

Assume that Oₜ is conditionally independent of \(\{q_{t-1}, q_{t-2}, \ldots, q₁, q₀, O_{t-1}, O_{t-2}, \ldots, O₁, O₀\}\) given qₜ.

In other words:

\[
P(Oₜ = X | qₜ = sᵢ) = P(Oₜ = X | qₜ = sᵢ, \text{any earlier history})
\]

Question: what'd be the best Bayes Net structure to represent the Joint Distribution of \(q₀, q₁, q₂, q₃, q₄, O₀, O₁, O₂, O₃, O₄\)?
**Example:** Noisy Proximity sensors. (unreliably tell us the contents of the 8 adjacent squares)

$W$ denotes "WALL"

What the robot sees: Observation $O_t$

Question: what'd be the best Bayes Net structure to represent the Joint Distribution of $(q_0, q_1, q_2, q_3, q_4, O_0, O_1, O_2, O_3, O_4)$?

Answer:

<table>
<thead>
<tr>
<th>$O_0$</th>
<th>$O_1$</th>
<th>$O_2$</th>
<th>$O_3$</th>
<th>$O_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$b_1(1)$</th>
<th>$b_1(2)$</th>
<th>$b_1(k)$</th>
<th>$b_1(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b_2(1)$</th>
<th>$b_2(2)$</th>
<th>$b_2(k)$</th>
<th>$b_2(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b_3(1)$</th>
<th>$b_3(2)$</th>
<th>$b_3(k)$</th>
<th>$b_3(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$b_4(1)$</th>
<th>$b_4(2)$</th>
<th>$b_4(k)$</th>
<th>$b_4(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question: what'd be the best Bayes Net structure to represent the Joint Distribution of $(q_0, q_1, q_2, q_3, q_4)$ and the corresponding observations?
Hidden Markov Models

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

- **Question 1: State Estimation**
  
  What is $P(q_T=S_i | O_1O_2...O_T)$
  
  It will turn out that a new cute D.P. trick will get this for us.

- **Question 2: Most Probable Path**
  
  Given $O_1O_2...O_T$, 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 $O_1O_2...O_T$, what is the maximum likelihood HMM that could have produced this string of observations?
  
  Very very useful. Uses the E.M. Algorithm

Are H.M.M.s Useful?

You bet!!

- **Robot planning + sensing when there's uncertainty**
  
  (e.g. Reid Simmons / Sebastian Thrun / Sven Koenig)

- **Speech Recognition/Understanding**
  
  Phones $\rightarrow$ Words, Signal $\rightarrow$ phones

- **Human Genome Project**
  
  Complicated stuff your lecturer knows nothing about.

- **Consumer decision modeling**

- **Economics & Finance.**

  Plus at least 5 other things I haven't thought of.
Some Famous HMM Tasks

Question 1: State Estimation
What is $P(q_t = S_i | O_1 O_2 \ldots O_t)$?
Some Famous HMM Tasks

Question 1: State Estimation
What is $P(q_t = S_i | O_1 O_2 ... O_t)$?

Question 2: Most Probable Path
Given $O_1 O_2 ... O_T$, what is the most probable path that I took?
Question 1: State Estimation
What is \( P(q_T = S_i | O_1 O_2 \ldots O_T) \)?

Question 2: Most Probable Path
Given \( O_1 O_2 \ldots O_T \), what is the most probable path that I took?

Some Famous HMM Tasks

Woke up at 8.35, Got on Bus at 9.46, Sat in lecture 10.05-11.22…
Some Famous HMM Tasks

Question 1: State Estimation
What is P(q_T=S_i | O_1O_2...O_T)?

Question 2: Most Probable Path
Given O_1O_2...O_T, what is the most probable path that I took?

Question 3: Learning HMMs:
Given O_1O_2...O_T, what is the maximum likelihood HMM that could have produced this string of observations?
Some Famous HMM Tasks

Question 1: State Estimation
   What is \( P(q_T=S_i | O_1O_2...O_T) \)?

Question 2: Most Probable Path
   Given \( O_1O_2...O_T \), what is the most probable path that I took?

Question 3: Learning HMMs:
   Given \( O_1O_2...O_T \), what is the maximum likelihood HMM that could have produced this string of observations?

Basic Operations in HMMs

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Algorithm</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation: Calculating ( P(q_t=S_i</td>
<td>O_1O_2...O_t) )</td>
<td>Forward-Backward</td>
</tr>
<tr>
<td>Inference: Computing ( Q^* = \text{argmax}_Q P(Q</td>
<td>O) )</td>
<td>Viterbi Decoding</td>
</tr>
<tr>
<td>Learning: Computing ( \lambda^* = \text{argmax}_\lambda P(O</td>
<td>\lambda) )</td>
<td>Baum-Welch (EM)</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
$T$ 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
  $P(q_0 = S_i) = \pi_i$
- $a_{11} \, a_{22} \, \ldots \, a_{1N}$
  $a_{21} \, a_{22} \, \ldots \, a_{2N}$
  $\vdots$
  $a_{N1} \, a_{N2} \, \ldots \, a_{NN}$

- $b_{1}(1) \, b_{1}(2) \, \ldots \, b_{1}(M)$
  $b_{2}(1) \, b_{2}(2) \, \ldots \, b_{2}(M)$
  $\vdots$
  $b_{N}(1) \, b_{N}(2) \, \ldots \, b_{N}(M)$

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.

\[ \begin{align*}
N &= 3 \\
M &= 3 \\
\pi_1 &= \frac{1}{2} \\
\pi_2 &= \frac{1}{2} \\
\pi_3 &= 0 \\
a_{11} &= 0 \\
a_{12} &= \frac{1}{3} \\
a_{13} &= \frac{2}{3} \\
a_{21} &= \frac{1}{3} \\
a_{22} &= 0 \\
a_{23} &= \frac{2}{3} \\
a_{31} &= \frac{1}{3} \\
a_{32} &= \frac{1}{3} \\
a_{33} &= \frac{1}{3} \\
b_1 (X) &= \frac{1}{2} \\
b_1 (Y) &= \frac{1}{2} \\
b_1 (Z) &= 0 \\
b_2 (X) &= 0 \\
b_2 (Y) &= \frac{1}{2} \\
b_2 (Z) &= \frac{1}{2} \\
b_3 (X) &= \frac{1}{2} \\
b_3 (Y) &= 0 \\
b_3 (Z) &= \frac{1}{2}
\end{align*} \]

Let’s generate a sequence of observations:

<table>
<thead>
<tr>
<th>( q_0 )</th>
<th>( q_1 )</th>
<th>( q_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( O_0 )</td>
<td>( O_1 )</td>
<td>( O_2 )</td>
</tr>
</tbody>
</table>

\[ \begin{align*}
\text{50-50 choice between } S_1 \text{ and } S_2
\end{align*} \]
Here’s an HMM

\[ \begin{align*}
N &= 3 \\
M &= 3 \\
\pi_1 &= \frac{1}{2} \\
\pi_2 &= \frac{1}{2} \\
\pi_3 &= 0 \\
a_{11} &= 0 \\
a_{12} &= \frac{1}{3} \\
a_{13} &= \frac{2}{3} \\
a_{12} &= \frac{1}{3} \\
a_{22} &= 0 \\
a_{13} &= \frac{2}{3} \\
a_{23} &= \frac{1}{3} \\
a_{32} &= \frac{1}{3} \\
a_{13} &= \frac{2}{3} \\
b_1(X) &= \frac{1}{2} \\
b_1(Y) &= \frac{1}{2} \\
b_1(Z) &= 0 \\
b_2(X) &= 0 \\
b_2(Y) &= \frac{1}{2} \\
b_2(Z) &= \frac{1}{2} \\
b_3(X) &= \frac{1}{2} \\
b_3(Y) &= 0 \\
b_3(Z) &= \frac{1}{2}
\end{align*} \]

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:

\[ \begin{array}{c|c|c}
q_0 &= S_1 & O_0 = \_ \\
q_1 &= \_ & O_1 = \_ \\
q_2 &= \_ & O_2 = \_ \\
\end{array} \]
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
\( \pi_1 = \frac{1}{2} \) \( \pi_2 = \frac{1}{2} \) \( \pi_3 = 0 \)

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

\( b_1 (X) = \frac{1}{2} \) \( b_1 (Y) = \frac{1}{2} \) \( b_1 (Z) = 0 \)
\( b_2 (X) = 0 \) \( b_2 (Y) = \frac{1}{3} \) \( b_2 (Z) = \frac{2}{3} \)
\( b_3 (X) = \frac{1}{2} \) \( b_3 (Y) = 0 \) \( b_3 (Z) = \frac{1}{2} \)

Copyright © Andrew W. Moore
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 = ⅓
a₁₃ = ⅔
da₁₂ = ⅓
a₂₂ = 0
a₂₃ = ⅔
da₁₃ = ⅓
a₃₂ = ⅓
a₃₃ = ⅓

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
State Estimation

N = 3
M = 3
π₁ = ½
π₂ = ½
π₃ = 0

a₁₁ = 0
a₁₂ = ⅓
a₁₃ = ⅔
a₂₁ = ⅔
a₂₂ = 0
a₂₃ = ⅓
a₃₁ = ⅓
a₃₂ = ⅓
a₃₃ = ⅓

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

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…

<table>
<thead>
<tr>
<th>q_0</th>
<th>q_1</th>
<th>q_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>O₀ = X</td>
<td>O₁ = X</td>
<td>O₂ = Z</td>
</tr>
</tbody>
</table>

Prob. of a series of observations

What is P(O) = P(O₁ O₂ O₃) = P(O₁ = X ^ O₂ = X ^ O₃ = Z)?

Slow, stupid way:

P(O) = Σ P(O ∧ Q)
     = Σ 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(XYZ) = P(Y = S₁ S₂ S₃) = P(O₃ = X | S₁ S₂ S₃, O₁ = X)
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:

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

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

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 $Q = S_1 S_3 S_3$:

$$= \frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} = \frac{1}{9}$$

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:

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

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

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 $Q = S_1 S_3 S_3$:

$$= P(X| S_1) P(X| S_3) P(Z| S_3) = \frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$
The 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:

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

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 \( 2^{20} = 1,048,576 \) computations and 1,048,576 \( P(O|Q) \) computations.

So let's be smarter...

The Prob. of a given series of observations, non-exponential-cost-style

Given observations \( O_1 \, O_2 \, \ldots \, O_T \)

Define

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

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

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

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

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

\[
= P(q_t = S_i) P(O_t \mid q_t = S_i)
\]

\[
= \pi_t b_i(O_t)
\]

\( \alpha_{t+1}(j) = P(O_{t+1}, O_t, q_{t+1} = S_j) \)

\[
= \sum_{t=1}^N \alpha_t(i) \cdot \alpha_{t+1}(j)
\]

\( \alpha_t(i) \): easy to define recursively

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

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

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

\[
= P(q_t = S_i) P(O_t \mid q_t = S_i)
\]

\[
= \pi_t b_i(O_t)
\]

\( \alpha_{t+1}(j) = P(O_{t+1}, O_t, q_{t+1} = S_j) \)

\[
= \sum_{t=1}^N \alpha_t(i) \cdot \alpha_{t+1}(j)
\]
in our example

\[ \alpha_t(i) = P(O_1O_2\ldots O_t \wedge q_t = S_i | \lambda) \]
\[ \alpha_1(i) = b_i(O_1) \pi_i \]
\[ \alpha_{t+1}(j) = \sum_i a_{ij} b_j(O_{t+1}) \alpha_t(i) \]

**WE SAW** \(O_1O_2O_3 = XZX\)

\[ \alpha_1(1) = \frac{1}{4} \quad \alpha_1(2) = 0 \quad \alpha_1(3) = 0 \]
\[ \alpha_2(1) = 0 \quad \alpha_2(2) = 0 \quad \alpha_2(3) = \frac{1}{12} \]
\[ \alpha_3(1) = 0 \quad \alpha_3(2) = \frac{1}{72} \quad \alpha_3(3) = \frac{1}{72} \]

**Easy Question**

We can cheaply compute

\[ \alpha_t(i) = P(O_1O_2\ldots O_t \wedge q_t = S_i) \]

(How) can we cheaply compute \(P(O_1O_2\ldots O_t)\)?

\[ \sum_i P(O_1\ldots O_t \wedge q_t = S_i) = \sum_i \alpha_t(i) \]

(How) can we cheaply compute

\[ P(q_t = S_i | O_1O_2\ldots O_t) \]

\[ \frac{\alpha_t(i)}{P(O_1\ldots O_t)} = \frac{\alpha_t(i)}{\sum_j \alpha_t(j)} \]
**Easy Question**

We can cheaply compute

\[ \alpha_t(i) = P(O_1O_2...O_t \land q_t=S_i) \]

(How) can we cheaply compute

\[ P(O_1O_2...O_t) ? \]

(How) can we cheaply compute

\[ P(q_t=S_i|O_1O_2...O_t) \]

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

**Most probable path given observations**

What's most probable path given \(O_1O_2...O_T\), i.e.

What is \( \arg\max_Q P(Q|O_1O_2...O_T) \)?

Slow, stupid answer:

\[ \arg\max_Q P(Q|O_1O_2...O_T) \]

\[ = \arg\max_Q \frac{P(O_1O_2...O_T|Q)P(Q)}{P(O_1O_2...O_T)} \]

\[ = \arg\max_Q P(O_1O_2...O_T|Q)P(Q) \]
Efficient MPP computation

We’re going to compute the following variables:

\[ \delta_t(i) = \max_{q_1q_2\ldots q_{t-1}} P(q_1, q_2 \ldots, q_{t-1} \land q_t = S_i \land O_1 \ldots O_t) \]

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

…OCCURRING

and

…ENDING UP IN STATE S_i

and

…PRODUCING OUTPUT O_1\ldots O_t

DEFINE: mpp_t(i) = that path

So: \[ \delta_t(i) = \text{Prob}(\text{mpp}_t(i)) \]

The Viterbi Algorithm

\[ \delta_t(i) = q_1q_2\ldots q_{t-1} \max_{q_t} P(q_1, q_2 \ldots, q_{t-1}, q_t = S_i \land O_1 \ldots O_t) \]

mpp_\text{arg max}_t(i) = q_1q_2\ldots q_{t-1} \max_{q_t} P(q_1, q_2 \ldots, q_{t-1}, q_t = S_i \land O_1 \ldots O_t)

\[ \delta_t(i) = \text{one choice } P(q_t = S_i \land O_t) \]

\[ = p(q_t = S_i)p(O_t | q_t = S_i) \]

\[ = \pi_i b_i(O_t) \]

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

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

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

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

\vdots

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

\rightarrow q_t \rightarrow \text{Prob}=\delta_t(N) \rightarrow S_N

\vdots

\rightarrow q_{t+1} \rightarrow S_i
The Viterbi Algorithm

The most probable path with last two states $S_i, S_j$

is

the most probable path to $S_i$, followed by transition $S_i \rightarrow S_j$

What is the probability of that path?

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

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

SO The most probable path to $S_j$ has $S_{i^*}$ as its penultimate state

where $i^* = \arg\max_i \delta_t(i) \cdot a_{ij} \cdot b_j(O_{t+1})$
The Viterbi Algorithm

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

What is the probability 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 $S_i^*$ as its penultimate state, where $i^* = \arg\max_i \delta(t(i)) a_{ij} b_j(O_{t+1})$

Summary:

$$\delta(t+1(j)) = \delta(t(i^*)) a_{ij} b_j(O_{t+1})$$

$$\text{mppt}(t+1(j)) = \text{mppt}(t(i^*)) S_i^*$$

What's Viterbi used for?

Classic Example

Speech recognition:

Signal $\rightarrow$ words

HMM $\rightarrow$ observable is signal

$\rightarrow$ 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.
HMMs are used and useful

But how do you design an HMM?

Occasionally, (e.g. in our robot example) it is reasonable to deduce the HMM from first principles.

But usually, especially in Speech or Genetics, it is better to infer it from large amounts of data.  \( O_1 O_2 \ldots O_T \) with a big “T”.

Observations previously in lecture

\[ O_1 O_2 \ldots O_T \]

Observations in the next bit

\[ O_1 O_2 \ldots O_T \]

Inferring an HMM

Remember, we’ve been doing things like

\[ P(O_1 O_2 \ldots O_T | \lambda) \]

That “\( \lambda \)” is the notation for our HMM parameters.

Now we have some observations and we want to estimate \( \lambda \) from them.

AS USUAL: We could use

(i) MAX LIKELIHOOD  

\[ \lambda = \text{argmax } P(O_1 \ldots O_T | \lambda) \]

(ii) BAYES

Work out \( P(\lambda | O_1 \ldots O_T) \) and then take \( E[\lambda] \) or \( \max \lambda P(\lambda | O_1 \ldots O_T) \)
Max likelihood HMM estimation

Define

\[ \gamma_i(t) = P(q_t = S_i | O_1O_2...O_T, \lambda) \]
\[ \epsilon_{i,j}(t) = P(q_t = S_i \wedge q_{t+1} = S_j | O_1O_2...O_T, \lambda) \]

\[ \gamma_i(t) \text{ and } \epsilon_{i,j}(t) \text{ can be computed efficiently } \forall i,j,t \]

(Details in Rabiner paper)

\[ \sum_{t=1}^{T-1} \gamma_i(t) = \text{ Expected number of transitions out of state } i \text{ during the path} \]
\[ \sum_{t=1}^{T-1} \epsilon_{i,j}(t) = \text{ Expected number of transitions from state } i \text{ to state } j \text{ during the path} \]

HMM estimation

Notice

\[ \frac{\sum_{t=1}^{T-1} \gamma_i(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} = \left( \frac{\text{expected frequency}}{i \rightarrow j} \right) \]
\[ \frac{\sum_{t=1}^{T-1} \epsilon_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} = \text{ Estimate of } \text{Prob(Next state } S_j | \text{This state } S_i) \]

We can re-estimate

\[ a_{ij} \leftarrow \frac{\sum_{t=1}^{T-1} \epsilon_{i,j}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \]

We can also re-estimate

\[ b_j(O_t) \leftarrow \ldots \]

(See Rabiner)
We want \( a_{ij}^{\text{new}} = \text{new estimate of } P(q_{t+1} = s_j \mid q_t = s_i) \)

\[
\gamma_{\text{old}} = \text{MLE a matrix } \lambda_{\text{old}} \text{ EASY}
\]

\[
\gamma_{\text{new}} = \text{Expected values of hidden states}
\]

\[
= \sum_{k=1}^{N} \text{Expected # transitions } i \rightarrow j \mid \lambda_{\text{old}, k}, O_1, O_2, \cdots O_T
\]

\[
= \sum_{k=1}^{N} \text{Expected # transitions } i \rightarrow k \mid \lambda_{\text{old}, k}, O_1, O_2, \cdots O_T
\]
We want $a_{ij}^{\text{new}} = \text{new estimate of } P(q_{t+1} = s_j \mid q_t = s_i)$

$$= \frac{\text{Expected } \# \text{ transitions } i \rightarrow j \mid \lambda_{\text{old}}, O_1, O_2, \cdots O_T}{\sum_{k=1}^{N} \text{Expected } \# \text{ transitions } i \rightarrow k \mid \lambda_{\text{old}}, O_1, O_2, \cdots O_T}$$

$$= \frac{\sum_{t=1}^{T-1} P(q_{t+1} = s_j, q_t = s_i \mid \lambda_{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T-1} P(q_{t+1} = s_k, q_t = s_i \mid \lambda_{\text{old}}, O_1, O_2, \cdots O_T)}$$
\[ P(q_{t+1} = s_j \mid q_t = s_i) \]
\[ = \sum_{k=1}^{N} \frac{\text{Expected } # \text{ transitions } i \rightarrow k \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T}{\text{Expected } # \text{ transitions } i \rightarrow j \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T} \]
\[ = \sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T) \]
\[ = \frac{\sum_{k=1}^{N} S_{ij}}{\sum_{k=1}^{N} S_{ik}} \]

We want \( a_{ij}^{\text{new}} \) = new estimate of \( P(q_{t+1} = s_j \mid q_t = s_i) \)

\[ a_{ij}^{\text{new}} = P(q_{t+1} = s_j \mid q_t = s_i) \]

\[ = \sum_{k=1}^{N} \frac{\text{Expected } # \text{ transitions } i \rightarrow j \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T}{\text{Expected } # \text{ transitions } i \rightarrow k \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T} \]

\[ = \sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda_{t}^{\text{old}}, O_1, O_2, \cdots O_T) \]

\[ = \frac{\sum_{k=1}^{N} S_{ij}}{\sum_{k=1}^{N} S_{ik}} \]

\[ = a_{ij} \sum_{t=1}^{T} \alpha_{t}(i) \beta_{t+1}(j) b_{f}(O_{t+1}) \]
We want \( a_{ij}^{\text{new}} = \frac{S_{ij}}{\sum_{k=1}^{N} S_{ik}} \) where \( S_{ij} = a_{ij} \sum_{t=1}^{T} \alpha_t(i) \beta_{t+1}(j) b_j(O_{t+1}) \)
**EM for HMMs**

If we knew $\lambda$ we could estimate EXPECTATIONS of quantities such as

- Expected number of times in state $i$
- Expected number of transitions $i \rightarrow j$

If we knew the quantities such as

- Expected number of times in state $i$
- Expected number of transitions $i \rightarrow j$

We could compute the MAX LIKELIHOOD estimate of

$$\lambda = \langle \{a_{ij}\}, \{b_i(j)\}, \pi_i \rangle$$

Roll on the EM Algorithm…

---

**EM 4 HMMs**

1. Get your observations $O_1 \ldots O_T$
2. Guess your first $\lambda$ estimate $\lambda(0)$, $k=0$
3. $k = k+1$
4. Given $O_1 \ldots O_T$, $\lambda(k)$ compute
   $$\gamma_t(i), \varepsilon_t(i,j) \quad \forall 1 \leq t \leq T, \quad \forall 1 \leq i \leq N, \quad \forall 1 \leq j \leq N$$
5. Compute expected freq. of state $i$, and expected freq. $i \rightarrow j$
6. Compute new estimates of $a_{ij}, b_i(j), \pi_i$ accordingly. Call them $\lambda(k+1)$
7. Goto 3, unless converged.
   - Also known (for the HMM case) as the BAUM-WELCH algorithm.
Bad News

• There are lots of local minima

Good News

• The local minima are usually adequate models of the data.

Notice

• EM does not estimate the number of states. That must be given.
• Often, HMMs are forced to have some links with zero probability. This is done by setting $a_{ij}=0$ in initial estimate $\lambda(0)$
• Easy extension of everything seen today: HMMs with real valued outputs

---

Trade-off between too few states (inadequately modeling the structure in the data) and too many (fitting the noise).
Thus #states is a regularization parameter.
Blah blah blah... bias variance tradeoff...blah blah...cross-validation...blah blah....AIC, BIC....blah blah (same ol' same ol')
What You Should Know

• What is an HMM?
• Computing (and defining) $\alpha_t(i)$
• The Viterbi algorithm
• Outline of the EM algorithm
• To be very happy with the kind of maths and analysis needed for HMMs
• Fairly thorough reading of Rabiner* up to page 266*
  [Up to but not including “IV. Types of HMMs”].


DON’T PANIC: starts on p. 257.