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
Associate Professor
School of Computer Science
Carnegie Mellon University
www.cs.cmu.edu/~awm
awm@cs.cmu.edu
412-268-7599

Copyright © 2001, Andrew W. Moore

Nov 29th, 2001

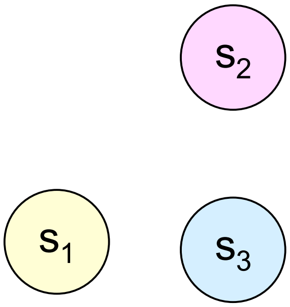
Announcements

- Hand in reviews
- Receive back reviews
- Papers look quite impressive
- Why do we move in the direction of the gradient in gradient descent? (What are the units?)

A Markov System

Has N states, called s_1, s_2 .. s_N

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



$N = 3$

$t=0$

Copyright © 2001, Andrew W. Moore

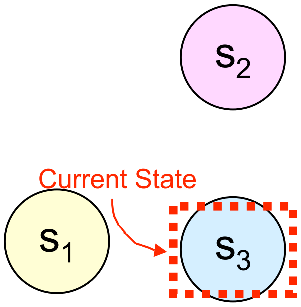
A Markov System

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 □ $\{s_1, s_2 .. s_N\}$



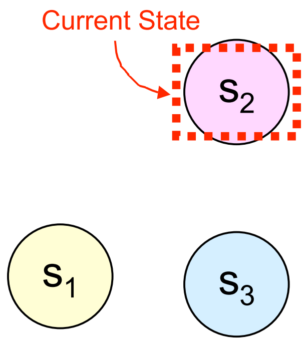
$N = 3$

$t=0$

$q_t=q_0=s_3$

Copyright © 2001, Andrew W. Moore

A Markov System



N = 3

t=1

$q_t=q_1=s_2$

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

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

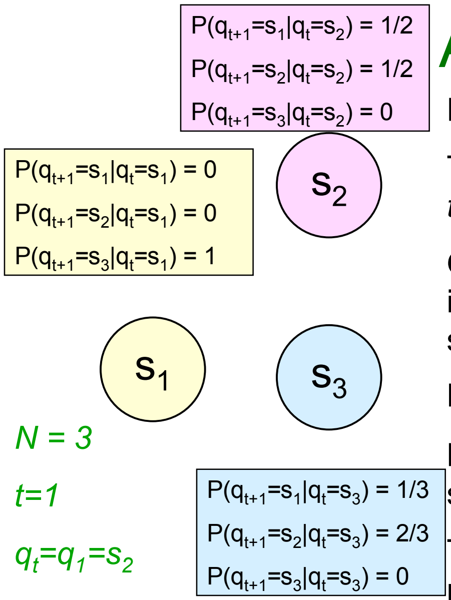
On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: q_t □ $\{s_1, s_2 .. s_N\}$

Between each timestep, the next state is chosen randomly.

Copyright © 2001, Andrew W. Moore

A Markov System



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

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

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

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

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

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

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

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

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

N = 3

t=1

$q_t=q_1=s_2$

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

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

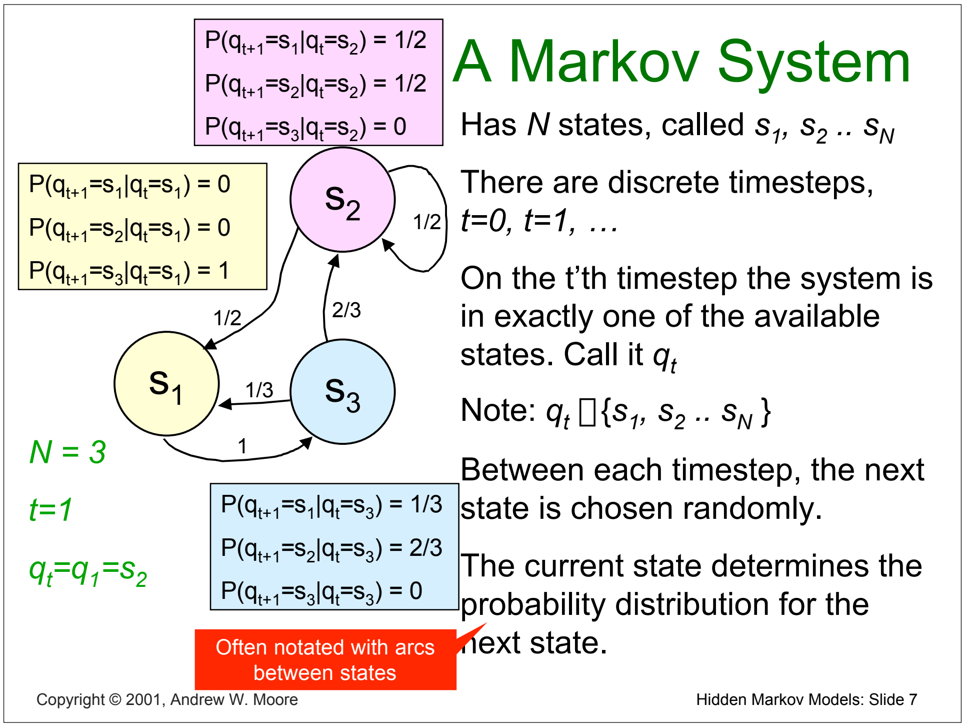
On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: q_t □ $\{s_1, s_2 .. s_N\}$

Between each timestep, the next state is chosen randomly.

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

Copyright © 2001, Andrew W. Moore



A Markov System

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

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

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

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

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

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

N = 3

t=1

$q_t=q_1=s_2$

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

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

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

Often notated with arcs between states

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

There are discrete timesteps, *t=0, t=1, …*

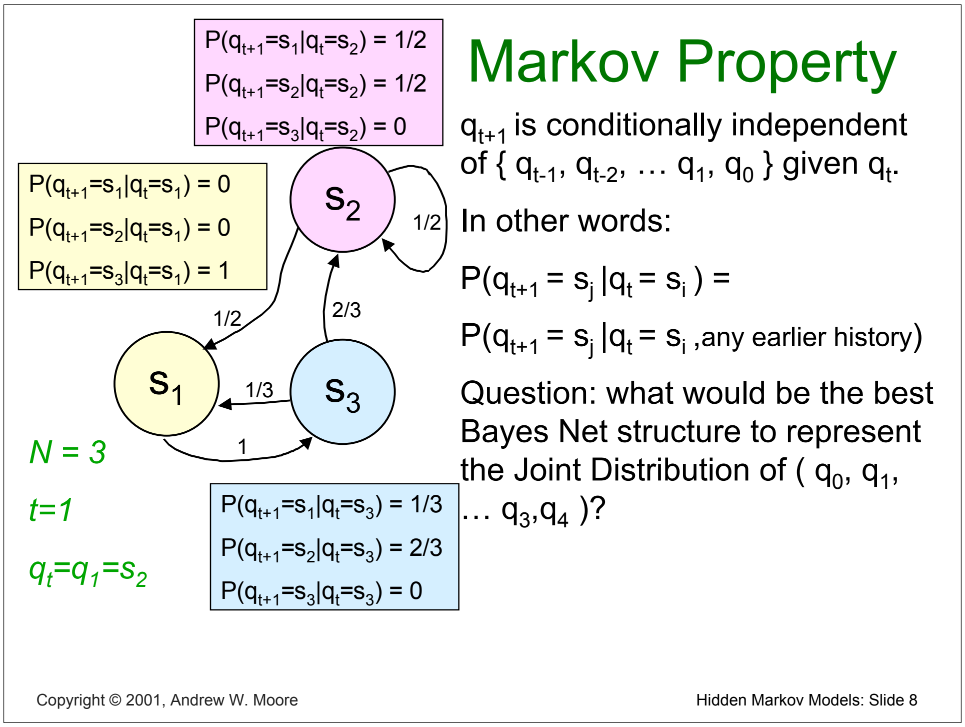
On the t'th timestep the system is in exactly one of the available states. Call it q_t

Note: q_t □ $\{s_1, s_2 .. s_N\}$

Between each timestep, the next state is chosen randomly.

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

Copyright © 2001, Andrew W. Moore



Markov Property

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

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

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

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

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

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

N = 3

t=1

$q_t=q_1=s_2$

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

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

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

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

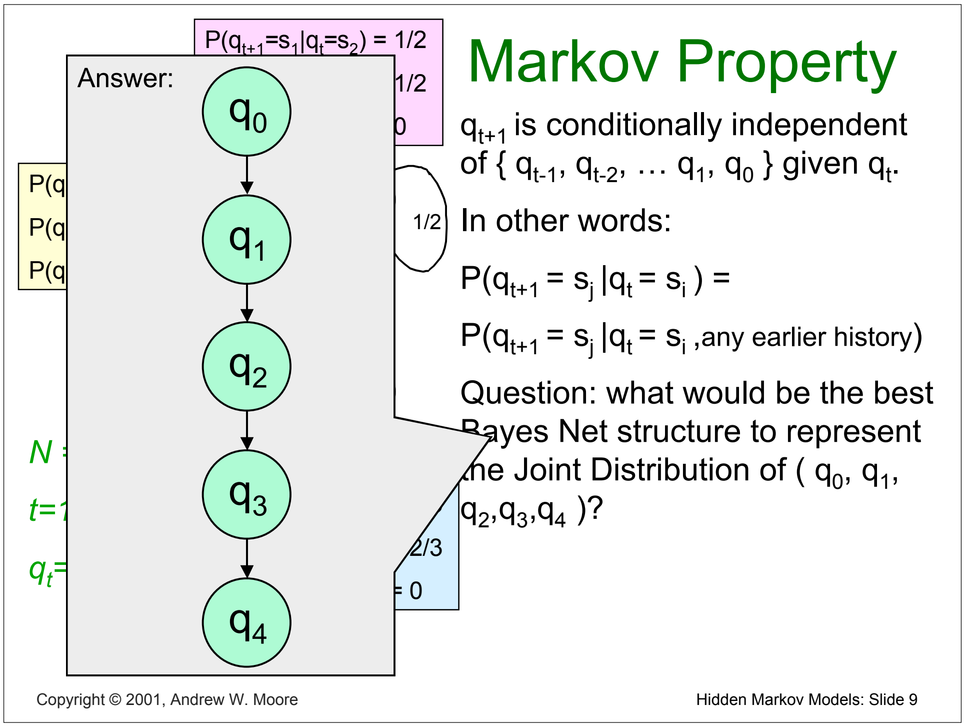
In other words:

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

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

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

Copyright © 2001, Andrew W. Moore



Markov Property

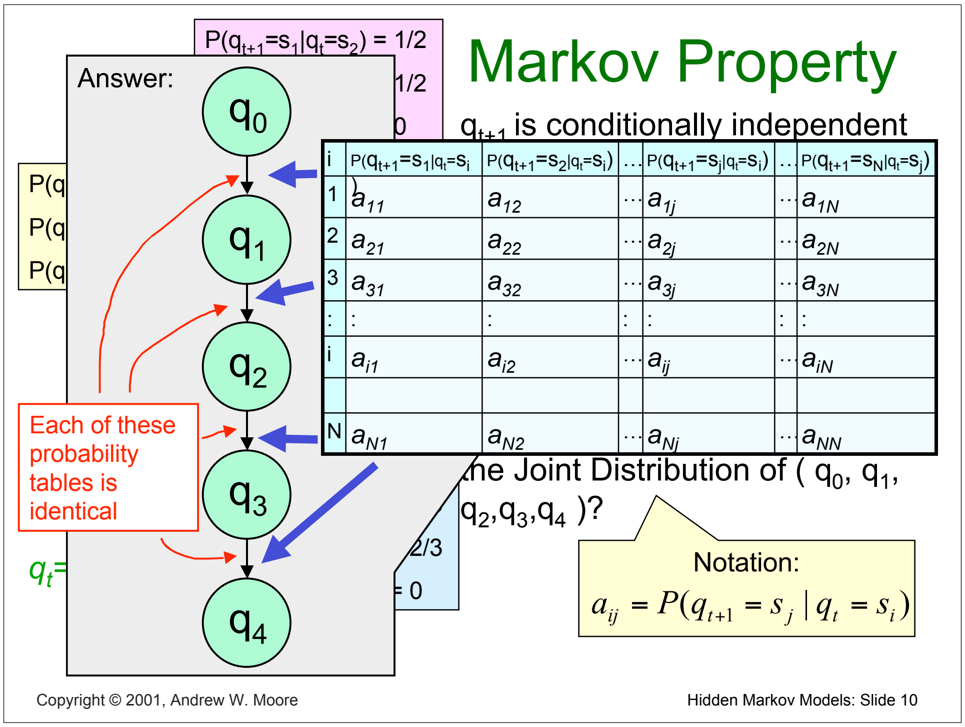
q_{t+1} is conditionally independent of { q_{t-1}, q_{t-2}, … q_1, q_0 } given q_t.

In other words:

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

P(q_{t+1} = s_j |q_t = s_i ,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

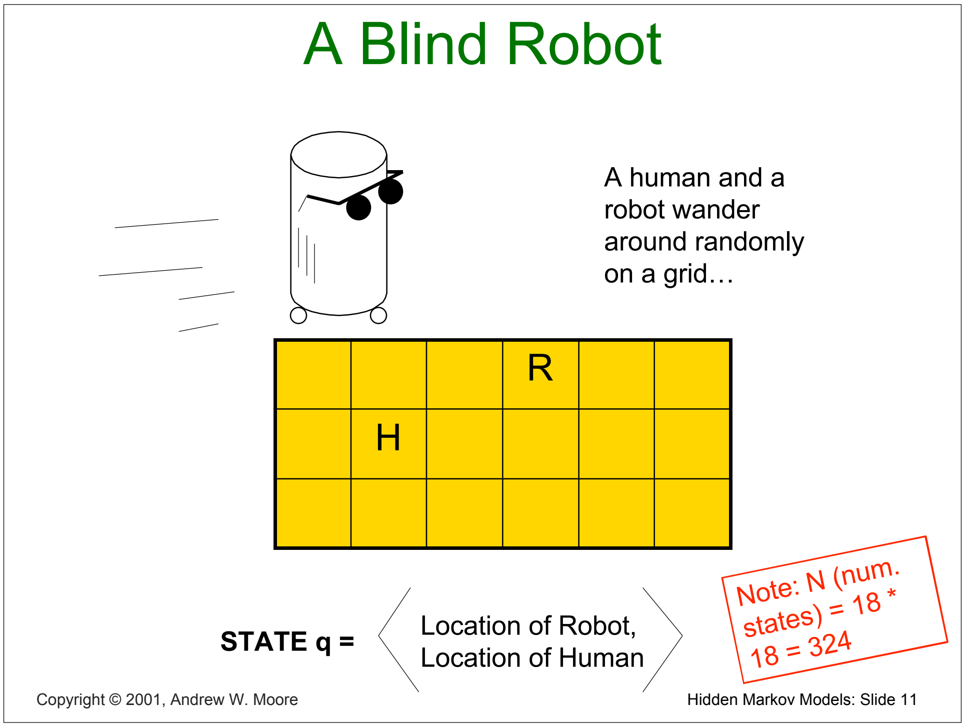
i	P(q_{t+1}=s_1|q_t=s_i)	P(q_{t+1}=s_2|q_t=s_i)	…	P(q_{t+1}=s_j|q_t=s_i)	…	P(q_{t+1}=s_N|q_t=s_j)
1	a_{11}	a_{12}	…	a_{1j}	…	a_{1N}
2	a_{21}	a_{22}	…	a_{2j}	…	a_{2N}
3	a_{31}	a_{32}	…	a_{3j}	…	a_{3N}
:	:	:	:	:	:	:
i	a_{i1}	a_{i2}	…	a_{ij}	…	a_{iN}
N	a_{N1}	a_{N2}	…	a_{Nj}	…	a_{NN}

the Joint Distribution of (q_0, q_1, q_2,q_3,q_4)?

Each of these probability tables is identical

Notation:

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



A Blind Robot

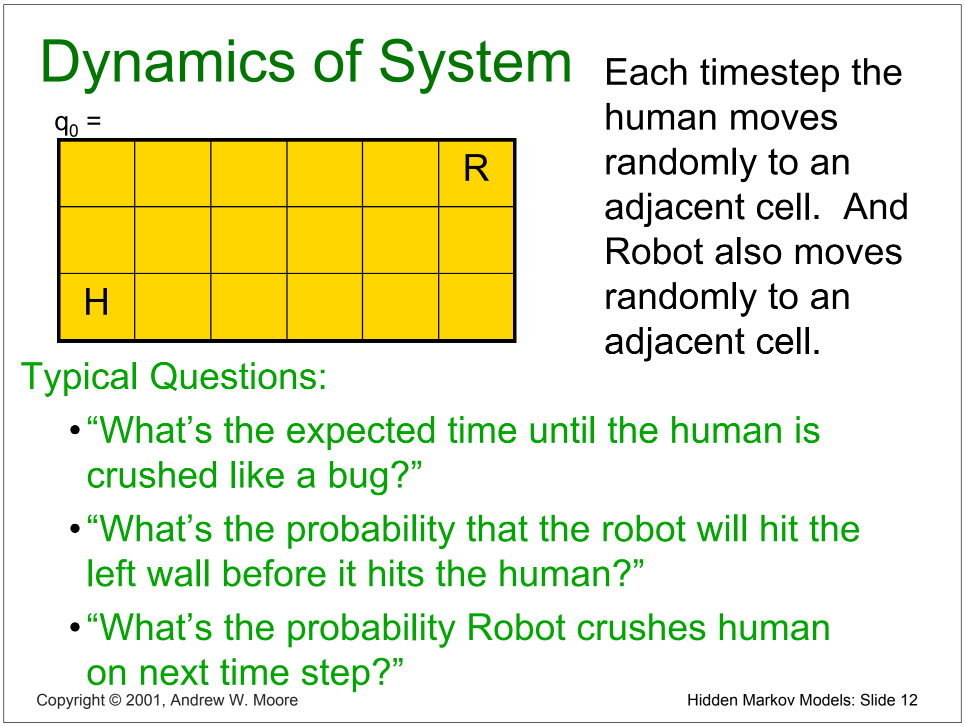
A human and a robot wander around randomly on a grid…

			R		
	H				

STATE q = ⟨ Location of Robot, Location of Human ⟩

Note: N (num. states) = 18 * 18 = 324

Copyright © 2001, Andrew W. Moore



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?"

Copyright © 2001, Andrew W. Moore

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.

We'll do this first

If robot is omnipotent:

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

Too Easy. We won't do this

If robot has some sensors, but incomplete state information …

Hidden Markov Models are applicable!

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 $Q = q_1\ q_2\ q_3\ ..\ q_t$

Given we know the start state q_1

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

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

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 $P(q_t = s)$

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

Computation is exponential in t

What is P(q_t =s) ? Clever answer

- For each state s_i, define

  $p_t(i)$ = Prob. state is s_i at time t

  = $P(q_t = s_i)$
- Easy to do inductive definition

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

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

Copyright © 2001, Andrew W. Moore

What is P(q_t =s) ? Clever answer

- For each state s_i, define

  $p_t(i)$ = Prob. state is s_i at time t

  = $P(q_t = s_i)$
- Easy to do inductive definition

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

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

Copyright © 2001, Andrew W. Moore

What is P(q$_t$ =s) ? Clever answer

- For each state s_i, define

  $p_t(i)$ = Prob. state is s_i 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

  $p_t(i)$ = Prob. state is s_i at time t

  $= P(q_t = s_i)$
- Easy to do inductive definition

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

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

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

Remember,

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

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

What is P(q_t =s) ? Clever answer

- For each state s_i, define

  $p_t(i)$ = Prob. state is s_i at time t

  $= P(q_t = s_i)$

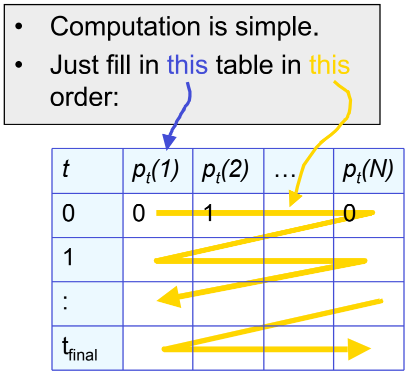
- Easy to do inductive definition

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

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

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

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



- 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				
:				
t_{final}				

Copyright © 2001, Andrew W. Moore

What is P(q_t =s) ? Clever answer

- For each state s_i, define

  $p_t(i)$ = Prob. state is s_i 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.

Copyright © 2001, Andrew W. Moore

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.

We'll do this first

If robot is omnipotent:

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

Too Easy. We won't do this

If robot has some sensors, but incomplete state information …

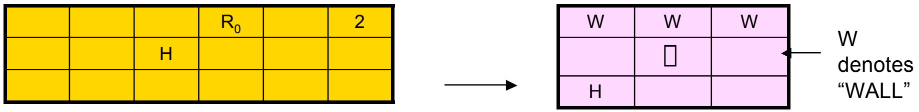
Hidden Markov Models are applicable!

Main Body of Lecture

Copyright © 2001, Andrew W. Moore

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)



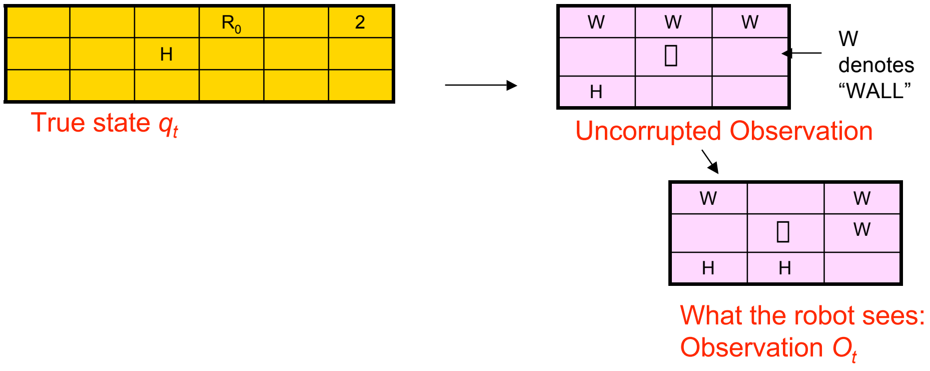
True state q_t

What the robot sees: Observation O_t

Copyright © 2001, Andrew W. Moore

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

W	W	W
	□	
H		

W denotes “WALL”

Uncorrupted Observation

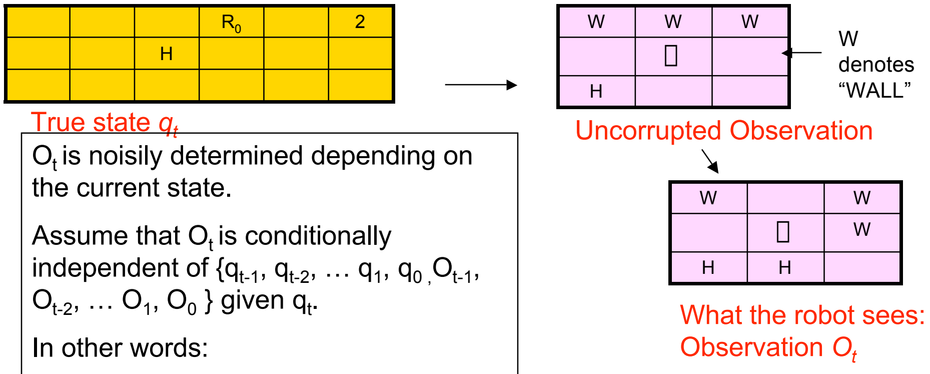
W		W
	□	W
H	H	

What the robot sees: Observation O_t

Copyright © 2001, Andrew W. Moore

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

W	W	W
	□	
H		

W denotes “WALL”

Uncorrupted Observation

W		W
	□	W
H	H	

What the robot sees: Observation O_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:

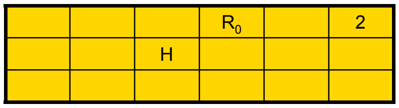
$P(O_t = X | q_t = s_i) =$

$P(O_t = X | q_t = s_i$,any earlier history)

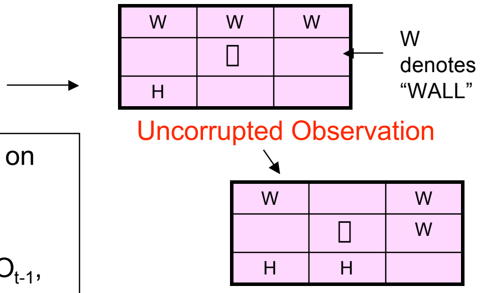
Copyright © 2001, Andrew W. Moore

Noisy Hidden State

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



True state q_t



W denotes "WALL"

Uncorrupted Observation

What the robot sees: Observation O_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:

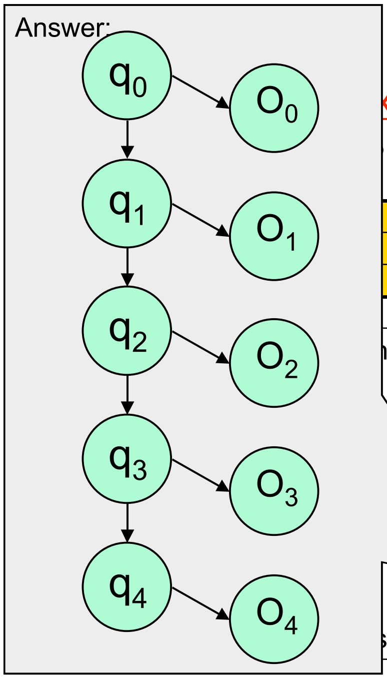
$P(O_t = X \mid q_t = s_i) =$

$P(O_t = X \mid q_t = s_i$,any earlier history)

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

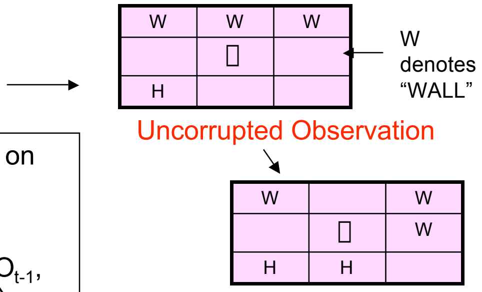
Copyright © 2001, Andrew W. Moore

Answer:



Hidden State

ximity sensors. (unreliably tell us adjacent squares)



W denotes "WALL"

Uncorrupted Observation

What the robot sees: Observation O_t

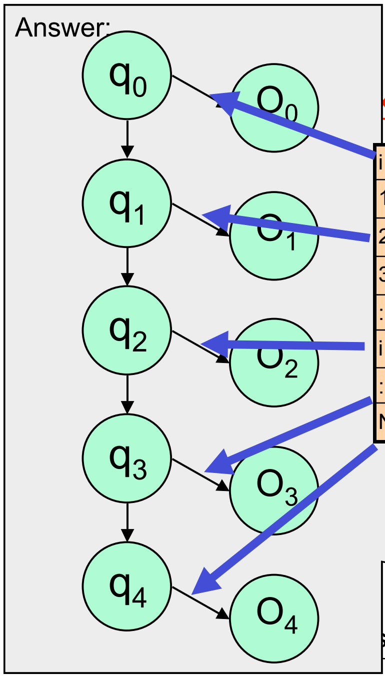
nding on

O_{t-1},

story)

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

Copyright © 2001, Andrew W. Moore



Hidden State

ximity sens s

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	$b_N(1)$	$b_N(2)$	…	$b_N(k)$	…	$b_N(M)$

O_{t-1},

H	H	

What the robot sees: Observation O_t

story)

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)?

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)
- Robot learning control (e.g. Yangsheng Xu's work)
- 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.

Copyright © 2001, Andrew W. Moore

HMM Notation (from Rabiner's Survey)

The states are labeled S_1 S_2 .. S_N

For a particular trial….

Let T be the number of observations

T is also the number of states passed through

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

$Q = q_1\ q_2\ ..\ q_T$ 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

Copyright © 2001, Andrew W. Moore

HMM Formal Definition

An HMM, _, is a 5-tuple consisting of

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

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

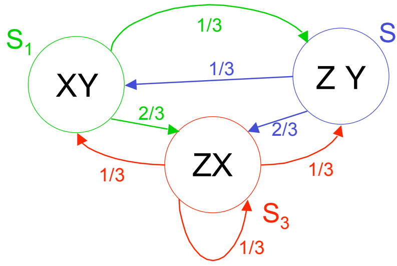
- $$\left.\begin{array}{cccc} a_{11} & a_{22} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ : & : & & : \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{array}\right\}$$ The state transition probabilities $P(q_{t+1}=S_j \mid q_t=S_i)=a_{ij}$
- $$\left.\begin{array}{cccc} b_1(1) & b_1(2) & \dots & b_1(M) \\ b_2(1) & b_2(2) & \dots & b_2(M) \\ : & : & & : \\ b_N(1) & b_N(2) & \dots & b_N(M) \end{array}\right\}$$ The observation probabilities $P(O_t=k \mid q_t=S_i)=b_i(k)$

Copyright © 2001, 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.

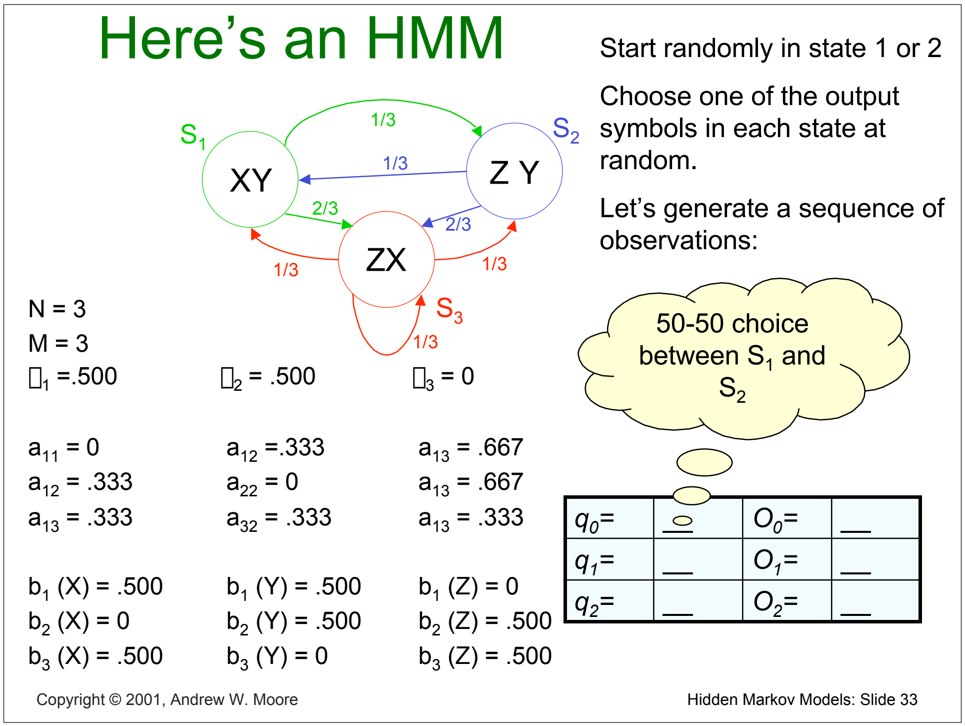


N = 3

M = 3

π_1 =.500	π_2 = .500	π_3 = 0
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333
b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

Copyright © 2001, 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:

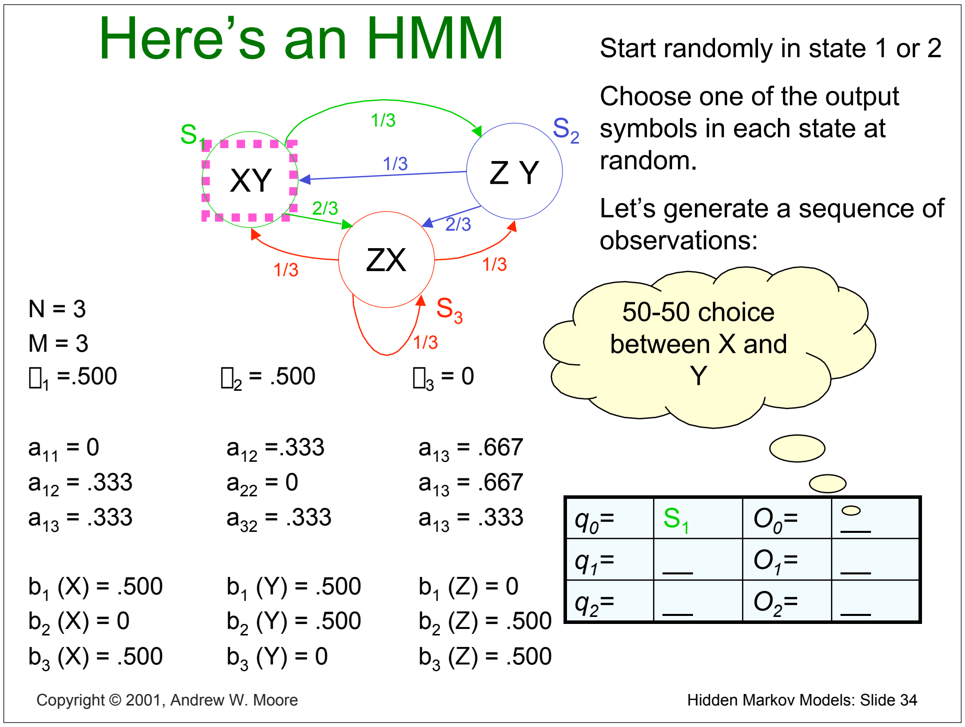
50-50 choice between S$_1$ and S$_2$

N = 3
M = 3

□$_1$ =.500	□$_2$ = .500	□$_3$ = 0
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333
b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

q_0=	___	O_0=	___
q_1=	___	O_1=	___
q_2=	___	O_2=	___

Copyright © 2001, 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:

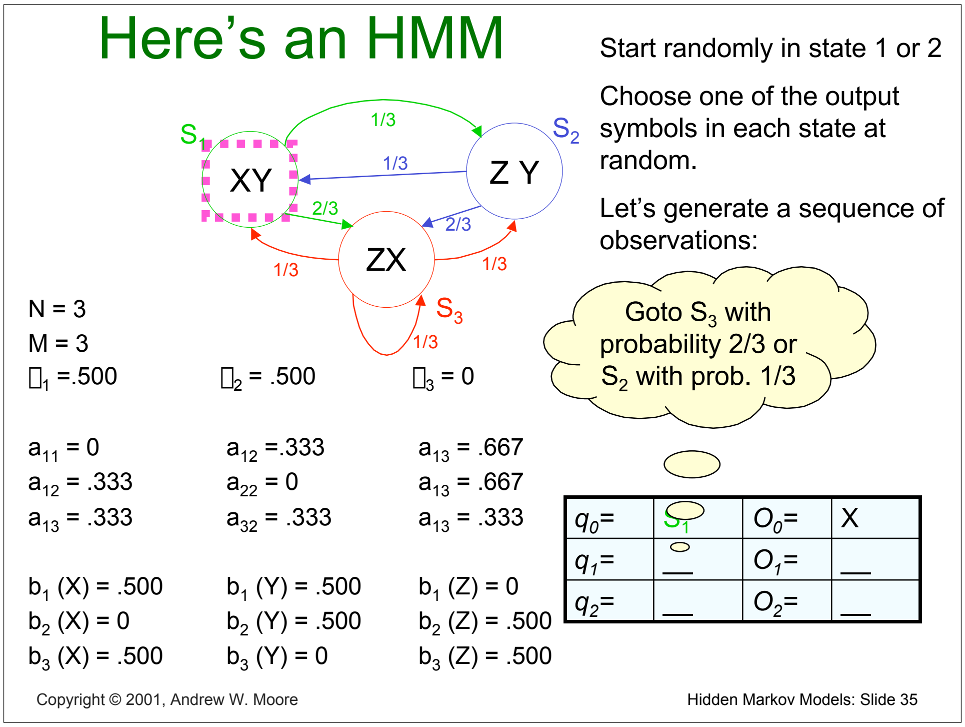
50-50 choice between X and Y

N = 3
M = 3

□$_1$ =.500	□$_2$ = .500	□$_3$ = 0
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333
b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

q_0=	S$_1$	O_0=	___
q_1=	___	O_1=	___
q_2=	___	O_2=	___

Copyright © 2001, 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:

Goto S_3 with probability 2/3 or S_2 with prob. 1/3

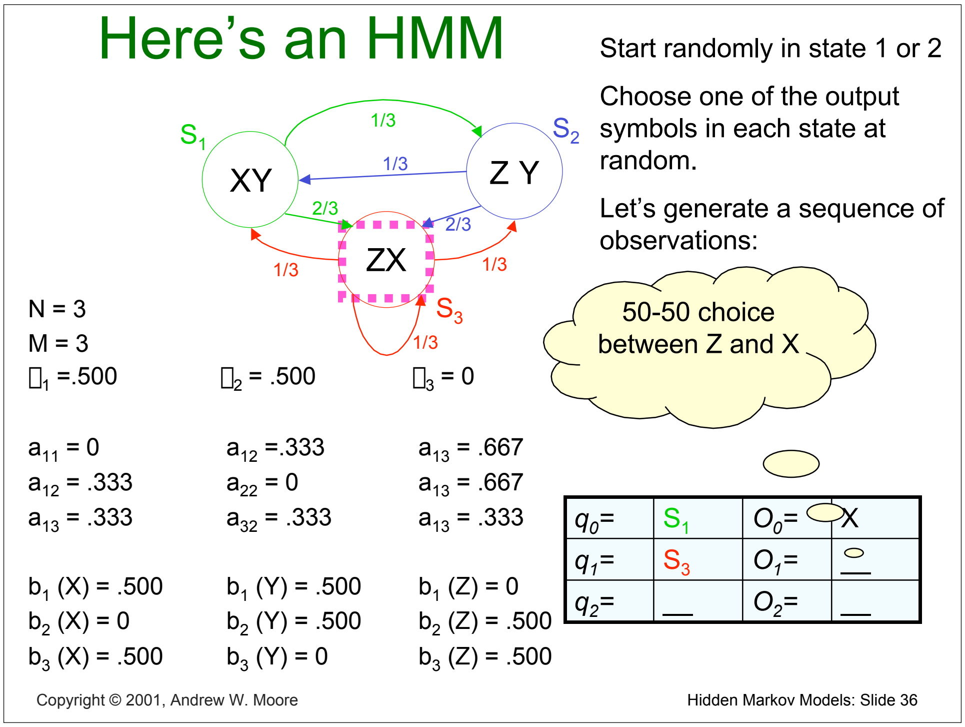
N = 3
M = 3
$\square_1$ =.500 $\square_2$ = .500 $\square_3$ = 0

a_{11} = 0 a_{12} =.333 a_{13} = .667
a_{12} = .333 a_{22} = 0 a_{13} = .667
a_{13} = .333 a_{32} = .333 a_{13} = .333

b_1 (X) = .500 b_1 (Y) = .500 b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = .500 b_2 (Z) = .500
b_3 (X) = .500 b_3 (Y) = 0 b_3 (Z) = .500

q_0=	S_1	O_0=	X
q_1=	__	O_1=	__
q_2=	__	O_2=	__

Copyright © 2001, 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:

50-50 choice between Z and X

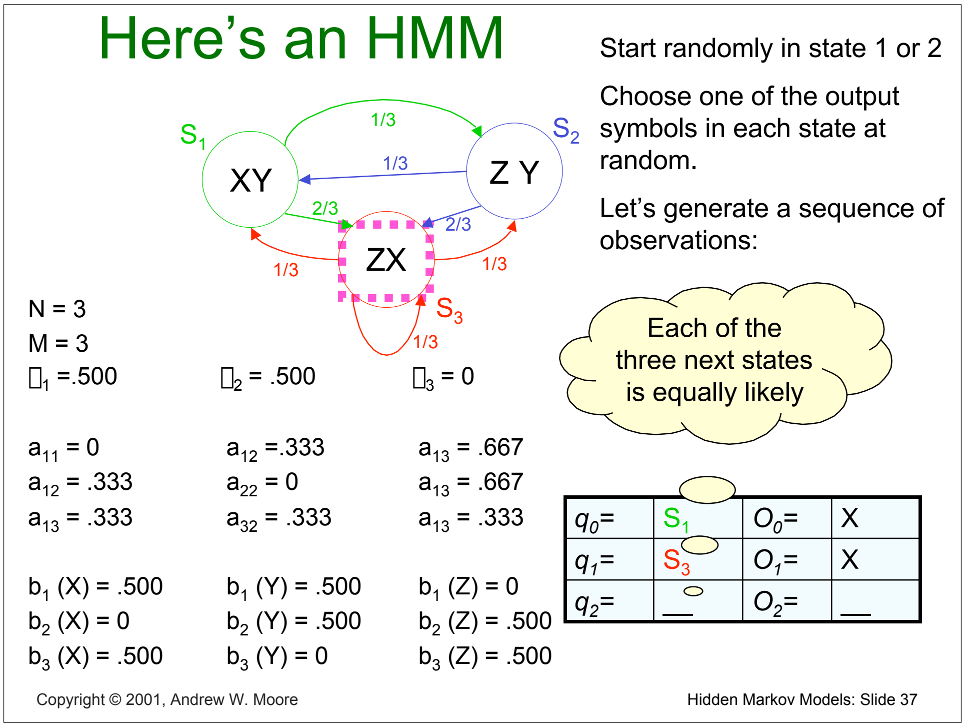
N = 3
M = 3
$\square_1$ =.500 $\square_2$ = .500 $\square_3$ = 0

a_{11} = 0 a_{12} =.333 a_{13} = .667
a_{12} = .333 a_{22} = 0 a_{13} = .667
a_{13} = .333 a_{32} = .333 a_{13} = .333

b_1 (X) = .500 b_1 (Y) = .500 b_1 (Z) = 0
b_2 (X) = 0 b_2 (Y) = .500 b_2 (Z) = .500
b_3 (X) = .500 b_3 (Y) = 0 b_3 (Z) = .500

q_0=	S_1	O_0=	X
q_1=	S_3	O_1=	__
q_2=	__	O_2=	__

Copyright © 2001, 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:

Each of the three next states is equally likely

N = 3
M = 3

$\square_1$ =.500 $\square_2$ = .500 $\square_3$ = 0

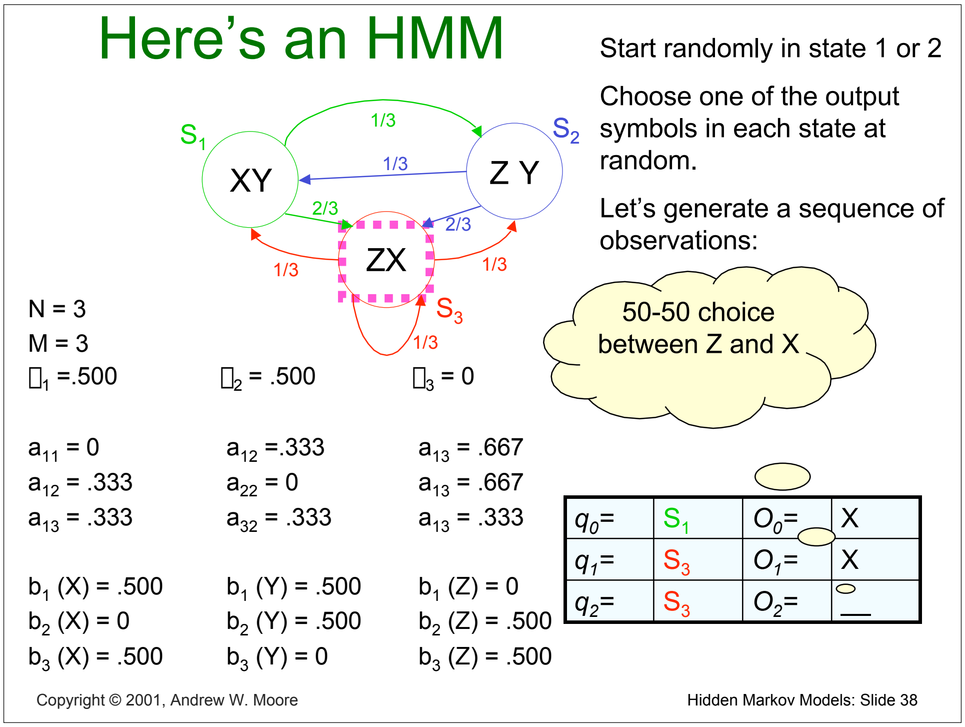
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333

b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

q_0=	S$_1$	O_0=	X
q_1=	S$_3$	O_1=	X
q_2=	__	O_2=	__

Copyright © 2001, Andrew W. Moore

Hidden Markov Models: Slide 37



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

$\square_1$ =.500 $\square_2$ = .500 $\square_3$ = 0

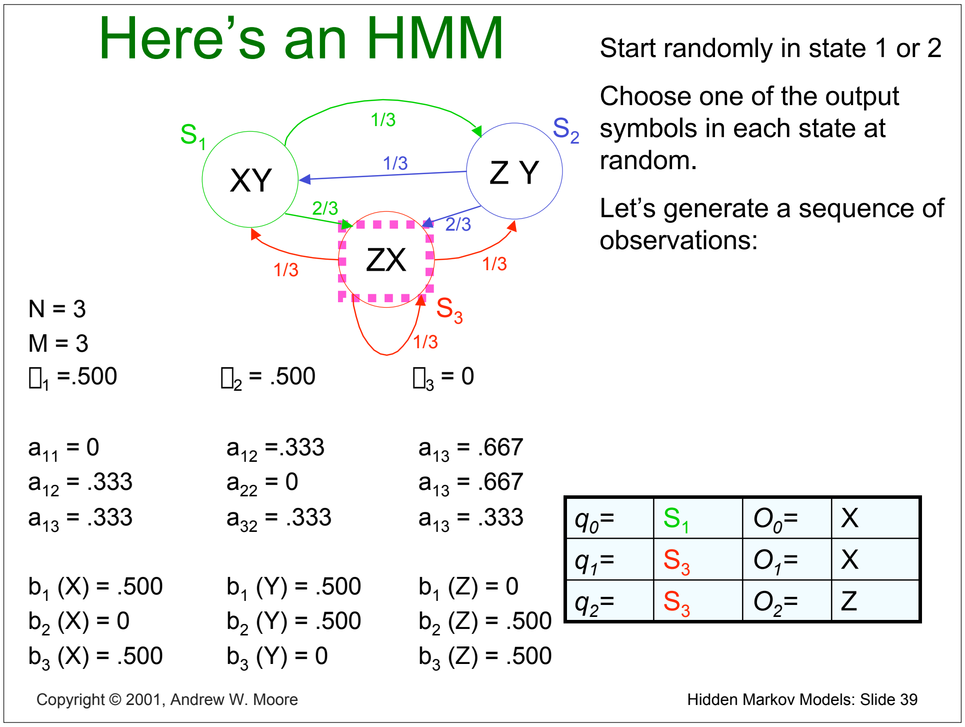
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333

b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

q_0=	S$_1$	O_0=	X
q_1=	S$_3$	O_1=	X
q_2=	S$_3$	O_2=	__

Copyright © 2001, Andrew W. Moore

Hidden Markov Models: Slide 38



Here’s an HMM

Start randomly in state 1 or 2

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

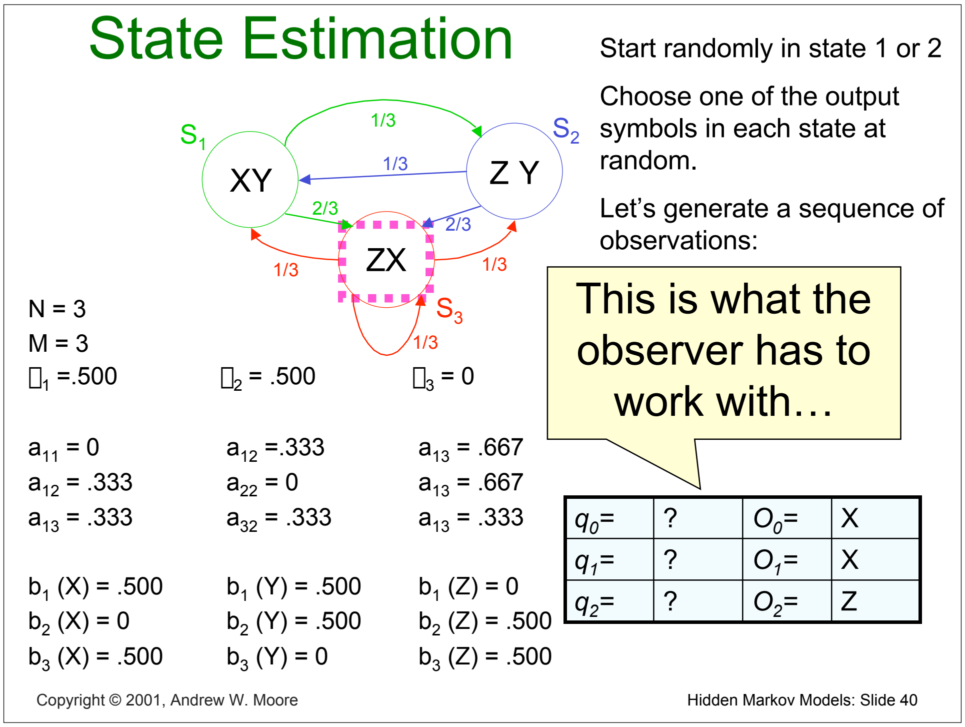
Let’s generate a sequence of observations:

N = 3
M = 3

$\square_1$ =.500	$\square_2$ = .500	$\square_3$ = 0
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333
b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

q_0=	S$_1$	O_0=	X
q_1=	S$_3$	O_1=	X
q_2=	S$_3$	O_2=	Z

Copyright © 2001, Andrew W. Moore



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

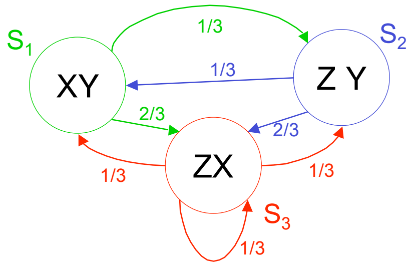
$\square_1$ =.500	$\square_2$ = .500	$\square_3$ = 0
a_{11} = 0	a_{12} =.333	a_{13} = .667
a_{12} = .333	a_{22} = 0	a_{13} = .667
a_{13} = .333	a_{32} = .333	a_{13} = .333
b_1 (X) = .500	b_1 (Y) = .500	b_1 (Z) = 0
b_2 (X) = 0	b_2 (Y) = .500	b_2 (Z) = .500
b_3 (X) = .500	b_3 (Y) = 0	b_3 (Z) = .500

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

Copyright © 2001, Andrew W. Moore

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:

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

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

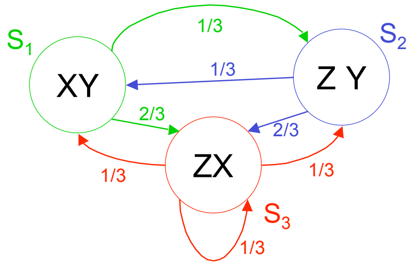
How do we compute P(Q) for an arbitrary path Q?

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

Copyright © 2001, Andrew W. Moore

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:

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

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

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

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

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

$=P(q_1) P(q_2,q_3|q_1)$ (chain rule)

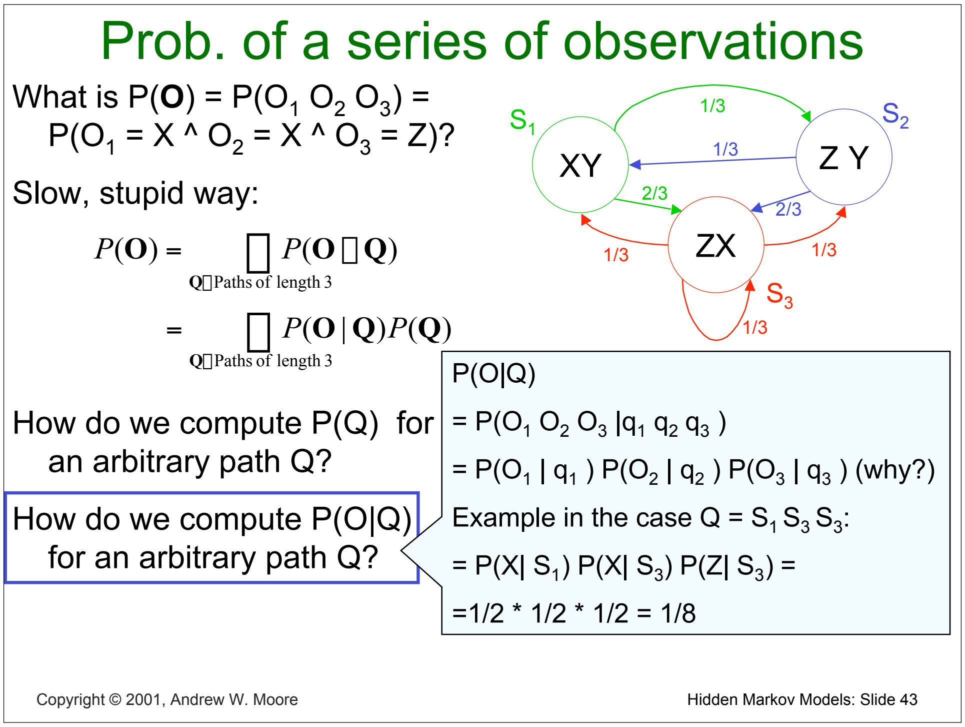
$=P(q_1) P(q_2|q_1) P(q_3| q_2,q_1)$ (chain)

$=P(q_1) P(q_2|q_1) P(q_3| q_2)$ (why?)

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

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

Copyright © 2001, Andrew W. Moore



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:

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

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

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

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

$P(O|Q)$

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

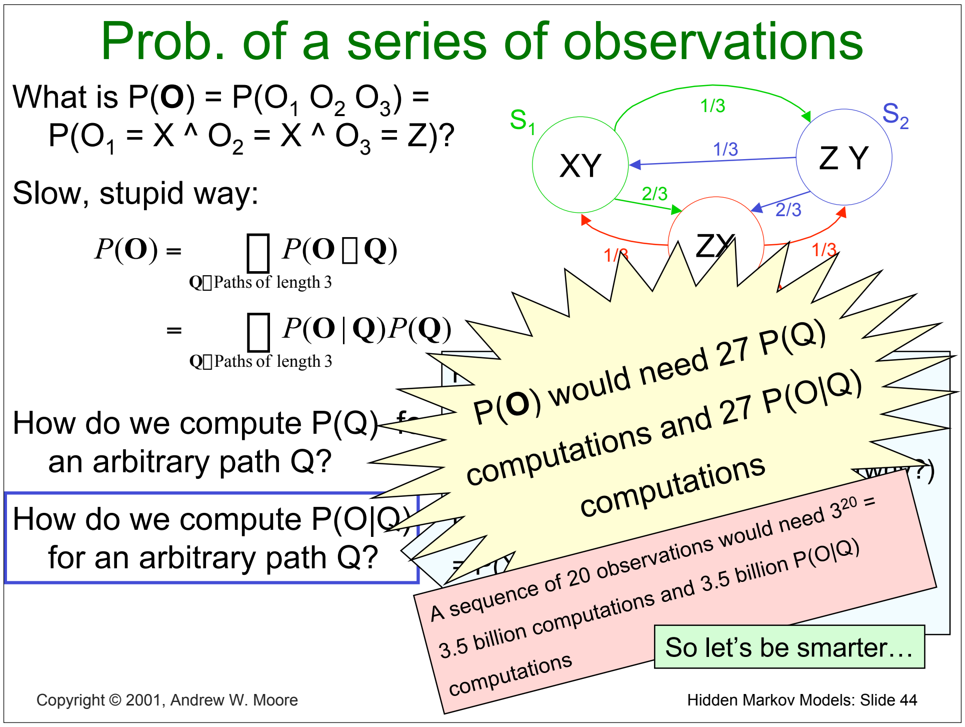
$= P(O_1 \mid q_1\ )\ P(O_2 \mid q_2\ )\ P(O_3 \mid q_3\ )$ (why?)

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

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

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

Copyright © 2001, Andrew W. Moore



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:

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

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

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

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

P(**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…

Copyright © 2001, Andrew W. Moore

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

Given observations $O_1\ O_2\ \ldots\ O_T$

Define

$\alpha_t(i) = P(O_1\ O_2\ \ldots\ O_t \wedge q_t = S_i \mid \lambda)$ where $1 \le t \le T$

$\alpha_t(i)$ = 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 $\alpha_2(3)$?

Copyright © 2001, Andrew W. Moore

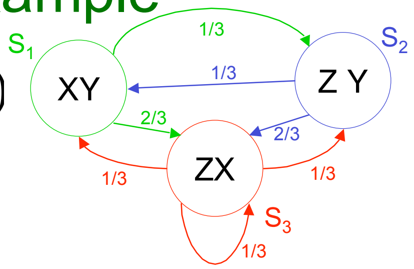
$\alpha_t(i)$: easy to define recursively

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

$$
\begin{aligned}
\alpha_1(i) &= P(O_1 \wedge q_1 = S_i)\\
&= P(q_1 = S_i)P(O_1 | q_1 = S_i)\\
&= \qquad\qquad \text{what?}\\
\alpha_{t+1}(j) &= P(O_1O_2\ldots O_tO_{t+1} \wedge q_{t+1} = S_j)\\
&= \sum_{i=1}^{N} P(O_1O_2\ldots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j)\\
&= \sum_{i=1}^{N} P(O_{t+1}, q_{t+1} = S_j | O_1O_2\ldots O_t \wedge q_t = S_i)P(O_1O_2\ldots O_t \wedge q_t = S_i)\\
&= \sum_{i} P(O_{t+1}, q_{t+1} = S_j | q_t = S_i)\alpha_t(i)\\
&= \sum_{i} P(q_{t+1} = S_j | q_t = S_i)P(O_{t+1} | q_{t+1} = S_j)\alpha_t(i)\\
&= \sum_{i} a_{ij} b_j(O_{t+1})\alpha_t(i)
\end{aligned}
$$

Copyright © 2001, Andrew W. Moore

in our example



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

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

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

WE SAW O_1 O_2 O_3 = X X Z

$$\begin{array}{lll}
\alpha_1(1) = \frac{1}{4} & \alpha_1(2) = 0 & \alpha_1(3) = 0 \\
\alpha_2(1) = 0 & \alpha_2(2) = 0 & \alpha_2(3) = \frac{1}{12} \\
\alpha_3(1) = 0 & \alpha_3(2) = \frac{1}{72} & \alpha_3(3) = \frac{1}{72}
\end{array}$$

Copyright © 2001, Andrew W. Moore

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

$P(O_1O_2\ldots O_t)$?

(How) can we cheaply compute

$P(q_t=S_i|O_1O_2\ldots O_t)$

Copyright © 2001, Andrew W. Moore

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

$P(O_1O_2\ldots O_t)$?

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

(How) can we cheaply compute

$P(q_t=S_i|O_1O_2\ldots O_t)$

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

Copyright © 2001, Andrew W. Moore

Most probable path given observations

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

What is $\underset{Q}{\operatorname{argmax}}\ \mathrm{P}\left(Q|O_1O_2\ldots O_T\right)$?

Slow, stupid answer:

$$\underset{Q}{\operatorname{argmax}}\ \mathrm{P}\left(Q|O_1O_2\ldots O_T\right)$$

$$=\underset{Q}{\operatorname{argmax}}\ \frac{\mathrm{P}\left(O_1O_2\ldots O_T|Q\right)\mathrm{P}(Q)}{\mathrm{P}\left(O_1O_2\ldots O_T\right)}$$

$$=\underset{Q}{\operatorname{argmax}}\ \mathrm{P}\left(O_1O_2\ldots O_T|Q\right)\mathrm{P}(Q)$$

Copyright © 2001, Andrew W. Moore

Efficient MPP computation

We're going to compute the following variables:

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

= 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: $\text{mpp}_t(i)$ = that path

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

Copyright © 2001, Andrew W. Moore

The Viterbi Algorithm

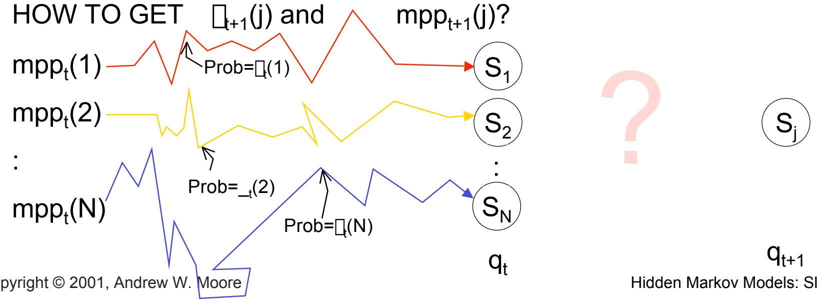
$$\delta_t(i) = \max_{q_1 q_2 \ldots q_{t-1}} \mathrm{P}(q_1 q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 .. O_t)$$

$$mpp_t(i) = \operatorname*{arg\,max}_{q_1 q_2 \ldots q_{t-1}} \mathrm{P}(q_1 q_2 \ldots q_{t-1} \wedge q_t = S_i \wedge O_1 O_2 .. O_t)$$

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

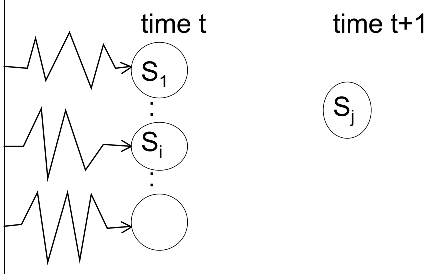
Now, suppose we have all the $\delta_t(i)$'s and $\text{mpp}_t(i)$'s for all i.

HOW TO GET $\delta_{t+1}(j)$ and $\text{mpp}_{t+1}(j)$?



Copyright © 2001, Andrew W. Moore

The Viterbi Algorithm



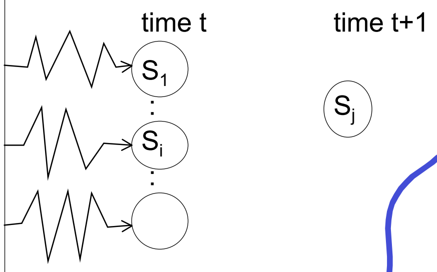
The most prob path with last two states S_i S_j

is

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

Copyright © 2001, Andrew W. Moore

The Viterbi Algorithm



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?

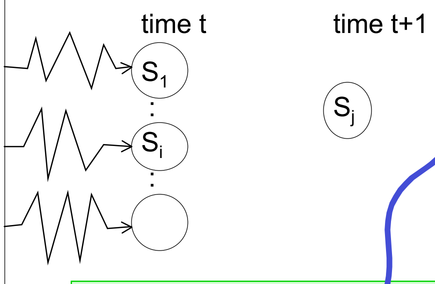
$$\delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1} \mid \lambda)$$

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

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

Copyright © 2001, Andrew W. Moore

The Viterbi Algorithm



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 □ S_j

What is the prob of that path?

$\square_t(i)$ x $P(S_i$ □ S_j □ $O_{t+1} | \square)$

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

SO The most probable

S_{i*} as its penultimate state

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

i

Summary:

$\square_{t+1}(j) = \square_t(i^*)\ a_{ij}\ b_j\ (O_{t+1})$
$\text{mpp}_{t+1}(j) = \text{mpp}_{t+1}(i^*)S_{i^*}$ } 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

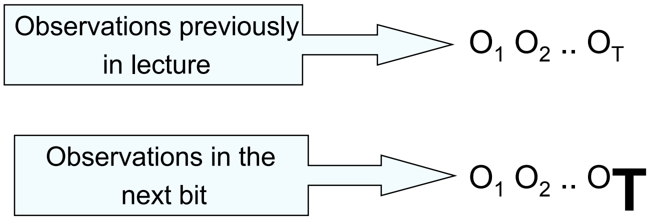
In practice: many levels of inference; not one big jump.

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. $O_1\ O_2\ ..\ O_T$ with a big “T”.


Observations previously in lecture
$O_1\ O_2\ ..\ O_T$
Observations in the next bit
$O_1\ O_2\ ..\ O_{\mathbf{T}}$

Inferring an HMM

Remember, we’ve been doing things like

$$P(O_1\ O_2\ ..\ O_T \mid □\ )$$

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

<u>Now</u> We have some observations and we want to estimate □ from them.

AS USUAL: We could use

(i) MAX LIKELIHOOD $□ = \underset{□}{\text{argmax}}\ P(O_1\ ..\ O_T \mid □)$

(ii) BAYES

Work out $P(□ \mid O_1\ ..\ O_T\ )$

and then take $E[□]$ or $\underset{□}{\max}\ P(□ \mid O_1\ ..\ O_T\ )$

Max likelihood HMM estimation

Define

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

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

$\gamma_t(i)$ and $\xi_t(i,j)$ can be computed efficiently $\forall i,j,t$

(Details in Rabiner paper)

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

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

Copyright © 2001, Andrew W. Moore

$\gamma_t(i) = \mathrm{P}(q_t = S_i | O_1 O_2 .. O_T , \lambda)$

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

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

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

HMM estimation

$$\text{Notice } \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\begin{pmatrix}\text{expected frequency}\\ \text{i} \rightarrow \text{j}\end{pmatrix}}{\begin{pmatrix}\text{expected frequency}\\ \text{i}\end{pmatrix}}$$

$= \text{Estimate of Prob}(\text{Next state } \mathrm{S_j} | \text{This state } \mathrm{S_i})$

We can re-estimate

$$\mathrm{a_{ij}} \leftarrow \frac{\sum \xi_t(i,j)}{\sum \gamma_t(i)}$$

We can also re-estimate

$\mathrm{b_j}(O_k) \leftarrow \cdots$ (See Rabiner)

Copyright © 2001, Andrew W. Moore

EM for HMMs

If we knew _ we could estimate EXPECTATIONS of quantities such as

Expected number of times in state i

Expected number of transitions i → j

If we knew the quantities such as

Expected number of times in state i

Expected number of transitions i → j

We could compute the MAX LIKELIHOOD estimate of

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

Roll on the EM Algorithm…

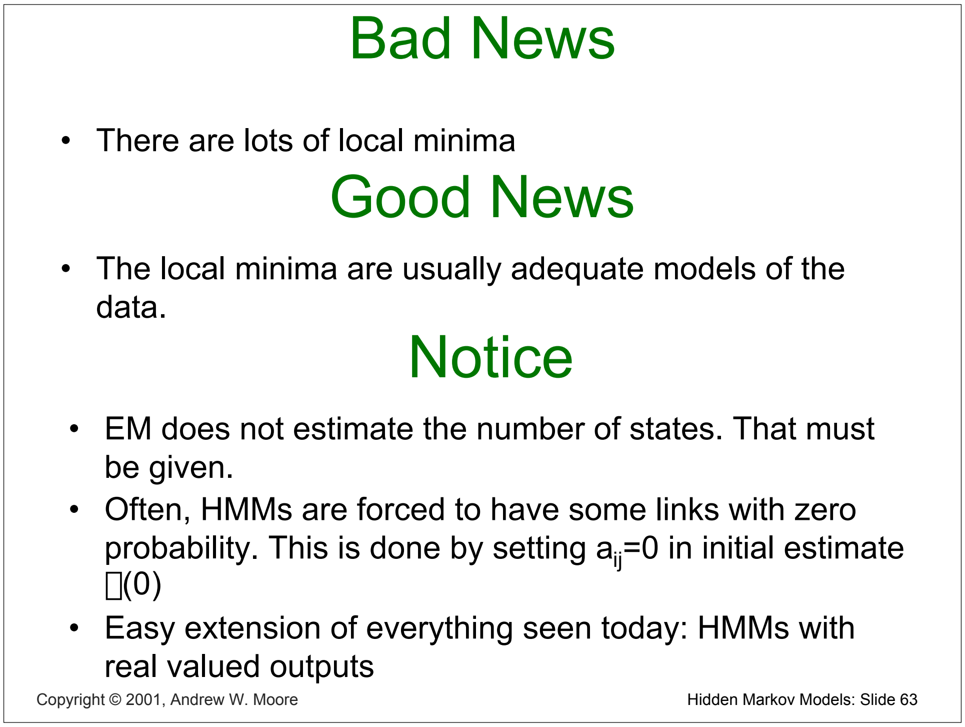
Copyright © 2001, Andrew W. Moore

EM 4 HMMs

1. Get your observations $O_1 \ldots O_T$
2. Guess your first λ estimate $\lambda(0)$, t=0
3. t = t+1
4. Given $O_1 \ldots O_T$, $\lambda(t)$ compute
   $\gamma_t(i)$, $\xi_t(i,j)$ $\forall 1 \leq t \leq T$, $\forall 1 \leq i \leq N$, $\forall 1 \leq j \leq N$
5. Compute expected freq. of state i, and expected freq. i→ j
6. Compute new estimates of a_{ij}, $b_j(k)$, π_i accordingly. Call them $\lambda(t+1)$
7. Goto 3, unless converged.

- **Also known (for the HMM case) as the BAUM-WELCH algorithm.**

Copyright © 2001, Andrew W. Moore



Bad News

- There are lots of local minima

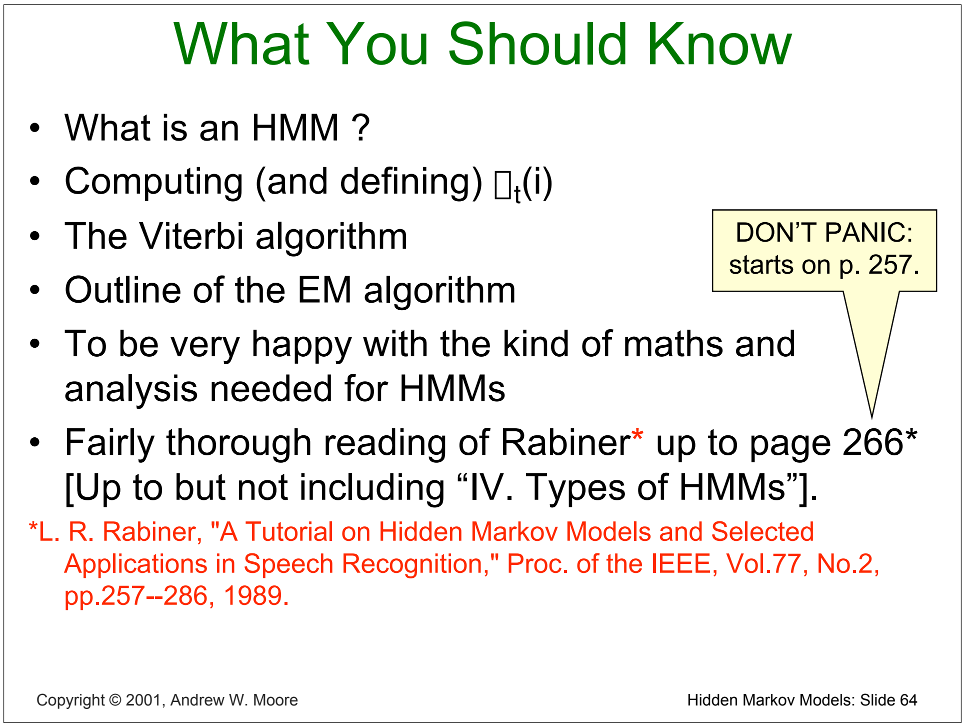
Good News

- 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 □(0)
- Easy extension of everything seen today: HMMs with real valued outputs

Copyright © 2001, Andrew W. Moore



What You Should Know

- What is an HMM ?
- Computing (and defining) $□_t$(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.

Copyright © 2001, Andrew W. Moore