

Note to other teachers and users of these slides. Andrew would be delighted if you found this source material useful in giving your own lectures. Feel free to use these slides verbatim, or to modify them to fit your own needs. PowerPoint originals are available. If you make use of a significant portion of these slides in your own lecture, please include this message, or the following link to the source repository of Andrew's tutorials: <http://www.cs.cmu.edu/~awm/tutorials> . Comments and corrections gratefully received.

Hidden Markov Models

Andrew W. Moore

Professor

School of Computer Science

Carnegie Mellon University

www.cs.cmu.edu/~awm

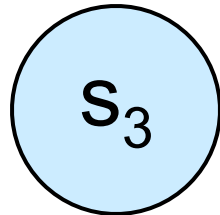
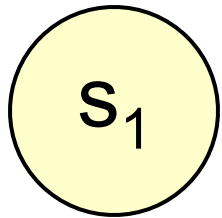
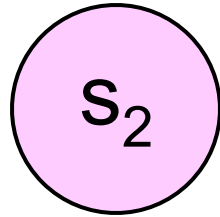
awm@cs.cmu.edu

412-268-7599

A Markov System

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

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



$N = 3$

$t=0$

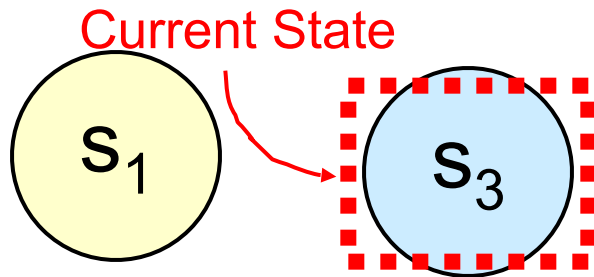
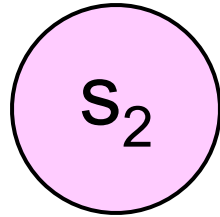
A Markov System

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

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

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

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



$N = 3$

$t=0$

$q_t = q_0 = s_3$

A Markov System

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

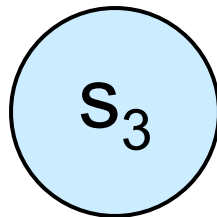
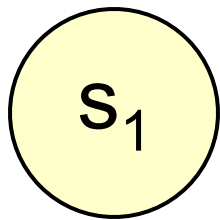
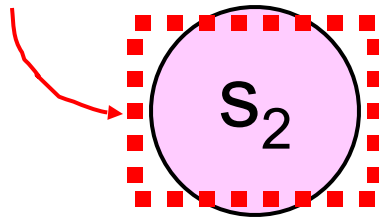
There are discrete timesteps,
 $t=0, t=1, \dots$

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

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

Between each timestep, the next
state is chosen randomly.

Current State



$N = 3$

$t=1$

$q_t = q_1 = s_2$

A Markov System

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

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

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

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

Between each timestep, the next
state is chosen randomly.

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

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

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

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

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

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

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

s_2

s_1

s_3

$N = 3$

$t=1$

$q_t=q_1=s_2$

$$P(q_{t+1}=s_1|q_t=s_3) = 1/3$$

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

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

A Markov System

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

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

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

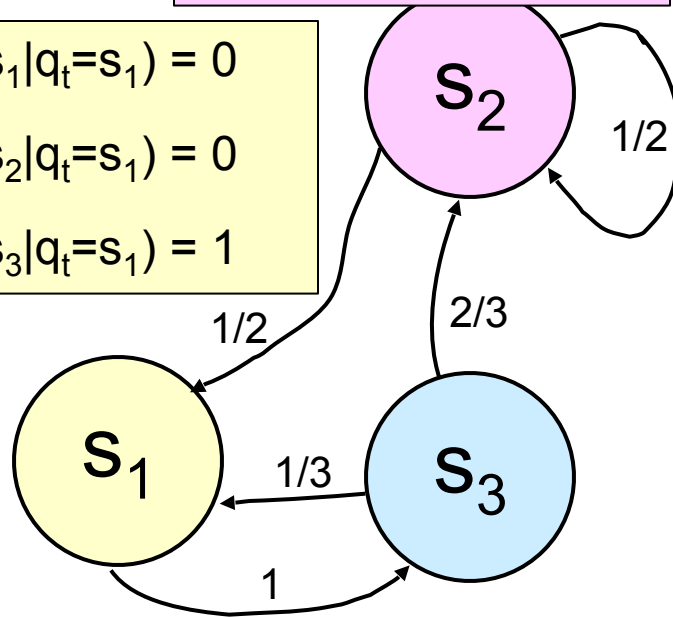
Note: $q_t \in \{s_1, s_2 \dots s_N\}$

Between each timestep, the next state is chosen randomly.

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

$$\begin{aligned} P(q_{t+1}=s_1|q_t=s_2) &= 1/2 \\ P(q_{t+1}=s_2|q_t=s_2) &= 1/2 \\ P(q_{t+1}=s_3|q_t=s_2) &= 0 \end{aligned}$$

$$\begin{aligned} P(q_{t+1}=s_1|q_t=s_1) &= 0 \\ P(q_{t+1}=s_2|q_t=s_1) &= 0 \\ P(q_{t+1}=s_3|q_t=s_1) &= 1 \end{aligned}$$



$N = 3$

$t=1$

$q_t=q_1=s_2$

$$\begin{aligned} P(q_{t+1}=s_1|q_t=s_3) &= 1/3 \\ P(q_{t+1}=s_2|q_t=s_3) &= 2/3 \\ P(q_{t+1}=s_3|q_t=s_3) &= 0 \end{aligned}$$

Often notated with arcs between states

Markov Property

q_{t+1} is conditionally independent of $\{q_{t-1}, q_{t-2}, \dots, q_1, q_0\}$ 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 $(q_0, q_1, \dots, q_3, q_4)$?

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

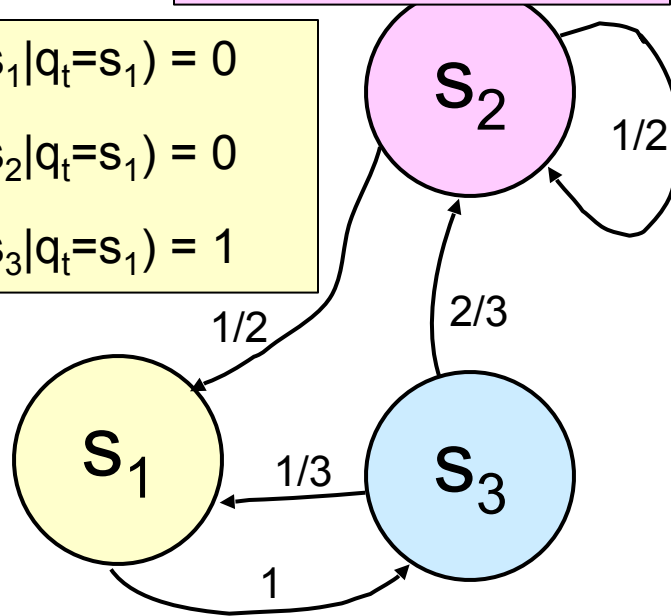
$$P(q_{t+1}=s_2|q_t=s_2) = 1/2$$

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

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

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

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



$$P(q_{t+1}=s_1|q_t=s_3) = 1/3$$

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

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

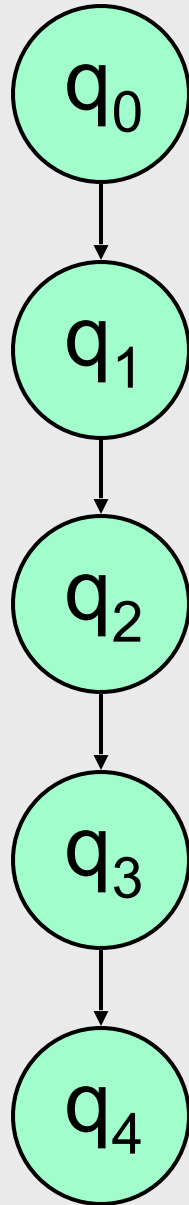
$N = 3$

$t=1$

$q_t=q_1=s_2$

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

Answer:



Markov Property

q_{t+1} is conditionally independent of $\{ q_{t-1}, q_{t-2}, \dots q_1, q_0 \}$ 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 $(q_0, q_1, q_2, q_3, q_4)$?

Markov Property

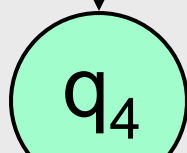
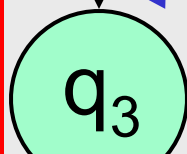
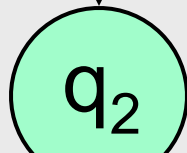
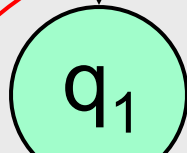
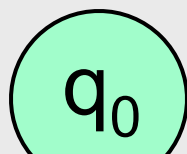
q_{t+1} is conditionally independent

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

Answer:

$P(q_t)$
 $P(q_t)$
 $P(q_t)$

Each of these probability tables is identical



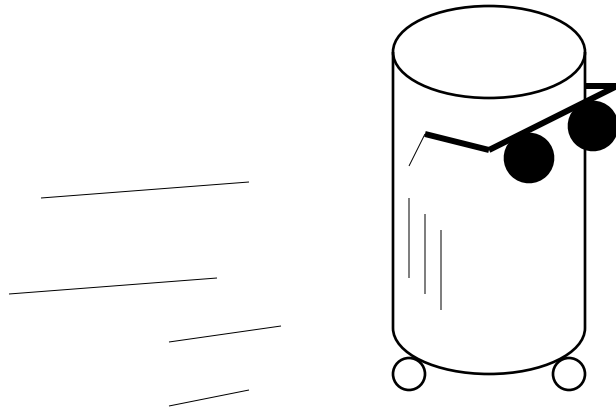
i	$P(q_{t+1}=s_1 q_t=s_i)$	$P(q_{t+1}=s_2 q_t=s_i)$...	$P(q_{t+1}=s_j q_t=s_i)$...	$P(q_{t+1}=s_N q_t=s_i)$
1	a_{11}	a_{12}	...	a_{1j}	...	a_{1N}
2	a_{21}	a_{22}	...	a_{2j}	...	a_{2N}
3	a_{31}	a_{32}	...	a_{3j}	...	a_{3N}
:	:	:	:	:	:	:
i	a_{i1}	a_{i2}	...	a_{ij}	...	a_{iN}
N	a_{N1}	a_{N2}	...	a_{Nj}	...	a_{NN}

q_2, q_3, q_4)?

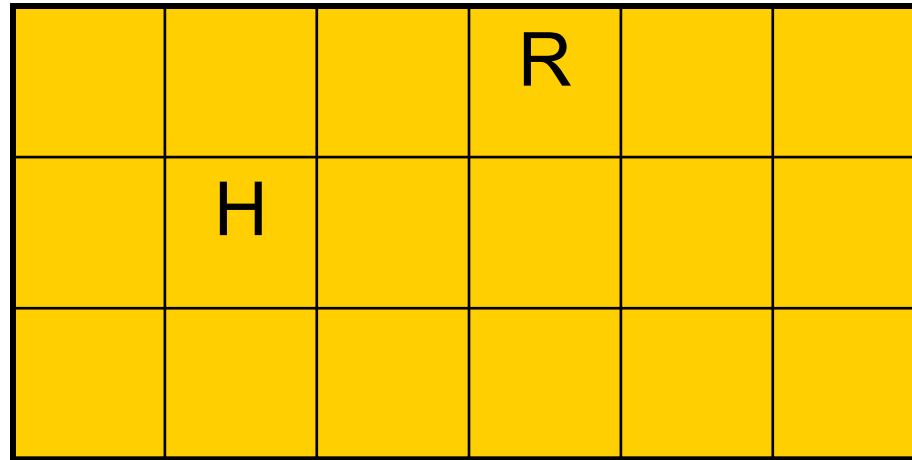
Notation:

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

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

$q_0 =$

					R
H					

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

Typical Questions:

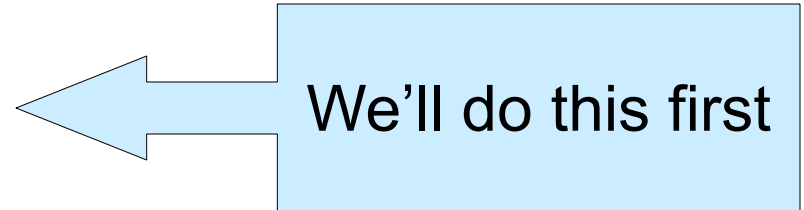
- “What’s the expected time until the human is crushed like a bug?”
- “What’s the probability that the robot will hit the left wall before it hits the human?”
- “What’s the probability Robot crushes human on next time step?”

Example Question

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

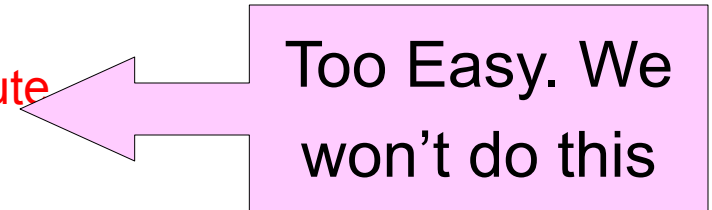
If robot is blind:

We can compute this in advance.



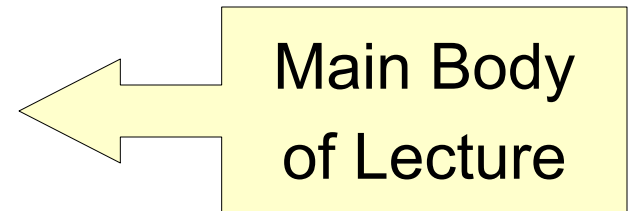
If robot is omnipotent:

(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!



What is $P(q_t = s)$? slow, stupid answer

Step 1: Work out how to compute $P(Q)$ for any path Q
 $= q_1 q_2 q_3 \dots q_t$

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

$$\begin{aligned} P(q_1 q_2 \dots q_t) &= P(q_1 q_2 \dots q_{t-1}) P(q_t | q_1 q_2 \dots q_{t-1}) \\ &= P(q_1 q_2 \dots q_{t-1}) P(q_t | q_{t-1}) \quad \text{WHY?} \\ &= P(q_2 | q_1) P(q_3 | q_2) \dots P(q_t | q_{t-1}) \end{aligned}$$

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

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

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

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

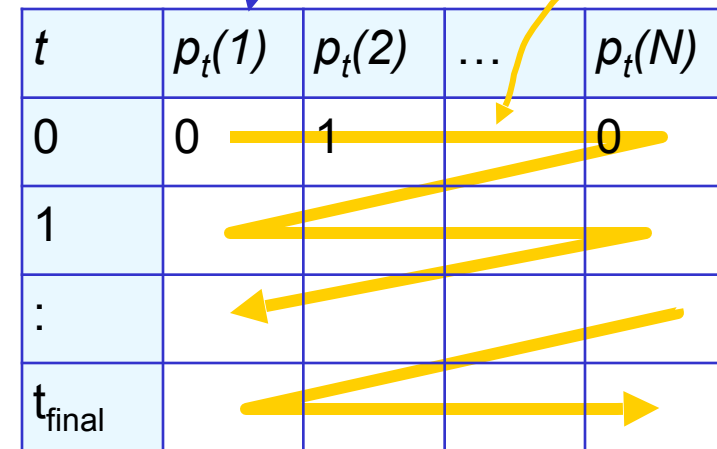
$$\forall i \quad p_0(i) = \begin{cases} 1 & \text{if } s_i \text{ is the start state} \\ 0 & \text{otherwise} \end{cases}$$

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

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

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

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



t	$p_t(1)$	$p_t(2)$...	$p_t(N)$
0	0	1		0
1				
\vdots				
t_{final}				

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

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

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

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

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

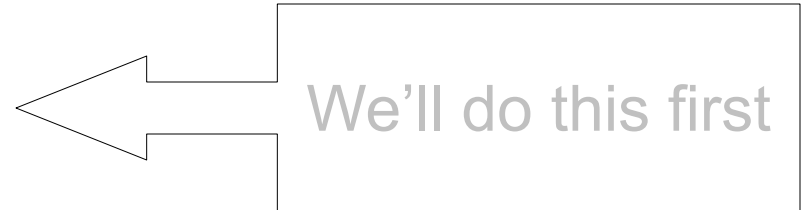
- Cost of computing $P_t(i)$ for all states S_i is now $O(t N^2)$
- The stupid way was $O(N^t)$
- This was a simple example
- It was meant to warm you up to this trick, called *Dynamic Programming*, because HMMs do many tricks like this.

Hidden State

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

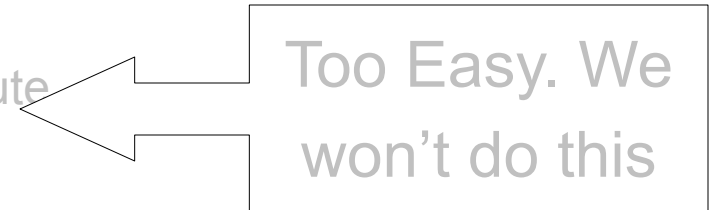
If robot is blind:

We can compute this in advance.



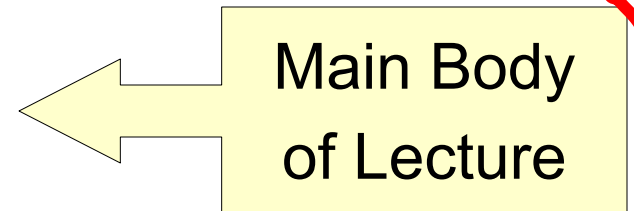
If robot is omnipotent:

(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!

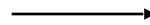


Hidden State

- 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)

			R_0		
		H			

True state q_t



W	W	W
	Ⓜ	
H		

What the robot sees:
Observation O_t

W
denotes
"WALL"

Noisy Hidden State

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

			R_0		
		H			

True state q_t



W	W	W
	Ⓡ	
H		

W
denotes
"WALL"

Uncorrupted Observation



W		W
	Ⓡ	W
H	H	

What the robot sees:
Observation O_t

Noisy Hidden State

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

			R_0		2
		H			

True state q_t

O_t is noisily determined depending on the current state.

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

In other words:

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

$$P(O_t = X | q_t = s_i, \text{any earlier history})$$

W	W	W
	Ⓡ	
H		

W denotes
"WALL"

Uncorrupted Observation

W		W
	Ⓡ	W
H	H	

What the robot sees:
Observation O_t

Noisy Hidden State

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

			R_0		2
		H			

True state q_t

O_t is noisily determined depending on the current state.

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

In other words:

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

$$P(O_t = X | q_t = s_i, \text{any earlier history})$$

W	W	W
	Ⓡ	
H		

W denotes
"WALL"

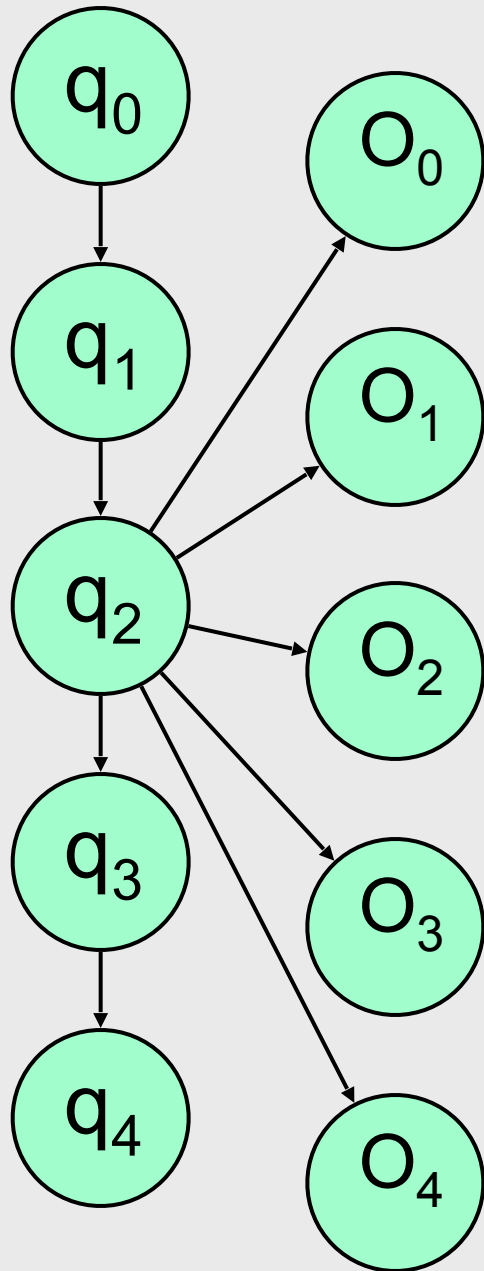
Uncorrupted Observation

W		W
	Ⓡ	W
H	H	

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:



Hidden State

Proximity sensors. (unreliably tell us adjacent squares)

W	W	W
	Ⓡ	
H		

W denotes
"WALL"

Uncorrupted Observation

W		W
	Ⓡ	W
H	H	

What the robot sees:
Observation O_t

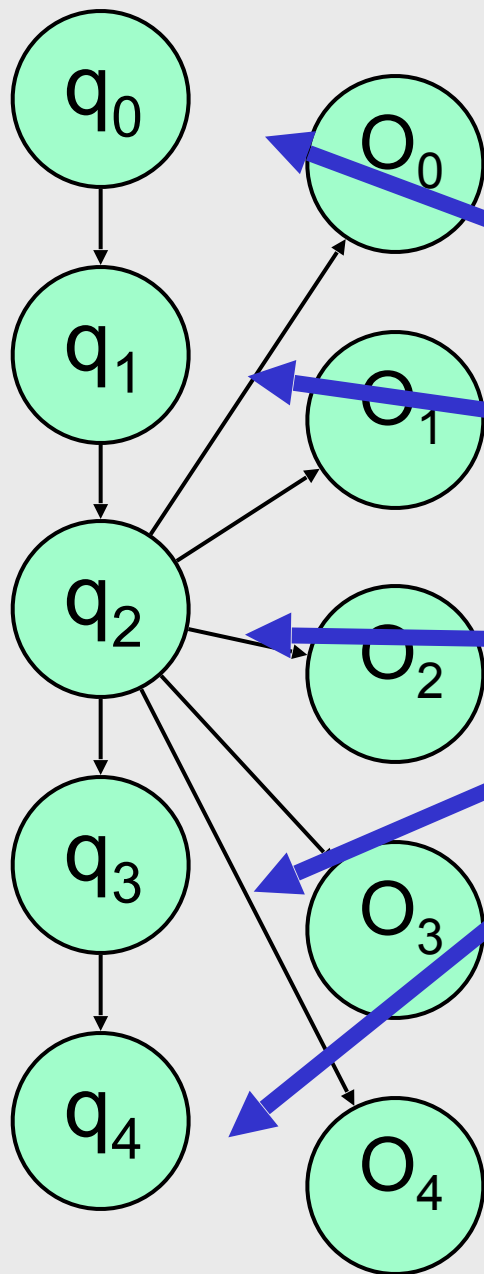
depending on

O_{t-1} ,

history)

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:



Hidden Markov Models

Proximity sensor

adjacent square

Notation:

$$b_i(k) = P(O_t = k \mid q_t = s_i)$$

i	$P(O_t=1 q_t=s_i)$	$P(O_t=2 q_t=s_i)$...	$P(O_t=k q_t=s_i)$...	$P(O_t=M q_t=s_i)$
1	$b_1(1)$	$b_1(2)$...	$b_1(k)$...	$b_1(M)$
2	$b_2(1)$	$b_2(2)$...	$b_2(k)$...	$b_2(M)$
3	$b_3(1)$	$b_3(2)$...	$b_3(k)$...	$b_3(M)$
:	:	:	:	:	:	:
i	$b_i(1)$	$b_i(2)$...	$b_i(k)$...	$b_i(M)$
:	:	:	:	:	:	:
N	$b_N(1)$	$b_N(2)$...	$b_N(k)$...	$b_N(M)$

What the robot sees:
Observation O_t

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

tory)

Hidden Markov Models

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

- **Question 1: State Estimation**
What is $P(q_T = S_i \mid O_1 O_2 \dots O_T)$
It will turn out that a new cute D.P. trick will get this for us.
- **Question 2: Most Probable Path**
Given $O_1 O_2 \dots O_T$, 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 $O_1 O_2 \dots O_T$, 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 → Words, Signal → 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

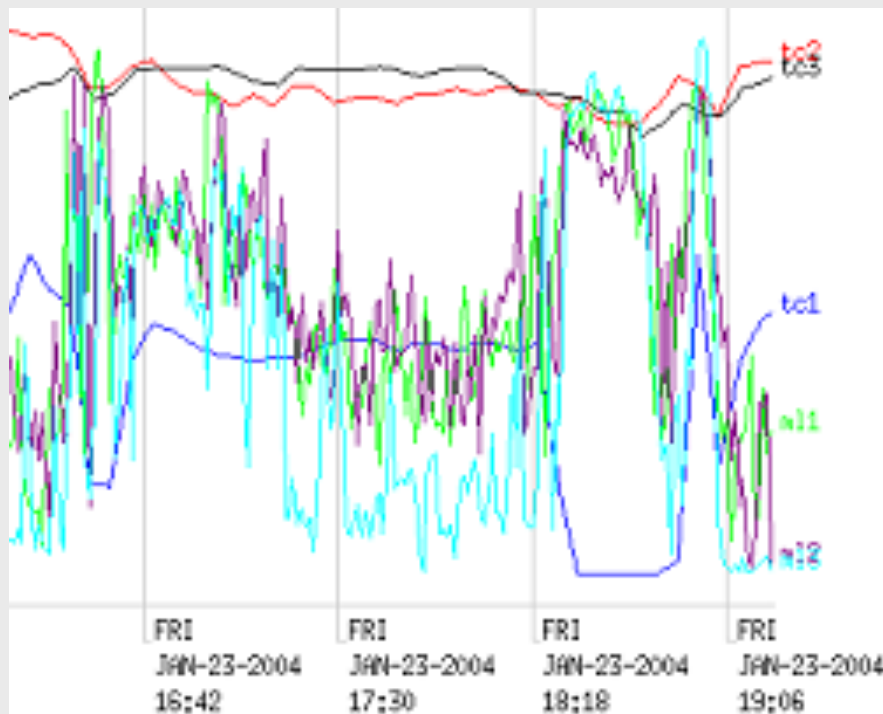
Question 1: State Estimation

What is $P(q_T = S_i \mid O_1 O_2 \dots O_t)$

Some Famous HMM Tasks

Question 1: State Estimation

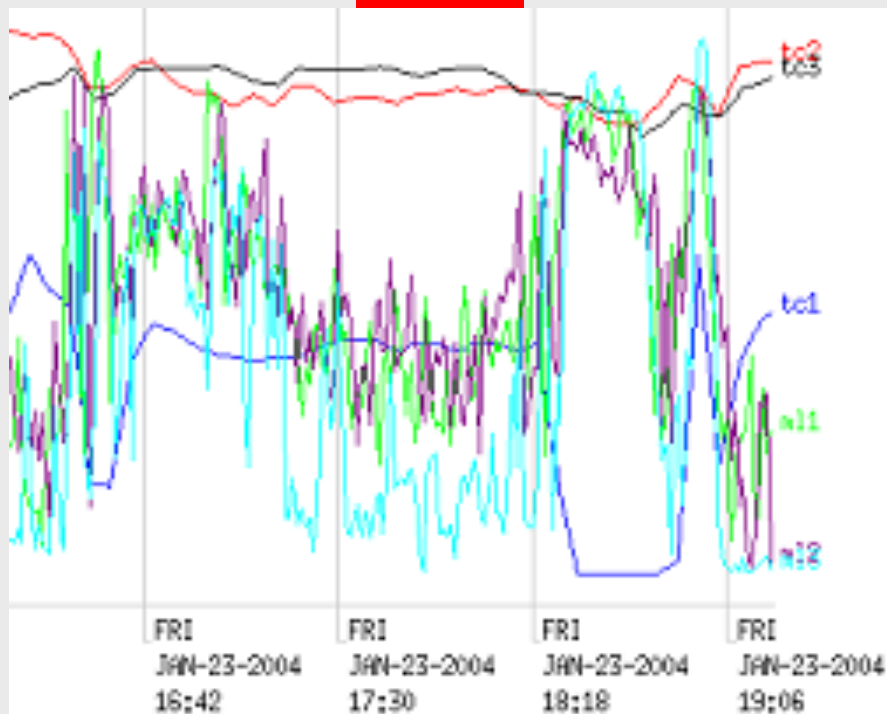
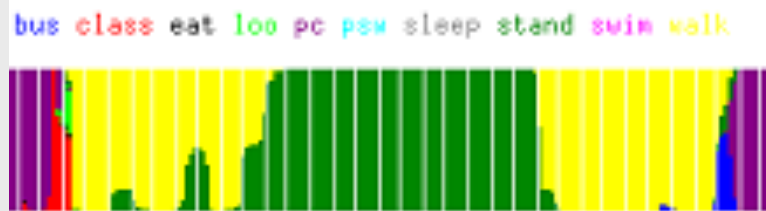
What is $P(q_T = c \mid O_1 O_2 \dots O_t)$



Some Famous HMM Tasks

Question 1: State Estimation

What is $P(q_T = c \mid O_1 O_2 \dots O_t)$



Some Famous HMM Tasks

Question 1: State Estimation

What is $P(q_T = S_i \mid O_1 O_2 \dots O_t)$

Question 2: Most Probable Path

Given $O_1 O_2 \dots O_T$, what is the most probable path that I took?

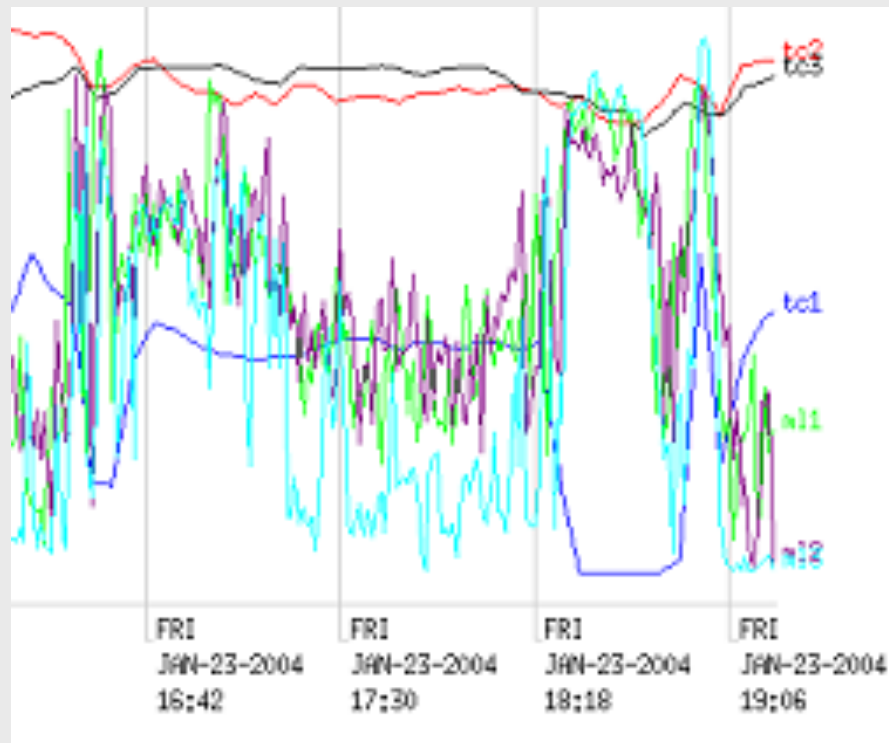
Some Famous HMM Tasks

Question 1: State Estimation

What is $P(q_T = S_i \mid O_1 O_2 \dots O_t)$

Question 2: Most Probable Path

Given $O_1 O_2 \dots O_T$, what is the most probable path that took?



Some Famous HMM Tasks

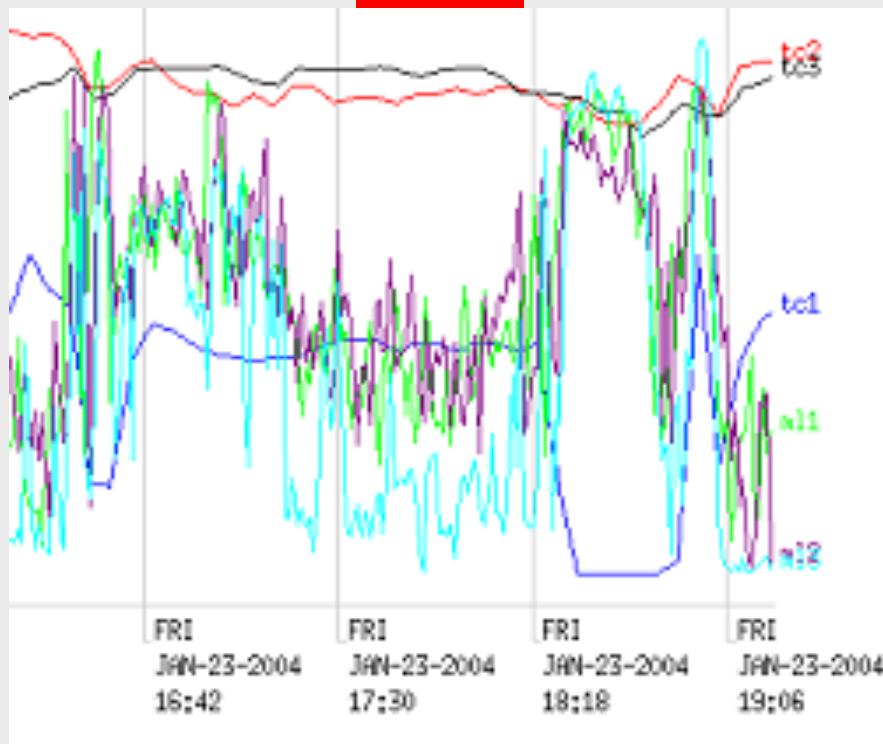
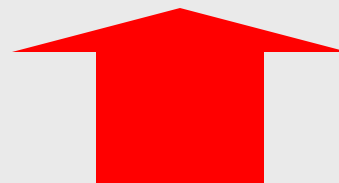
Question 1: State Estimation

What is $P(q_T = S_i \mid O_1 O_2 \dots O_t)$?

Question 2: Most Probable Path

Given $O_1 O_2 \dots O_T$, what is the most probable path that took?

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 $P(q_T = S_i \mid O_1 O_2 \dots O_t)$

Question 2: Most Probable Path

Given $O_1 O_2 \dots O_T$, what is the most probable path that I took?

Question 3: Learning HMMs:

Given $O_1 O_2 \dots O_T$, what is the maximum likelihood HMM that could have produced this string of observations?

Some Famous

Question 1: State Estimation

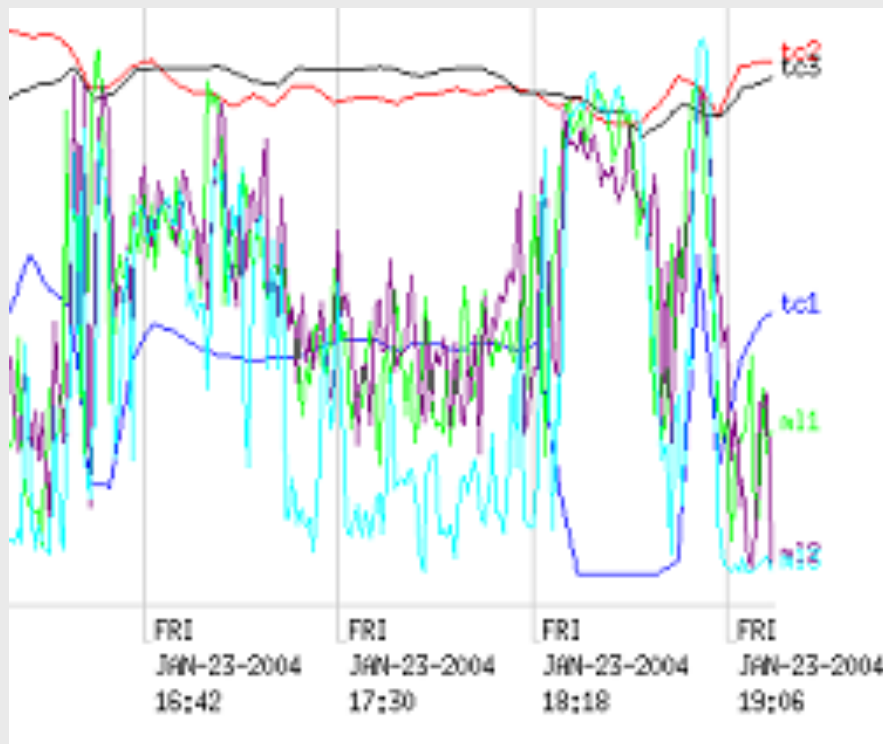
What is $P(q_T = S_i \mid O_1 O_2 \dots O_T)$?

Question 2: Most Probable Path

Given $O_1 O_2 \dots O_T$, what is the most probable path that the system took?

Question 3: Learning HMMs:

Given $O_1 O_2 \dots O_T$, what is the maximum likelihood HMM that could have produced this string of observations?



Some Famous

Question 1: State Estimation

What is $P(q_T = S_i \mid O_1 O_2 \dots O_T)$?

Question 2: Most Probable Path

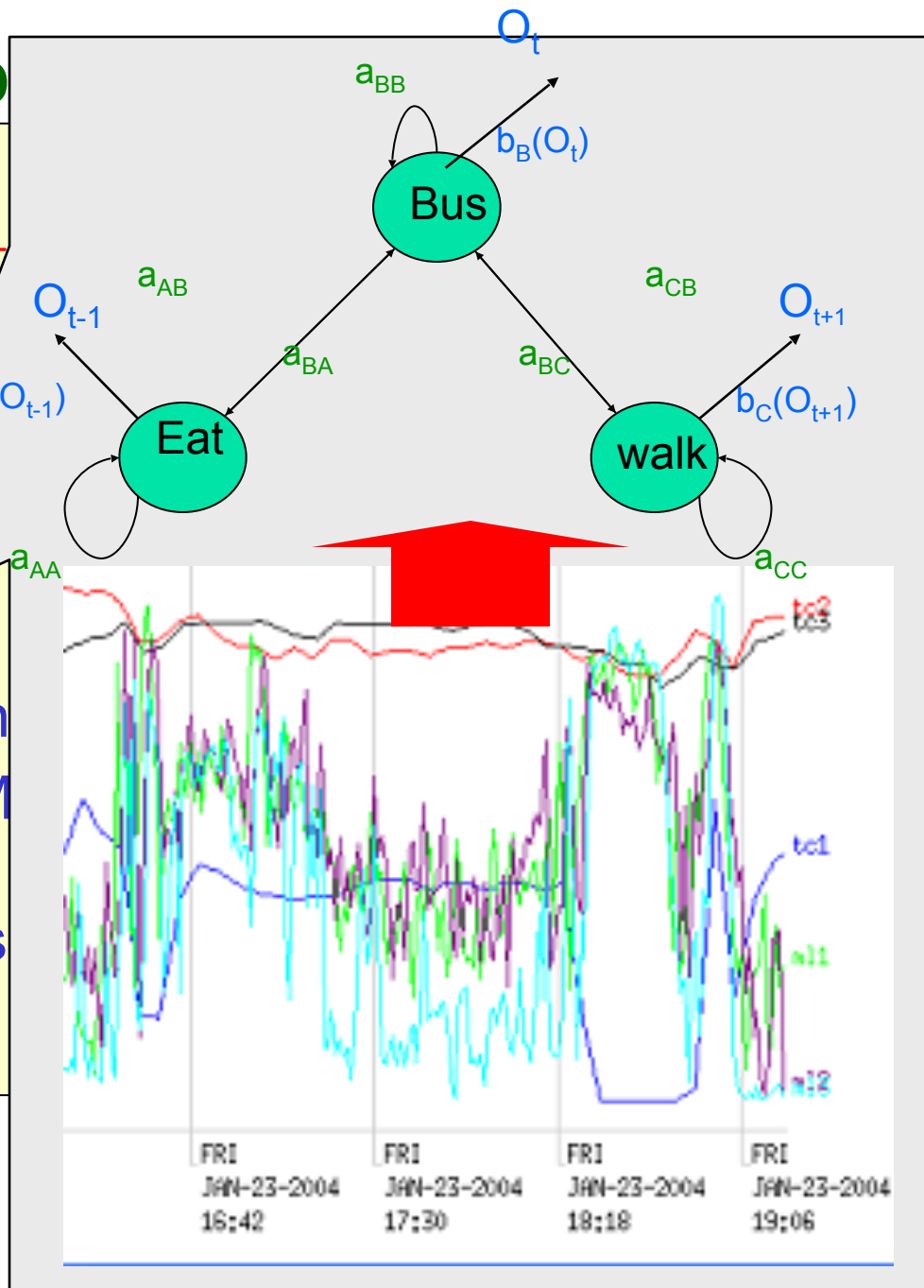
Given $O_1 O_2 \dots O_T$, what is the

most probable path that
took?

Question 3: Learning HMMs:

Given $O_1 O_2 \dots O_T$, what is the

maximum likelihood HMM
that could have produced
this string of observations



Basic Operations in HMMs

For an observation sequence $O = O_1 \dots O_T$, the three basic HMM operations are:

Problem	Algorithm	Complexity +
<i>Evaluation:</i> Calculating $P(q_t = S_i \mid O_1 O_2 \dots O_t)$	Forward-Backward	$O(TN^2)$
<i>Inference:</i> Computing $Q^* = \operatorname{argmax}_Q P(Q \mid O)$	Viterbi Decoding	$O(TN^2)$
<i>Learning:</i> Computing $\lambda^* = \operatorname{argmax}_\lambda P(O \mid \lambda)$	Baum-Welch (EM)	$O(TN^2)$

$T = \#$ timesteps, $N = \#$ states



HMM Notation (from Rabiner's Survey)

The states are labeled $S_1 S_2 \dots S_N$

*L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Vol.77, No.2, pp.257--286, 1989.

Available from

<http://ieeexplore.ieee.org/iel5/5/698/00018626.pdf?arnumber=18626>

For a particular trial....

Let T be the number of observations

T is also the number of states passed through

$O = O_1 O_2 \dots O_T$ is the sequence of observations

$Q = q_1 q_2 \dots q_T$ 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, λ , is a 5-tuple consisting of

- N the number of states
- M the number of possible observations
- $\{\pi_1, \pi_2, \dots, \pi_N\}$ The starting state probabilities

$$P(q_0 = S_i) = \pi_i$$

This is new. In our previous example, start state was deterministic

- $$\begin{matrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{matrix}$$

The state transition probabilities

$$P(q_{t+1}=S_j \mid q_t=S_i)=a_{ij}$$

- $$\begin{matrix} b_1(1) & b_1(2) & \dots & b_1(M) \\ b_2(1) & b_2(2) & \dots & b_2(M) \\ \vdots & \vdots & & \vdots \\ b_N(1) & b_N(2) & \dots & b_N(M) \end{matrix}$$

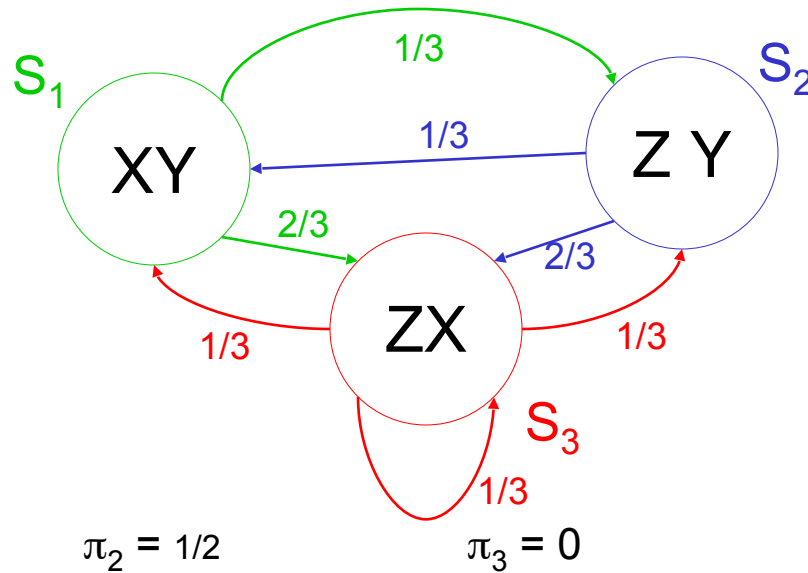
The observation probabilities

$$P(O_t=k \mid q_t=S_i)=b_i(k)$$

Here's an HMM

Start randomly in state 1 or 2

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



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3 \quad a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1 (1/2 = (X$$

$$b_1 (Y) = 1/2$$

$$b_1 (Z) = 0$$

$$b_2 (X) = 0 \quad b_2 (Y) = 1/2$$

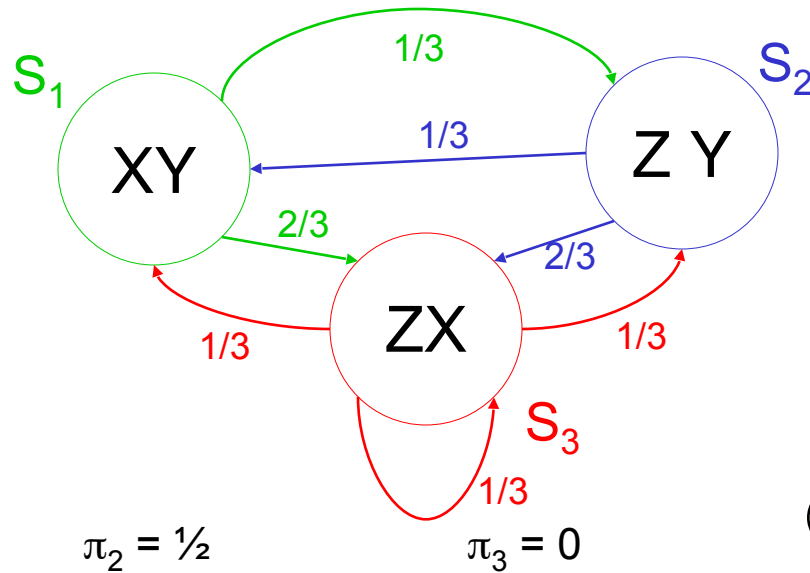
$$b_2 (Z) = 1/2$$

$$b_3 (X) = 1/2$$

$$b_3 (Y) = 0$$

$$b_3 (Z) = 1/2$$

Here's an HMM



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

$$b_3(X) = 1/2$$

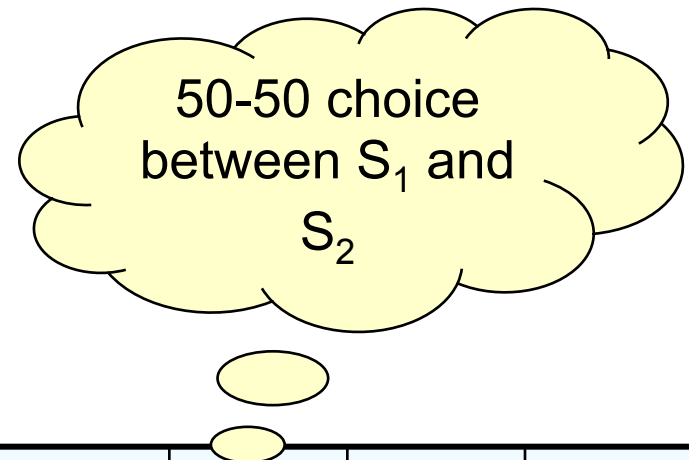
$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

Start randomly in state 1 or 2

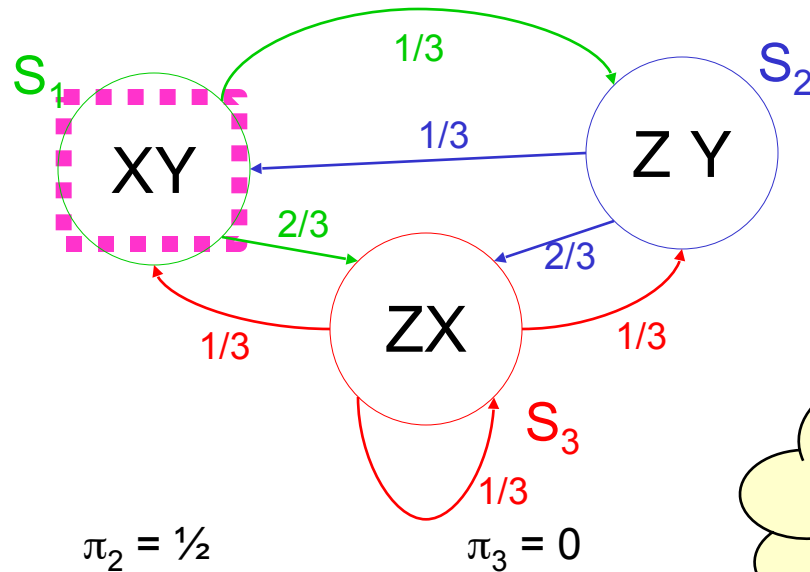
Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:



$q_0 =$	<u>0</u>	$O_0 =$	—
$q_1 =$	—	$O_1 =$	—
$q_2 =$	—	$O_2 =$	—

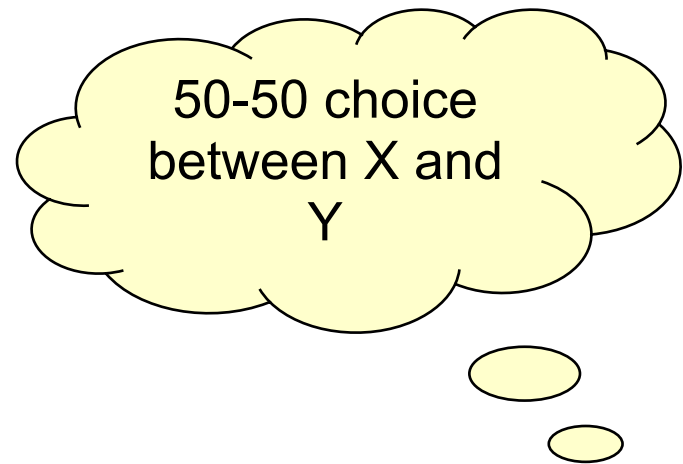
Here's an HMM



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:



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

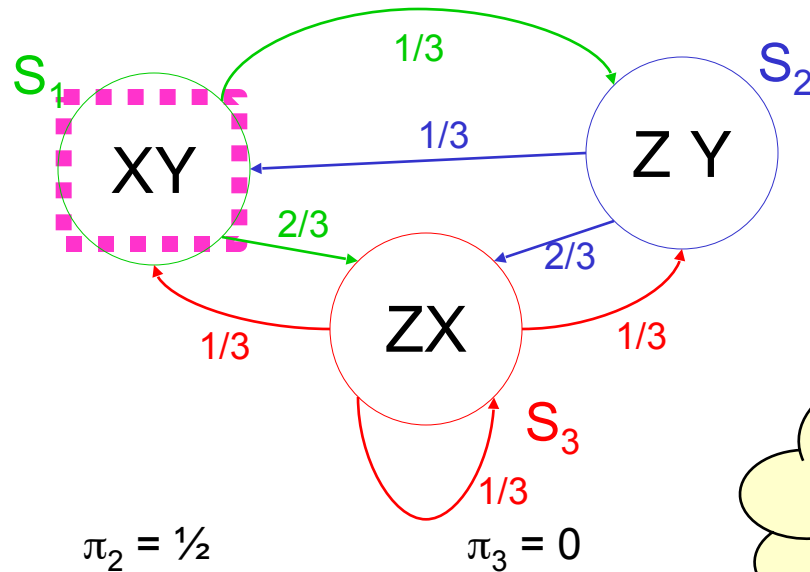
$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

$q_0 =$	S_1	$O_0 =$	<u> </u>
$q_1 =$	<u> </u>	$O_1 =$	<u> </u>
$q_2 =$	<u> </u>	$O_2 =$	<u> </u>

Here's an HMM



$N = 3$
 $M = 3$
 $\pi_1 = 1/2$

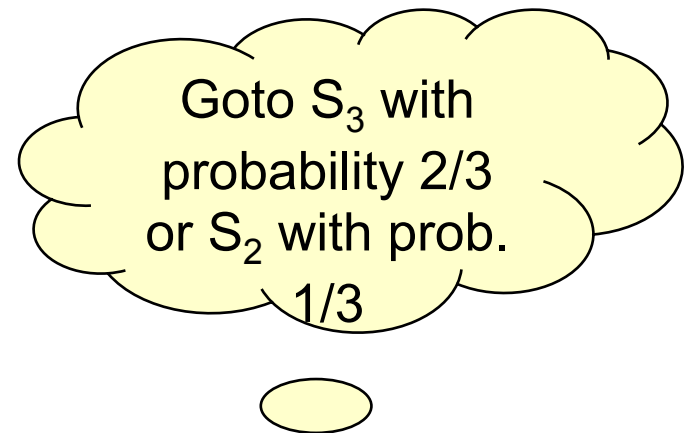
$a_{11} = 0$
 $a_{12} = 1/3$
 $a_{13} = 1/3$

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

Start randomly in state 1 or 2

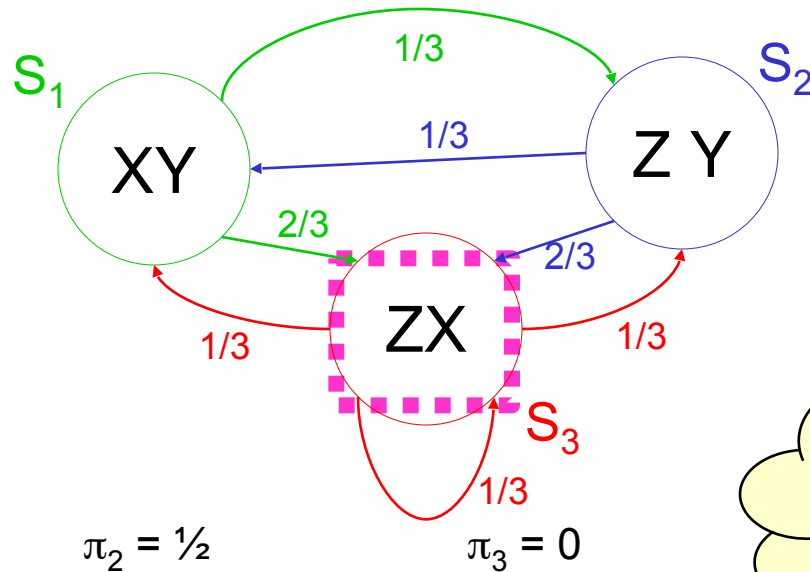
Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:



$q_0 =$	S_1	$O_0 =$	X
$q_1 =$	—	$O_1 =$	—
$q_2 =$	—	$O_2 =$	—

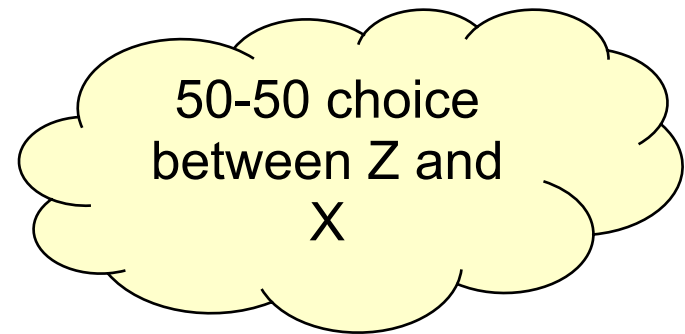
Here's an HMM



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:



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

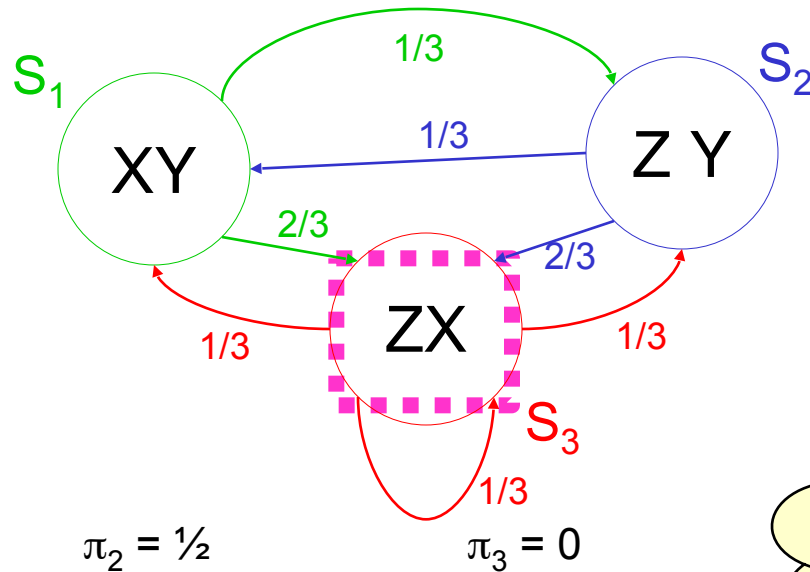
$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

$q_0 =$	S_1	$O_0 =$	X
$q_1 =$	S_3	$O_1 =$	—
$q_2 =$	—	$O_2 =$	—

Here's an HMM



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

Start randomly in state 1 or 2

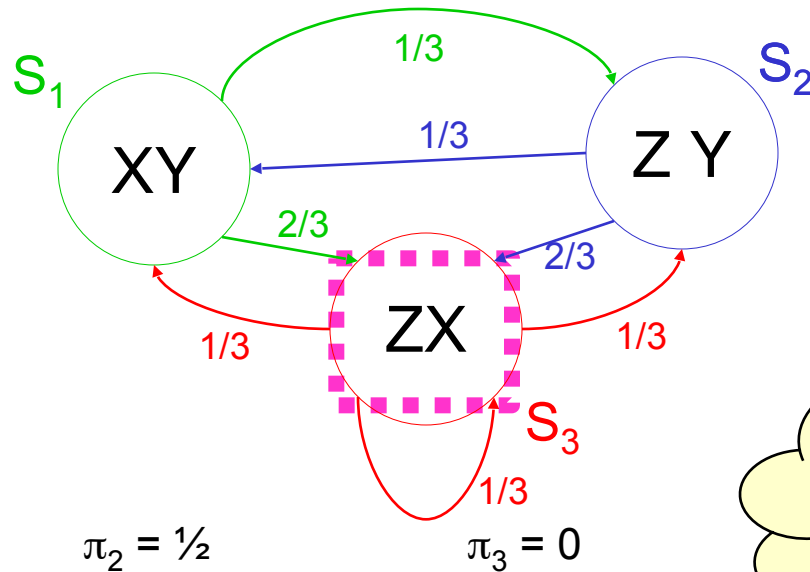
Choose one of the output symbols in each state at random.

Let's generate a sequence of observations:

Each of the three next states is equally likely

$q_0 =$	S_1	$O_0 =$	X
$q_1 =$	S_3	$O_1 =$	X
$q_2 =$	—	$O_2 =$	—

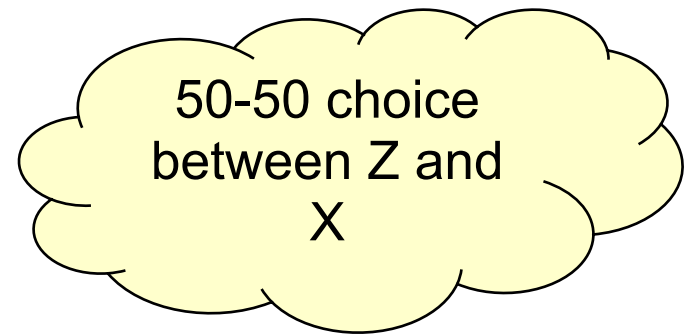
Here's an HMM



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:



$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

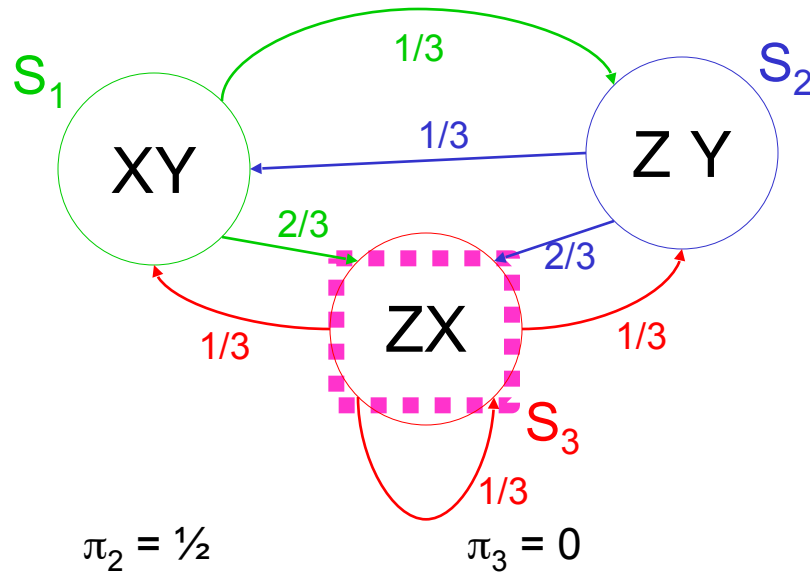
$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

$q_0 =$	S_1	$O_0 =$	X
$q_1 =$	S_3	$O_1 =$	X
$q_2 =$	S_3	$O_2 =$	—

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:

$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

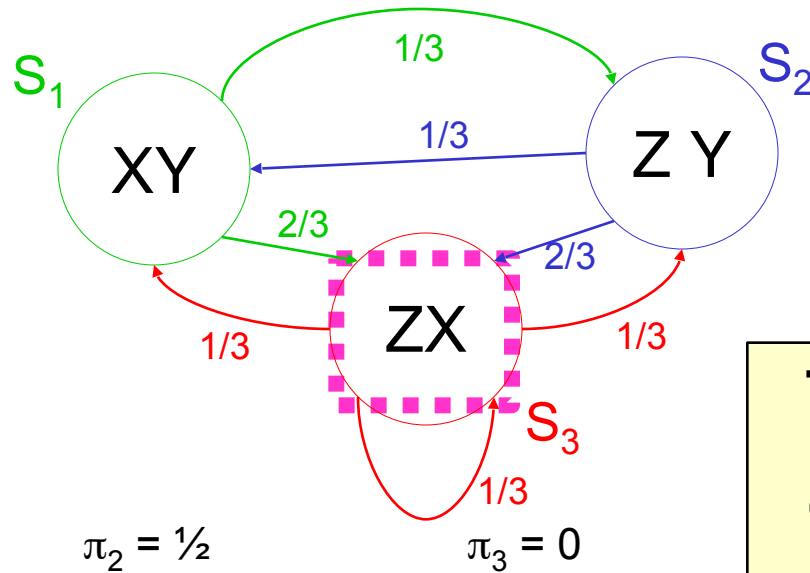
$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

$q_0 =$	S_1	$O_0 =$	X
$q_1 =$	S_3	$O_1 =$	X
$q_2 =$	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...

$$N = 3$$

$$M = 3$$

$$\pi_1 = 1/2$$

$$\pi_2 = 1/2$$

$$\pi_3 = 0$$

$$a_{11} = 0$$

$$a_{12} = 1/3$$

$$a_{13} = 2/3$$

$$a_{12} = 1/3$$

$$a_{22} = 0$$

$$a_{13} = 2/3$$

$$a_{13} = 1/3$$

$$a_{32} = 1/3$$

$$a_{13} = 1/3$$

$$b_1(X) = 1/2$$

$$b_1(Y) = 1/2$$

$$b_1(Z) = 0$$

$$b_2(X) = 0 \quad b_2(Y) = 1/2$$

$$b_2(Z) = 1/2$$

$$b_3(X) = 1/2$$

$$b_3(Y) = 0$$

$$b_3(Z) = 1/2$$

$q_0 =$?	$O_0 =$	X
$q_1 =$?	$O_1 =$	X
$q_2 =$?	$O_2 =$	Z

Prob. of a series of observations

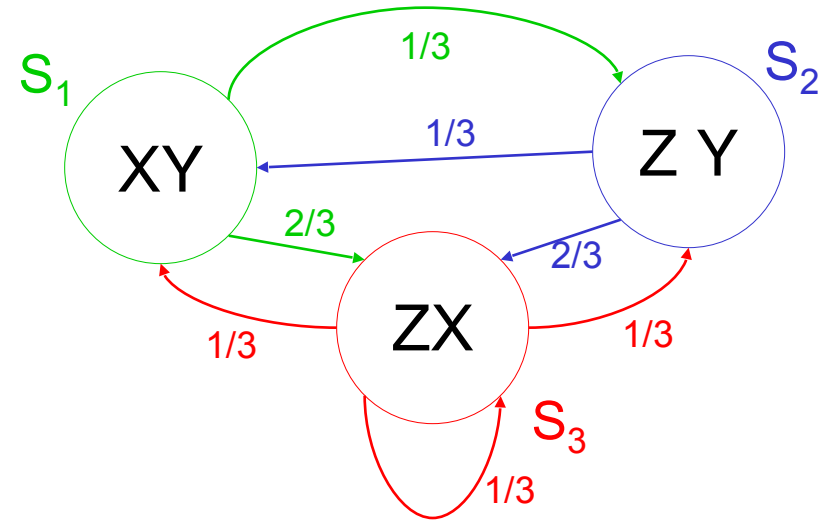
What is $P(\mathbf{O}) = P(O_1 O_2 O_3) =$
 $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?

Slow, stupid way:

$$\begin{aligned} P(\mathbf{O}) &= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} \wedge \mathbf{Q}) \\ &= \sum_{\mathbf{Q} \in \text{Paths of length 3}} P(\mathbf{O} | \mathbf{Q}) P(\mathbf{Q}) \end{aligned}$$

How do we compute $P(\mathbf{Q})$ for
an arbitrary path \mathbf{Q} ?

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



Prob. of a series of observations

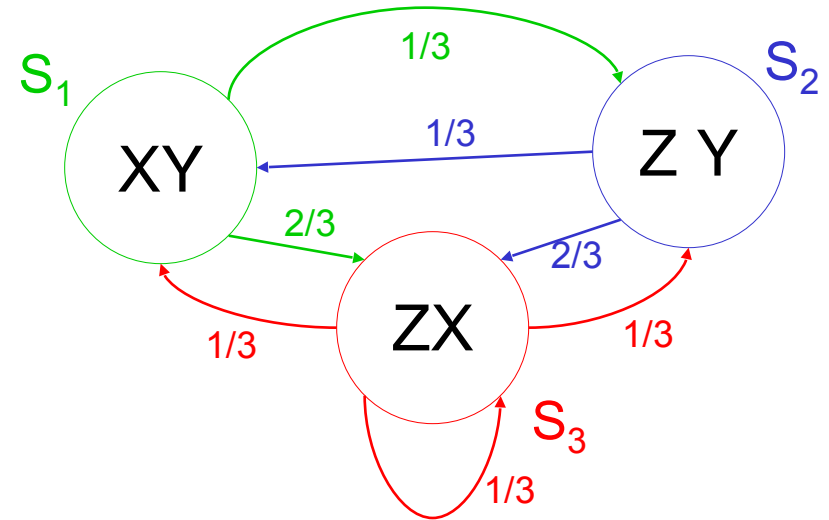
What is $P(\mathbf{O}) = P(O_1 O_2 O_3) =$
 $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?

Slow, stupid way:

$$\begin{aligned} P(\mathbf{O}) &= \sum_{Q \in \text{Paths of length 3}} P(\mathbf{O} \wedge Q) \\ &= \sum_{Q \in \text{Paths of length 3}} P(\mathbf{O} | Q) P(Q) \end{aligned}$$

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

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



$$P(Q) = P(q_1, q_2, q_3)$$

$$= P(q_1) P(q_2, q_3 | q_1) \text{ (chain rule)}$$

$$= P(q_1) P(q_2 | q_1) P(q_3 | q_2, q_1) \text{ (chain)}$$

$$= P(q_1) P(q_2 | q_1) P(q_3 | q_2) \text{ (why?)}$$

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

$$= 1/2 * 2/3 * 1/3 = 1/9$$

Prob. of a series of observations

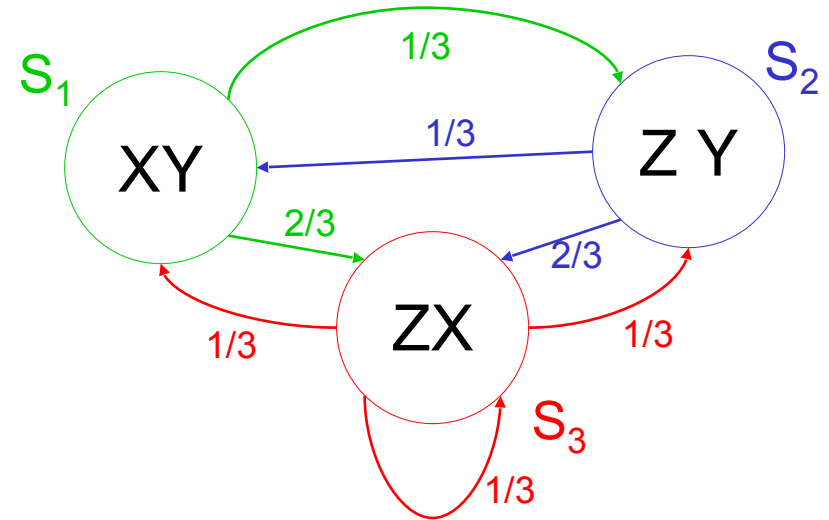
What is $P(\mathbf{O}) = P(O_1 O_2 O_3) =$
 $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?

Slow, stupid way:

$$\begin{aligned} P(\mathbf{O}) &= \sum_{Q \in \text{Paths of length 3}} P(\mathbf{O} \wedge Q) \\ &= \sum_{Q \in \text{Paths of length 3}} P(\mathbf{O} | Q) P(Q) \end{aligned}$$

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

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



$P(\mathbf{O}|Q)$

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

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

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

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

$= 1/2 * 1/2 * 1/2 = 1/8$

Prob. of a series of observations

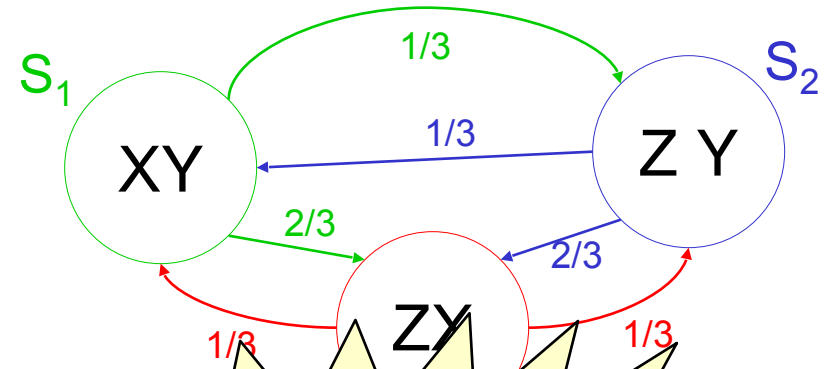
What is $P(\mathbf{O}) = P(O_1 O_2 O_3) =$
 $P(O_1 = X \wedge O_2 = X \wedge O_3 = Z)$?

Slow, stupid way:

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

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

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



$P(\mathbf{O})$ would need 27 $P(Q)$
computations and 27 $P(O|Q)$
computations

A sequence of 20 observations would need $3^{20} =$
3.5 billion computations and 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 $O_1 O_2 \dots O_T$

Define

$$\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i \mid \lambda) \quad \text{where } 1 \leq t \leq T$$

$\alpha_t(i)$ = 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

$\alpha_t(i) = P(O_1 O_2 \dots O_T \wedge q_t = S_i \mid \lambda)$ ($\alpha_t(i)$ can be defined stupidly by considering all paths length “t”. How?)

$$\begin{aligned}\alpha_1(i) &= P(O_1 \wedge q_1 = S_i) \\ &= P(q_1 = S_i) P(O_1 \mid q_1 = S_i) \\ &= \text{what?} \\ \alpha_{t+1}(j) &= P(O_1 O_2 \dots O_t O_{t+1} \wedge q_{t+1} = S_j) \\ &= \end{aligned}$$

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

$\alpha_t(i) = P(O_1 O_2 \dots O_T \wedge q_t = S_i \mid \lambda)$ ($\alpha_t(i)$ can be defined stupidly by considering all paths length “t”. How?)

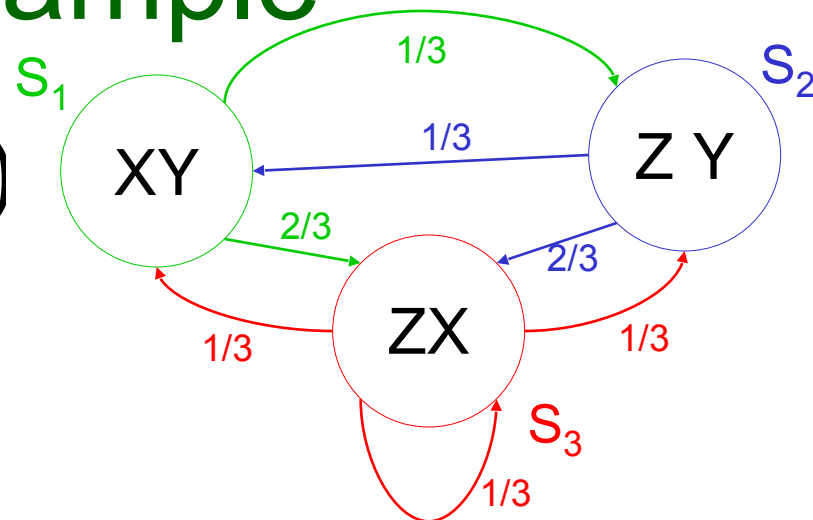
$$\begin{aligned}
 \alpha_1(i) &= P(O_1 \wedge q_1 = S_i) \\
 &= P(q_1 = S_i) P(O_1 | q_1 = S_i) \\
 &= \text{what?} \\
 \alpha_{t+1}(j) &= P(O_1 O_2 \dots O_t O_{t+1} \wedge q_{t+1} = S_j) \\
 &= \sum_{i=1}^N P(O_1 O_2 \dots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j) \\
 &= \sum_{i=1}^N P(O_{t+1}, q_{t+1} = S_j | O_1 O_2 \dots O_t \wedge q_t = S_i) P(O_1 O_2 \dots O_t \wedge q_t = S_i) \\
 &= \sum_i P(O_{t+1}, q_{t+1} = S_j | q_t = S_i) \alpha_t(i) \\
 &= \sum_i P(q_{t+1} = S_j | q_t = S_i) P(O_{t+1} | q_{t+1} = S_j) \alpha_t(i) \\
 &= \sum_i a_{ij} b_j(O_{t+1}) \alpha_t(i)
 \end{aligned}$$

in our example

$$\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i | \lambda)$$

$$\alpha_1(i) = b_i(O_1) \pi_i$$

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



WE SAW $O_1 O_2 O_3 = X X Z$

$$\alpha_1(1) = \frac{1}{4}$$

$$\alpha_1(2) = 0$$

$$\alpha_1(3) = 0$$

$$\alpha_2(1) = 0$$

$$\alpha_2(2) = 0$$

$$\alpha_2(3) = \frac{1}{12}$$

$$\alpha_3(1) = 0$$

$$\alpha_3(2) = \frac{1}{72}$$

$$\alpha_3(3) = \frac{1}{72}$$

Easy Question

We can cheaply compute

$$\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i)$$

(How) can we cheaply compute

$$P(O_1 O_2 \dots O_t) \quad ?$$

(How) can we cheaply compute

$$P(q_t = S_i | O_1 O_2 \dots O_t)$$

Easy Question

We can cheaply compute

$$\alpha_t(i) = P(O_1 O_2 \dots O_t \wedge q_t = S_i)$$

(How) can we cheaply compute

$$P(O_1 O_2 \dots O_t) \quad ?$$

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

(How) can we cheaply compute

$$P(q_t = S_i | O_1 O_2 \dots O_t)$$

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

Most probable path given observations

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

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

Slow, stupid answer :

$$\begin{aligned} & \underset{Q}{\operatorname{argmax}} P(Q|O_1O_2...O_T) \\ &= \underset{Q}{\operatorname{argmax}} \frac{P(O_1O_2...O_T|Q)P(Q)}{P(O_1O_2...O_T)} \\ &= \underset{Q}{\operatorname{argmax}} P(O_1O_2...O_T|Q)P(Q) \end{aligned}$$

Efficient MPP computation

We're going to compute the following variables:

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

= 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 $O_1 \dots O_t$

DEFINE: $mpp_t(i) =$ that path

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

The Viterbi Algorithm

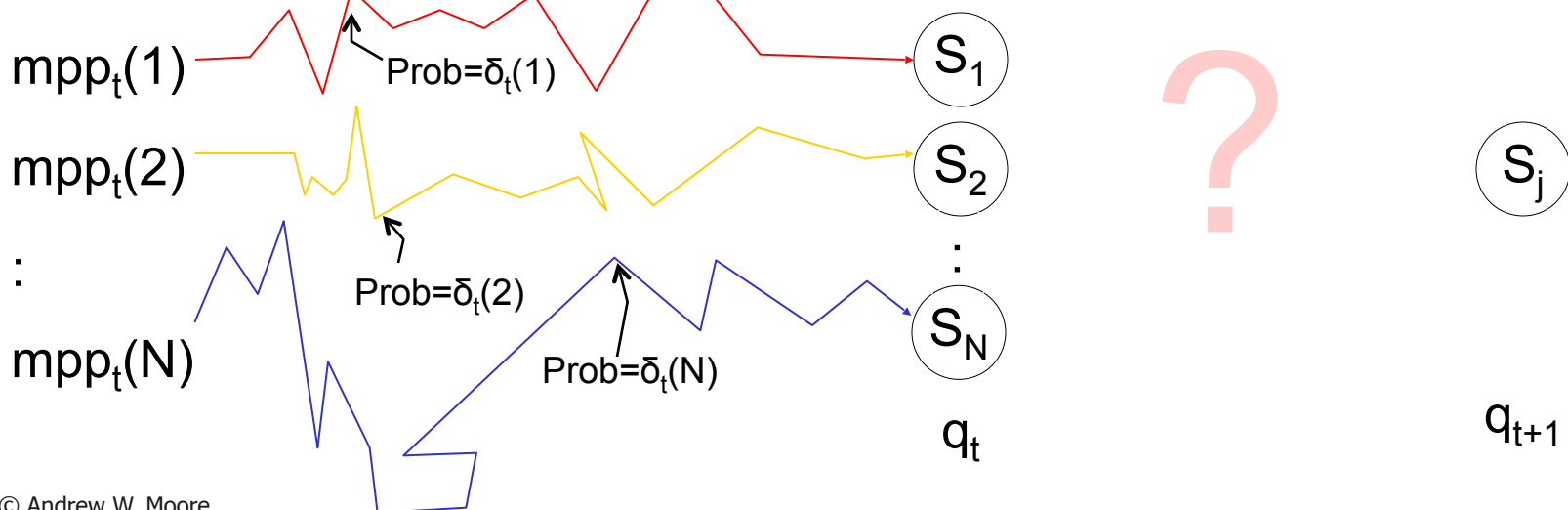
$$\delta_t(i) = \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 \dots O_t)$$

$$mpp_t(i) = \arg \max_{q_1 q_2 \dots q_{t-1}} P(q_1 q_2 \dots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 \dots O_t)$$

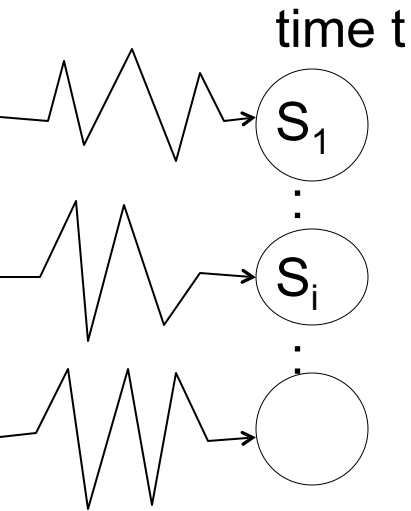
$$\begin{aligned} \delta_1(i) &= \max_{\text{one choice}} P(q_1 = S_i \wedge O_1) \\ &= P(q_1 = S_i) P(O_1 | q_1 = S_i) \\ &= \pi_i b_i(O_1) \end{aligned}$$

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 $mpp_{t+1}(j)$?



The Viterbi Algorithm



time t+1

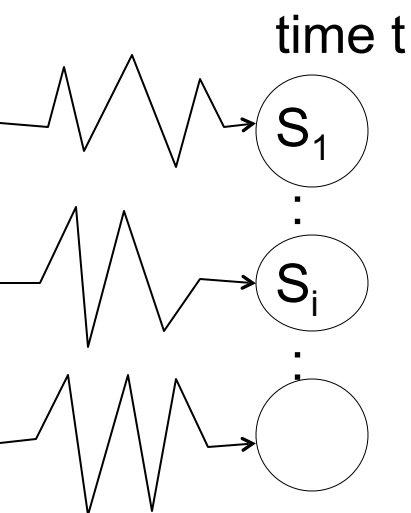


The most prob path with last two
states S_i S_j

is

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

The Viterbi Algorithm



time t+1



The most prob path with last two states S_i S_j

is

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

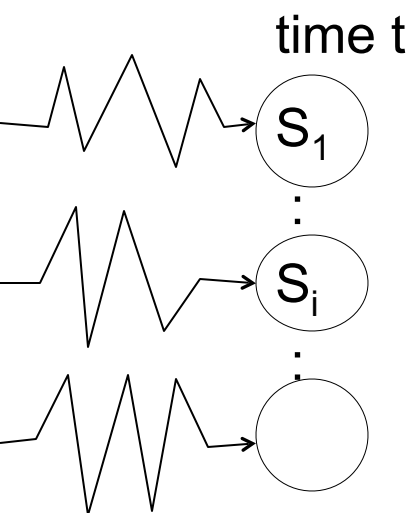
What is the prob of that path?

$$\begin{aligned} & \delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1} \mid \lambda) \\ = & \delta_t(i) a_{ij} b_j(O_{t+1}) \end{aligned}$$

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

$$\text{where } i^* = \underset{i}{\operatorname{argmax}} \delta_t(i) a_{ij} b_j(O_{t+1})$$

The Viterbi Algorithm



time t+1



The most prob path with last two states S_i S_j

is

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

What is the prob of that path?

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

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

SO The most probable path
as its penultimate state

$$\text{where } i^* = \underset{i}{\operatorname{argmax}} \delta_t(i) a_{ij} b_j(O_{t+1})$$

Summary:

$$\left. \begin{aligned} \delta_{t+1}(j) &= \delta_t(i^*) a_{ij} b_j(O_{t+1}) \\ \text{mpp}_{t+1}(j) &= \text{mpp}_{t+1}(i^*) S_{i^*} \end{aligned} \right\} \text{ with } i^* \text{ defined to the left}$$

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

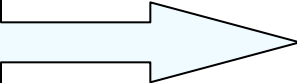
In practice: many levels of inference; not one big jump.

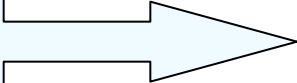
HMMs are used and useful

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. $O_1 O_2 \dots O_T$ with a big “T”.

Observations previously
in lecture  $O_1 O_2 \dots O_T$

Observations in the
next bit  $O_1 O_2 \dots O_T$

Inferring an HMM

Remember, we've been doing things like

$$P(O_1 O_2 \dots O_T | \lambda)$$

That “ λ ” is the notation for our HMM parameters.

Now We have some observations and we want to estimate λ from them.

AS USUAL: We could use

(i) MAX LIKELIHOOD $\lambda = \underset{\lambda}{\operatorname{argmax}} P(O_1 \dots O_T | \lambda)$

(ii) BAYES

Work out $P(\lambda | O_1 \dots O_T)$

and then take $E[\lambda]$ or $\max_{\lambda} P(\lambda | O_1 \dots O_T)$

Max likelihood HMM estimation

Define

$$\gamma_t(i) = P(q_t = S_i \mid O_1 O_2 \dots O_T, \lambda)$$

$$\varepsilon_t(i,j) = P(q_t = S_i \wedge q_{t+1} = S_j \mid O_1 O_2 \dots O_T, \lambda)$$

$\gamma_t(i)$ and $\varepsilon_t(i,j)$ can be computed efficiently $\forall i,j,t$
(Details in Rabiner paper)

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{Expected number of transitions out of state } i \text{ during the path}$$

$$\sum_{t=1}^{T-1} \varepsilon_t(i,j) = \text{Expected number of transitions from state } i \text{ to state } j \text{ during the path}$$

$$\gamma_t(i) = P(q_t = S_i | O_1 O_2 \dots O_T, \lambda)$$

$$\varepsilon_t(i, j) = P(q_t = S_i \wedge q_{t+1} = S_j | O_1 O_2 \dots O_T, \lambda)$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions out of state } i \text{ during path}$$

$$\sum_{t=1}^{T-1} \varepsilon_t(i, j) = \text{expected number of transitions out of } i \text{ and into } j \text{ during path}$$

HMM estimation

Notice
$$\frac{\sum_{t=1}^{T-1} \varepsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\left(\begin{array}{c} \text{expected frequency} \\ i \rightarrow j \end{array} \right)}{\left(\begin{array}{c} \text{expected frequency} \\ i \end{array} \right)}$$

= Estimate of Prob(Next state S_j | This state S_i)

We can re - estimate

$$a_{ij} \leftarrow \frac{\sum_{t=1}^{T-1} \varepsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

We can also re - estimate

$$b_j(O_k) \leftarrow \boxed{?} \quad (\text{See Rabiner})$$

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

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

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

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

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

$$= \frac{\sum_{t=1}^T P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}{\sum_{k=1}^N \sum_{t=1}^T P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}$$

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

$$\begin{aligned}
 &= \frac{\text{Expected \# transitions } i \rightarrow j \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T}{\sum_{k=1}^N \text{Expected \# transitions } i \rightarrow k \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T} \\
 &= \frac{\sum_{t=1}^T P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}{\sum_{k=1}^N \sum_{t=1}^T P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{S_{ij}}{\sum_{k=1}^N S_{ik}} \text{ where } S_{ij} = \sum_{t=1}^T P(q_{t+1} = s_j, q_t = s_i, O_1, \boxed{?} O_T \mid \lambda^{\text{old}}) \\
 &= \text{What?}
 \end{aligned}$$

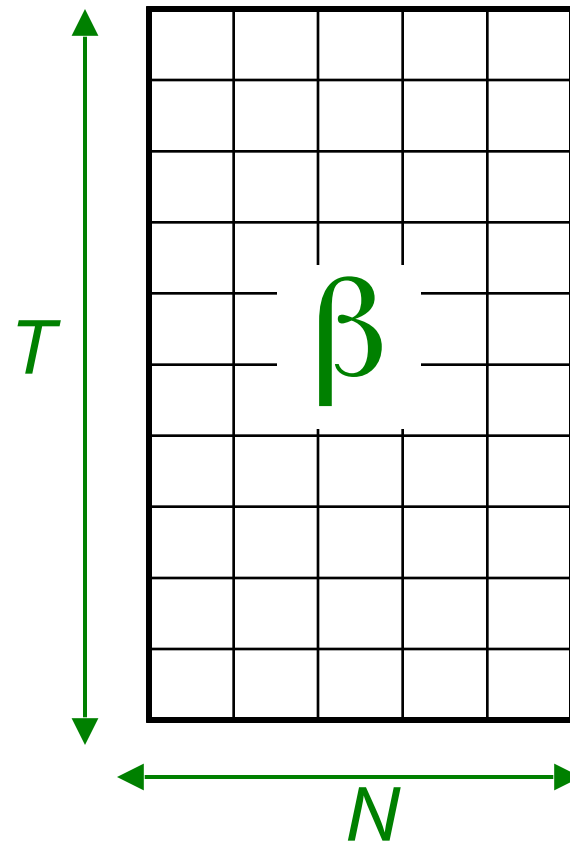
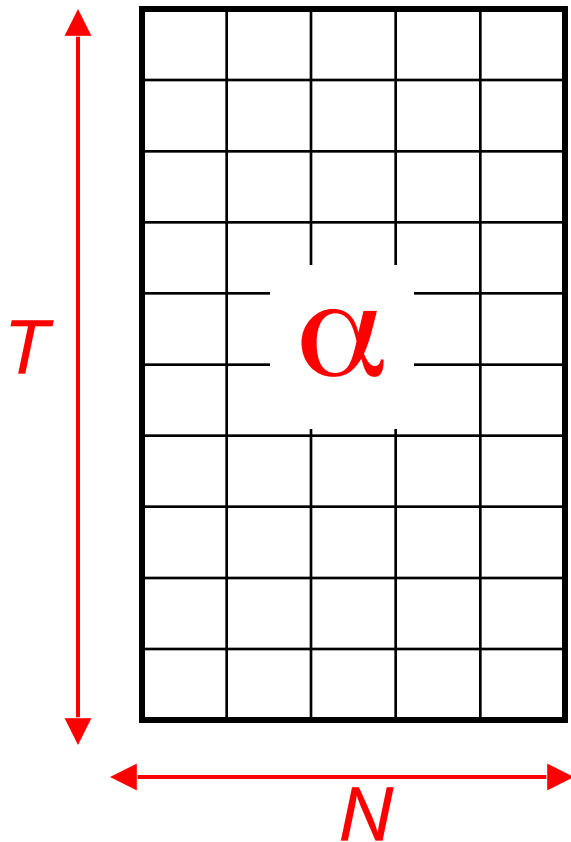
We want a_{ij}^{new} = new estimate of $P(q_{t+1} = s_j \mid q_t = s_i)$

$$\begin{aligned}
 &= \frac{\text{Expected \# transitions } i \rightarrow j \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T}{\sum_{k=1}^N \text{Expected \# transitions } i \rightarrow k \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T} \\
 &= \frac{\sum_{t=1}^T P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}{\sum_{k=1}^N \sum_{t=1}^T P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \boxed{?} O_T)}
 \end{aligned}$$

$$\begin{aligned}
 &= \frac{S_{ij}}{\sum_{k=1}^N S_{ik}} \text{ where } S_{ij} = \sum_{t=1}^T P(q_{t+1} = s_j, q_t = s_i, O_1, \boxed{?} O_T \mid \lambda^{\text{old}}) \\
 &\quad = a_{ij} \sum_{t=1}^T \alpha_t(i) \beta_{t+1}(j) b_j(O_{t+1})
 \end{aligned}$$

We want $a_{ij}^{\text{new}} = S_{ij} / \sum_{k=1}^N S_{ik}$ where $S_{ij} = a_{ij} \sum_{t=1}^T \alpha_t(i) \beta_{t+1}(j) b_j(O_{t+1})$

We want $a_{ij}^{\text{new}} = S_{ij} / \sum_{k=1}^N S_{ik}$ 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 λ 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 $O_1 \dots O_T$
 2. Guess your first λ estimate $\lambda(0)$, $k=0$
 3. $k = k+1$
 4. Given $O_1 \dots O_T$, $\lambda(k)$ compute $\gamma_t(i)$
 $\epsilon_t(i,j) \quad \forall 1 \leq t \leq T, \quad \forall 1 \leq i \leq N, \quad \forall 1 \leq j \leq N$
 5. Compute expected freq. of state i , and expected freq. $i \rightarrow j$
 6. Compute new estimates of a_{ij} , $b_j(k)$, π_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).

- There are lots of

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')

- The local minimum is often a bad one for 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

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.

*L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. of the IEEE, Vol.77, No.2, pp.257--286, 1989.

<http://ieeexplore.ieee.org/iel5/5/698/00018626.pdf?arnumber=18626>