Lecture 13: Hidden Markov Model

Shuai Li

John Hopcroft Center, Shanghai Jiao Tong University

https://shuaili8.github.io

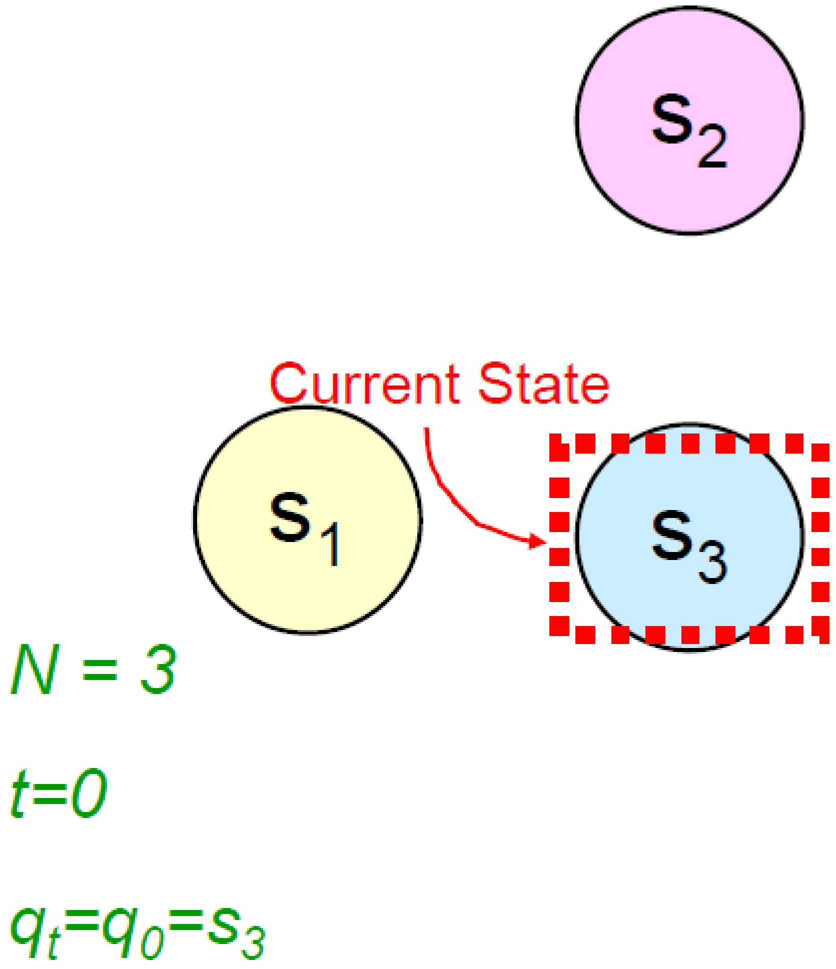
https://shuaili8.github.io/Teaching/VE445/index.html

A Markov system

- There are N states $S_1, S_2, \dots, S_N$, and the time steps are discrete, $t = 0,1,2, \dots$
- On the t-th time step the system is in exactly one of the available states. Call it q_t
- Between each time step, the next state is chosen only based on the information provided by the current state q_t
- The current state determines the probability distribution for the next state

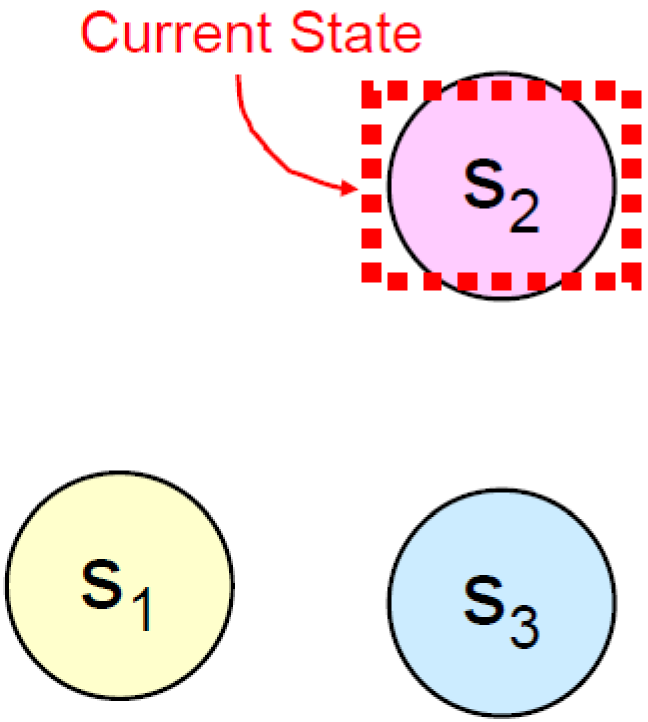
Example

- Three states
- Current state: S_3



Example (cont.)

- Three states
- Current state: S_2



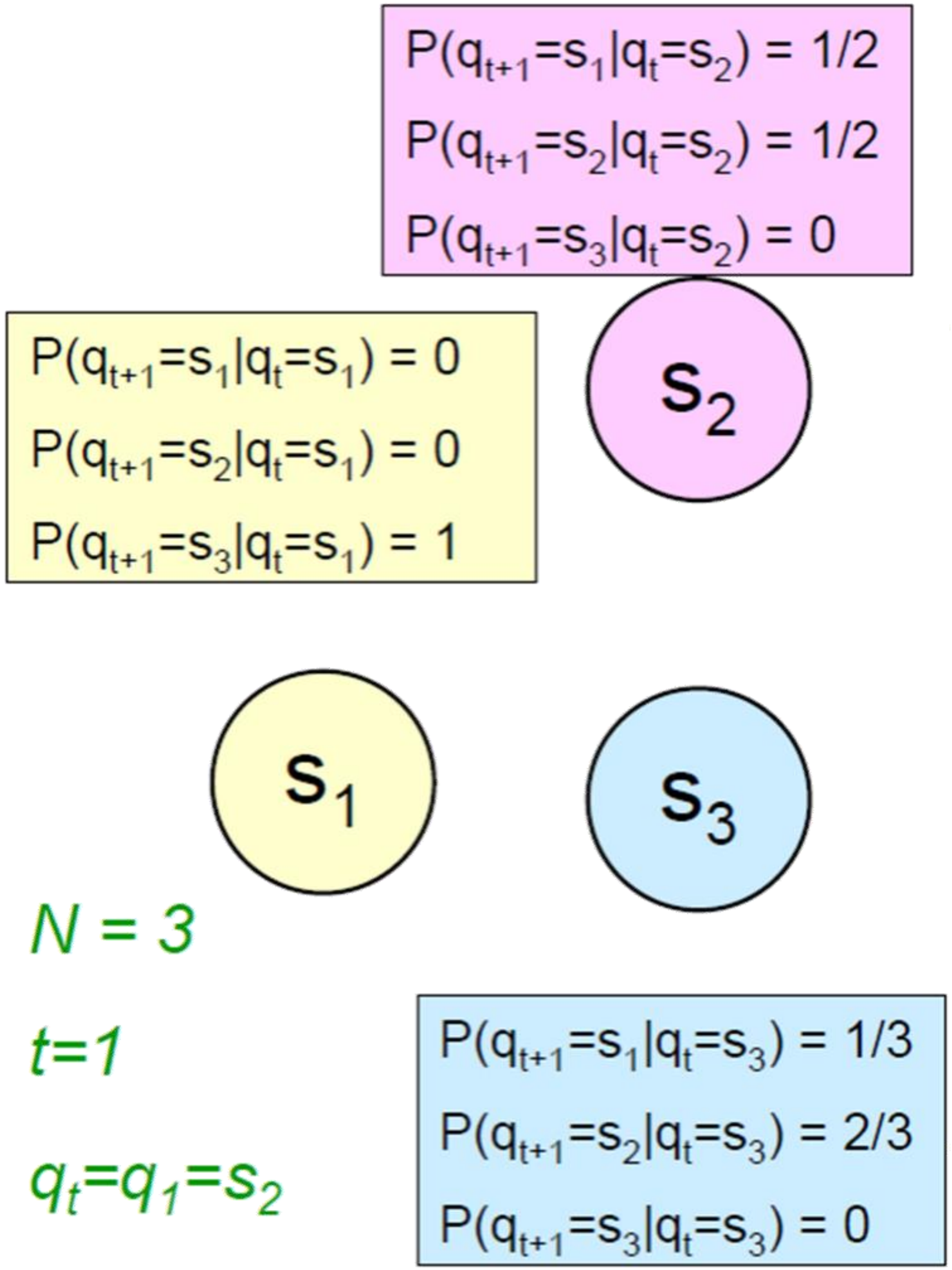
$N = 3$

$t=1$

$q_t=q_1=s_2$

Example (cont.)

- Three states
- The transition matrix



Example (cont.)

Markovian property

- q_{t+1} is independent of $\{q_{t-1}, q_{t-2}, \dots, q_0\}$ given q_t
- In other words:

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

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

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

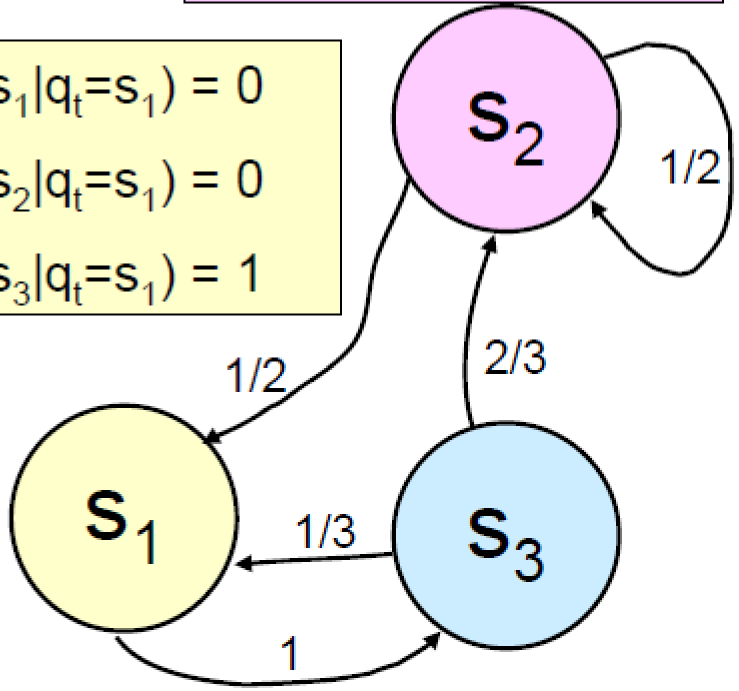
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$

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

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

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

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



$N = 3$

$t=1$

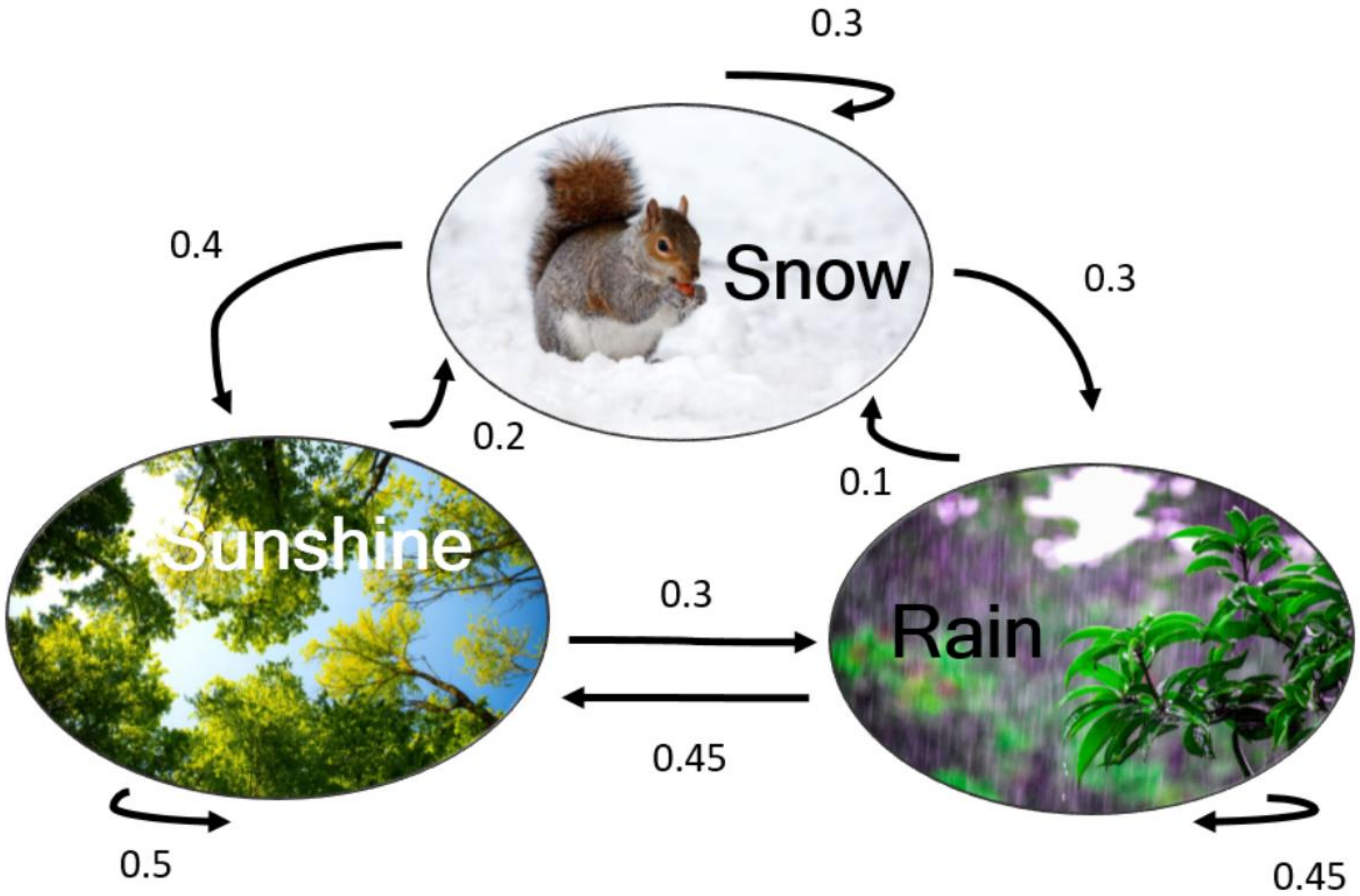
$q_t=q_1=s_2$

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

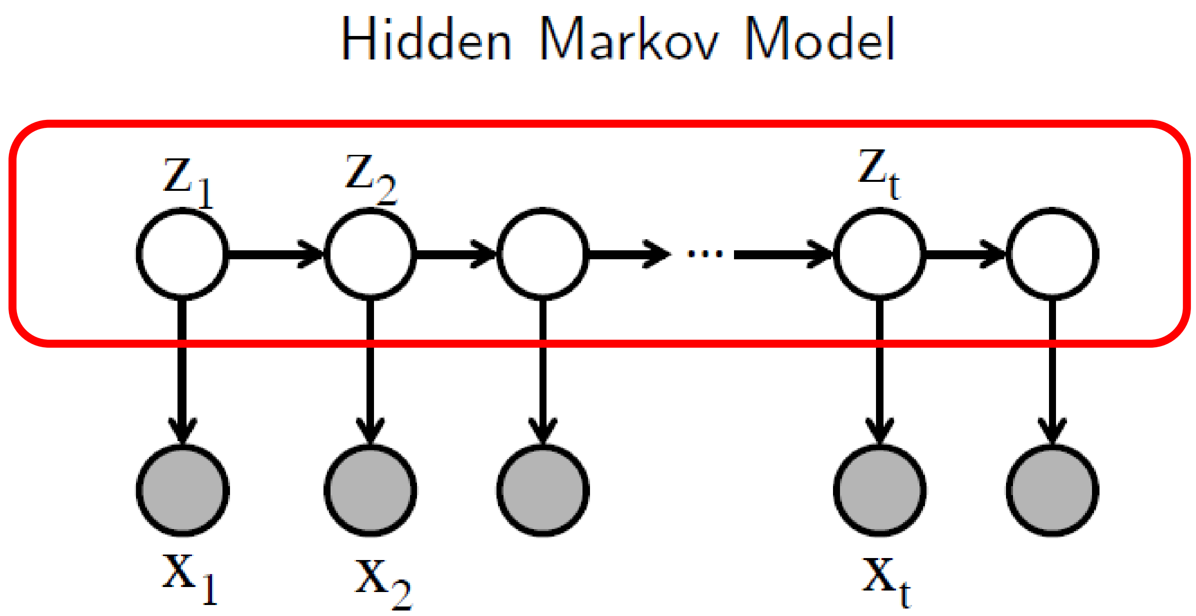
$P(q_{t+1}=s_2|q_t=s_3) = 2/3$

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

Example 2

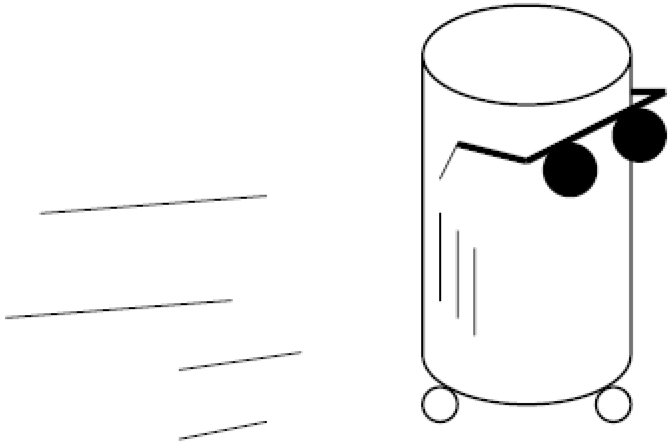


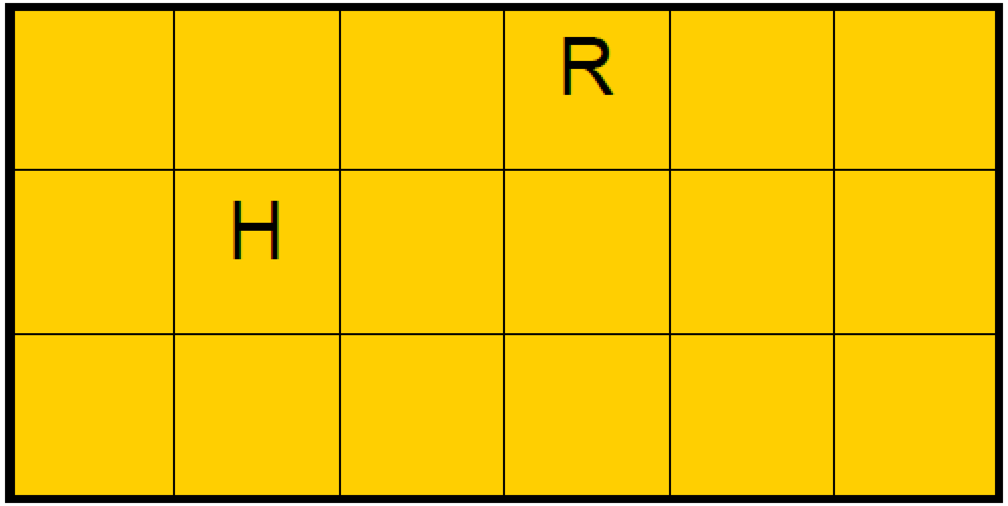
Markovian property



Example

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



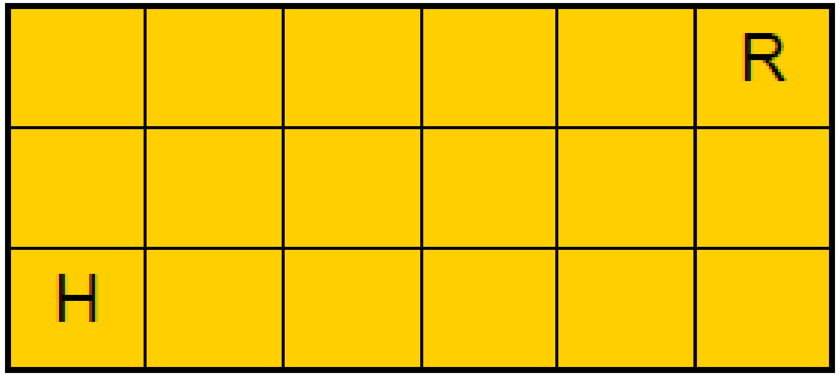


STATE q = $\left\langle \begin{array}{l}\text{Location of Robot,}\\ \text{Location of Human}\end{array} \right\rangle$

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

Example (cont.)

- Each time step the human/robot 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 (cont.)

- The currently time is 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 current state):
  - 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

$P(q_t = s)$ -- A clumsy solution

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

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

$$
\begin{aligned}
P(q_1\, q_2\, ..\, q_t) &= P(q_1\, q_2\, ..\, q_{t-1})\, P(q_t | q_1\, q_2\, ..\, q_{t-1}) \\
&= P(q_1\, q_2\, ..\, q_{t-1})\, P(q_t | q_{t-1}) \qquad \textit{WHY?} \\
&= P(q_2|q_1) P(q_3|q_2) \ldots P(q_t|q_{t-1})
\end{aligned}
$$

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

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

$P(q_t = s)$ -- A cleverer solution

- For each state S_i, define $p_t(i) = P(q_t = S_i)$ to be the probability of state S_i at time t
- Easy to do inductive computation

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

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

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

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

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

$P(q_t = s)$ -- A cleverer solution

- For each state S_i, define $p_t(i) = P(q_t = S_i)$ to be the probability of state S_i at time t
- Easy to do inductive computation

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

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

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

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

- 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}				

Complexity comparison

- Cost of computing $p_t(i)$ for all states S_i is now $O(tN^2)$
  - Why?
- The first method has $O(N^t)$
  - Why?
- This is the power of dynamic programming that is widely used in HMM

Example (cont.)

- 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

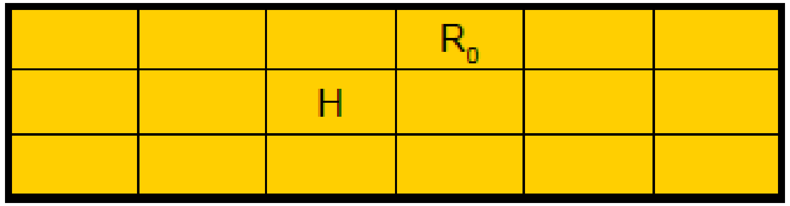
Too Easy. We won't do this

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

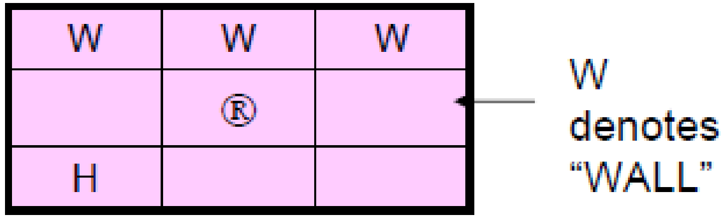
Main Body of Lecture

Hidden state

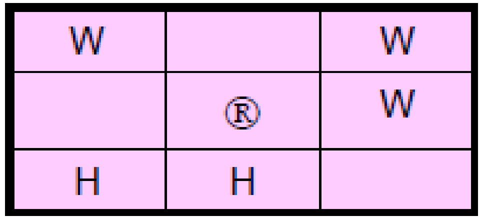
- The previous example tries to estimate $P(q_t = S_i)$ unconditionally (no other information)
- Suppose we can observe something that's affected by the true state



True state q_t



What the robot see (uncorrupted data)



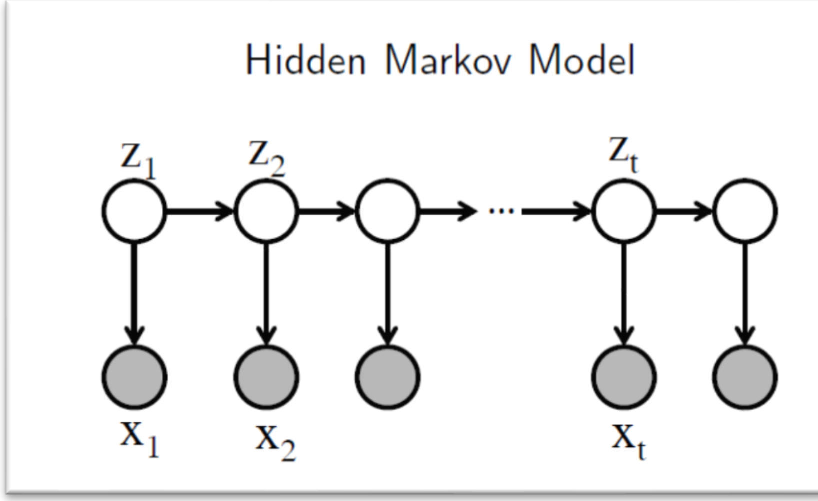
What the robot see (corrupted data)

Noisy observation of hidden state

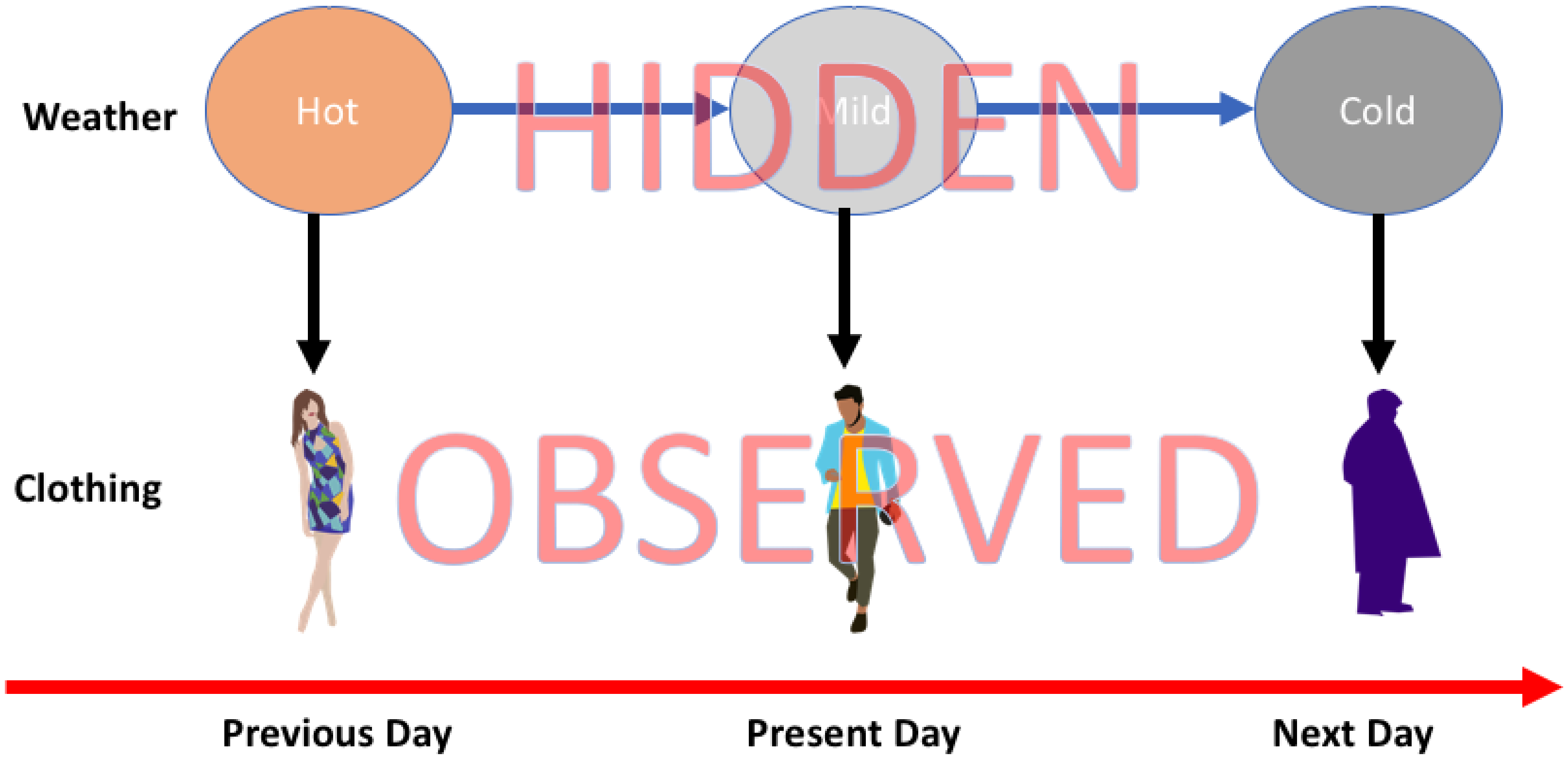
- Let's denote the observation at time t by O_t
- O_t is noisily determined depending on the current state
- Assume that O_t is conditionally independent of $\{q_{t-1}, q_{t-2}, \dots, q_0, O_{t-1}, O_{t-2}, \dots, O_1, O_0\}$ given q_t
- In other words

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

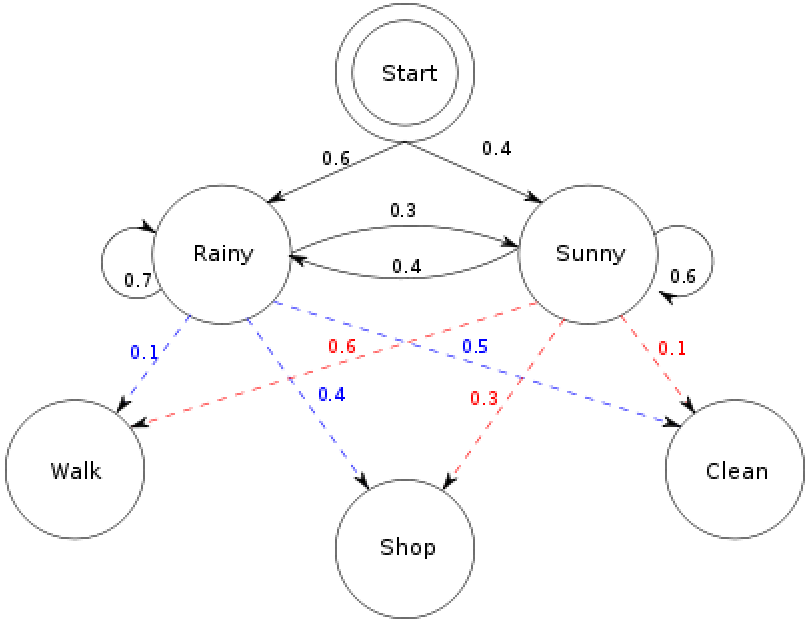
$$P(O_t = X \mid q_t = s_i, \text{any earlier history})$$

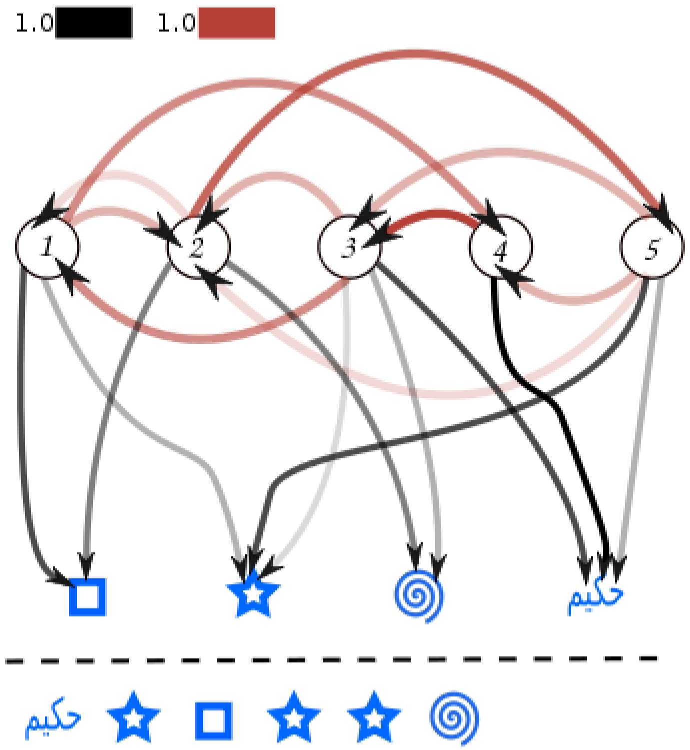


Example


Weather
Hot
Mild
Cold
HIDDEN
Clothing
OBSERVED
Previous Day
Present Day
Next Day

Example (cont.)





Hidden Markov models

- The robot with noisy sensors is a good example

- Question 1: (Evaluation) State estimation:
  - what is $P(q_t = S_i | O_1, \ldots, O_t)$
- Question 2: (Inference) Most probable path:
  - Given $O_1, \ldots, O_t$, what is the most probable path of the states? And what is the probability?
- Question 3: (Leaning) Learning HMMs:
  - Given $O_1, \ldots, O_t$, what is the maximum likelihood HMM that could have produced this string of observations?
  - MLE

Application of HMM

- Robot planning + sensing when there's uncertainty
- Speech recognition/understanding
  - Phones → Words, Signal → phones
- Human genome project
- Consumer decision modeling
- Economics and finance
- …

Basic operations in HMMs

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

Problem	Algorithm	Complexity
Evaluation: Calculating $P(q_t=S_i \mid O_1 O_2 \ldots O_t)$	Forward-Backward	$O(TN^2)$
Inference: Computing $Q^* = argmax_Q P(Q\|O)$	Viterbi Decoding	$O(TN^2)$
Learning: Computing $\lambda^* = argmax_\lambda P(O\|\lambda)$	Baum-Welch (EM)	$O(TN^2)$

T = # timesteps, N = # states

Formal definition of HMM

- The states are labeled $S_1, S_2, \dots, S_N$
- For a particular trial, let
  - T be the number of observations
  - N be the number of states
  - M be the number of possible observations
  - $(\pi_1, \pi_2, \dots, \pi_N)$ is the starting state probabilities
  - $O = O_1 \dots O_T$ is a sequence of observations
  - $Q = q_1 q_2 \cdots q_t$ is a path of states
- Then $\lambda = \langle N, M, \{\pi_{i,}\}, \{a_{ij}\}, \{b_i(j)\} \rangle$ is the specification of an HMM
  - ➢The definition of a_{ij} and $b_i(j)$ will be introduced in next page

Formal definition of HMM (cont.)

- The definition of a_{ij} and $b_i(j)$

$$\left.\begin{array}{cccc} a_{11} & a_{22} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{array}\right\}$$

The state transition probabilities

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

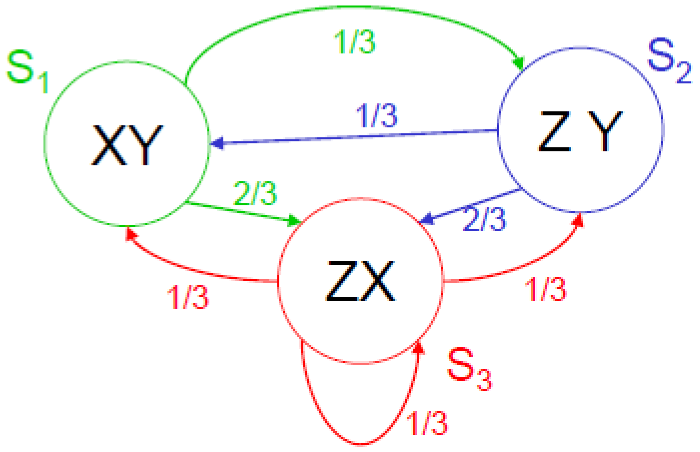
$$\left.\begin{array}{cccc} b_1(1) & b_1(2) & \cdots & b_1(M) \\ b_2(1) & b_2(2) & \cdots & b_2(M) \\ \vdots & \vdots & & \vdots \\ b_N(1) & b_N(2) & \cdots & b_N(M) \end{array}\right\}$$

The observation probabilities

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

Example

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



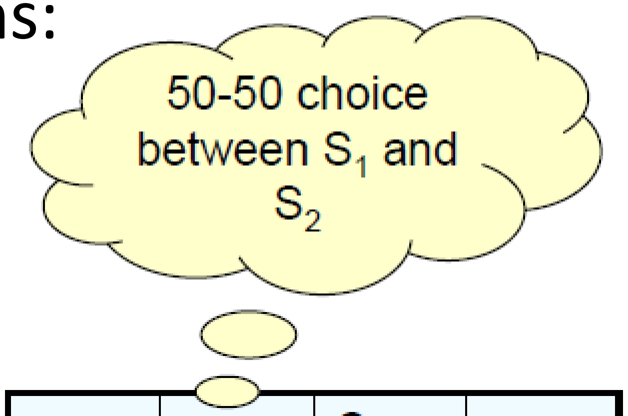
$N = 3$

$M = 3$

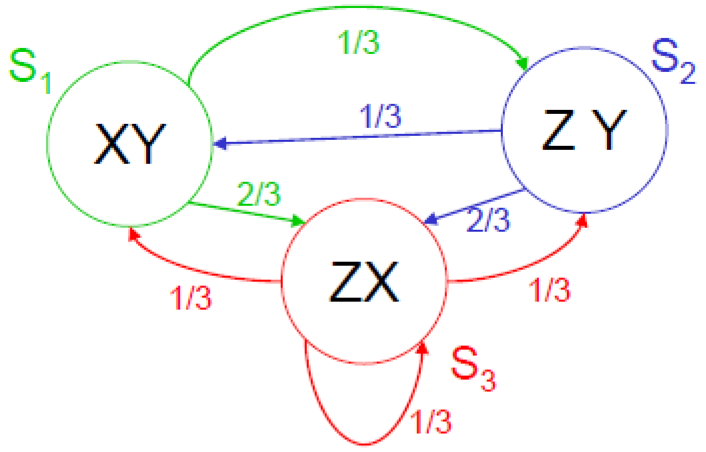
$\pi_1 = 1/2$	$\pi_2 = 1/2$	$\pi_3 = 0$
$a_{11} = 0$	$a_{12} = 1/3$	$a_{13} = 2/3$
$a_{12} = 1/3$	$a_{22} = 0$	$a_{13} = 2/3$
$a_{13} = 1/3$	$a_{32} = 1/3$	$a_{13} = 1/3$
$b_1(X) = 1/2$	$b_1(Y) = 1/2$	$b_1(Z) = 0$
$b_2(X) = 0$	$b_2(Y) = 1/2$	$b_2(Z) = 1/2$
$b_3(X) = 1/2$	$b_3(Y) = 0$	$b_3(Z) = 1/2$

Example (cont.)

- 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=	__	O_0=	__
q_1=	__	O_1=	__
q_2=	__	O_2=	__



$N = 3$
$M = 3$

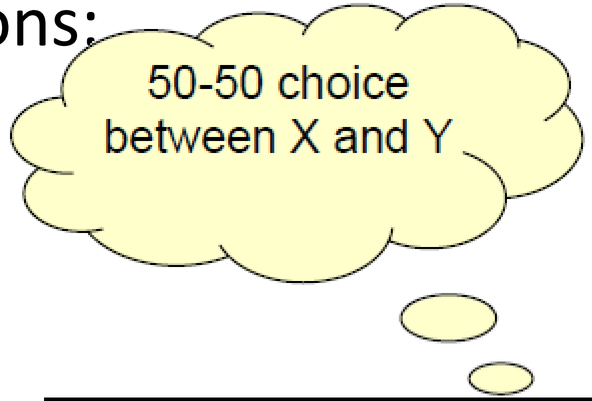
$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$	$a_{12} = ⅓$	$a_{13} = ⅔$
$a_{12} = ⅓$	$a_{22} = 0$	$a_{13} = ⅔$
$a_{13} = ⅓$	$a_{32} = ⅓$	$a_{13} = ⅓$

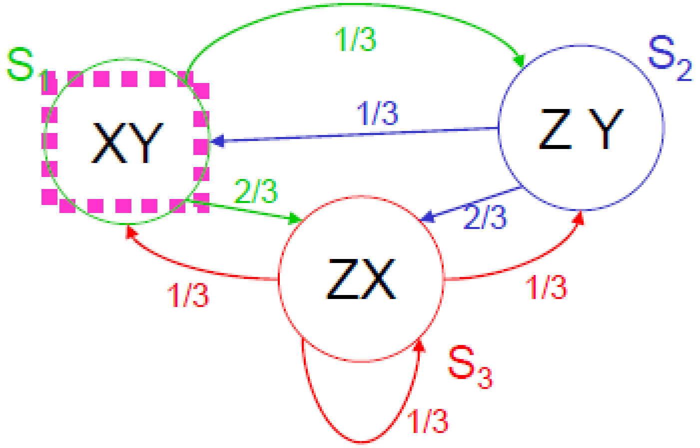
$b_1 (X) = ½$	$b_1 (Y) = ½$	$b_1 (Z) = 0$
$b_2 (X) = 0$	$b_2 (Y) = ½$	$b_2 (Z) = ½$
$b_3 (X) = ½$	$b_3 (Y) = 0$	$b_3 (Z) = ½$

Example (cont.)

- 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=	__
q_1=	__	O_1=	__
q_2=	__	O_2=	__



N = 3
M = 3

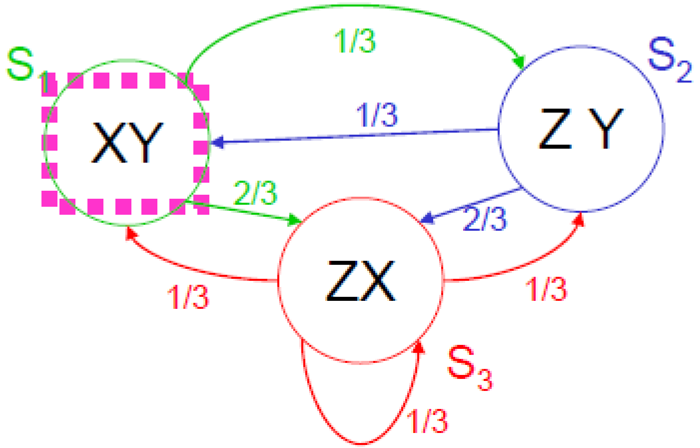
$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$
$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$
$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$
$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$
$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

Example (cont.)

- 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

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

N = 3
M = 3

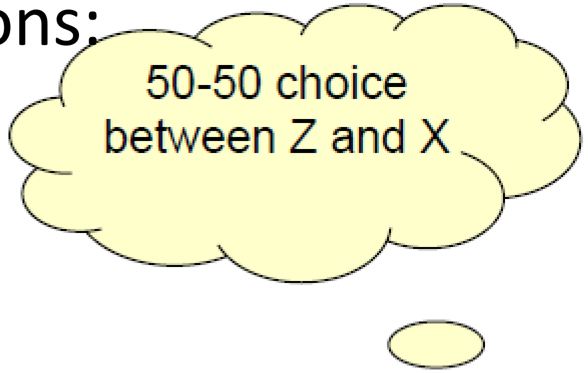
$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$
$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$
$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

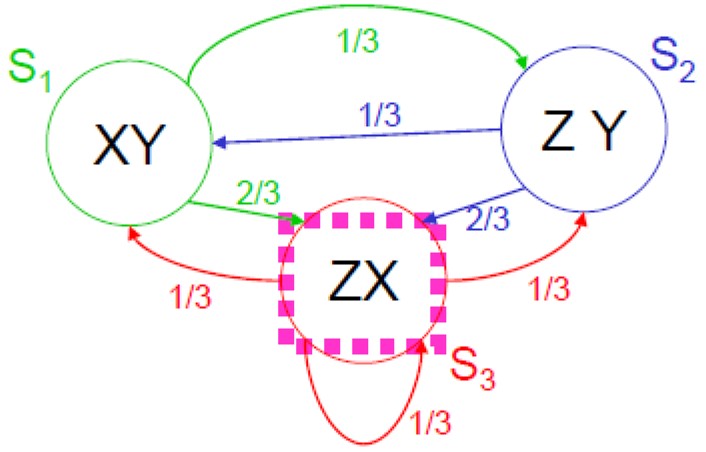
$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$
$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$
$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

Example (cont.)

- 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=	S_3	O_1=	__
q_2=	__	O_2=	__



N = 3
M = 3

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$
$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$
$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

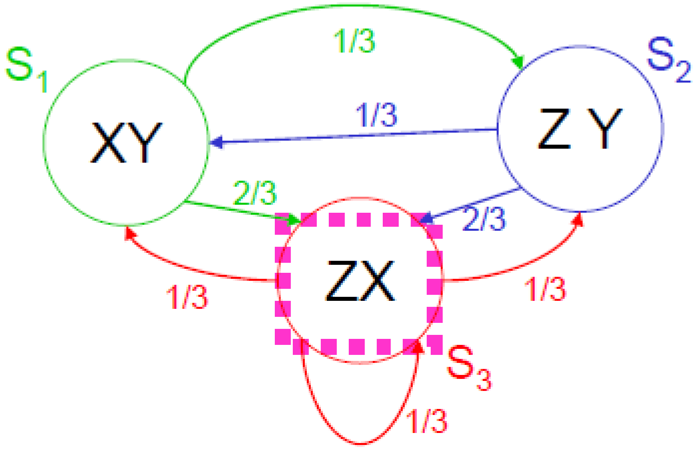
$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$
$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$
$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

Example (cont.)

- 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=	__



N = 3
M = 3

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$
$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$
$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

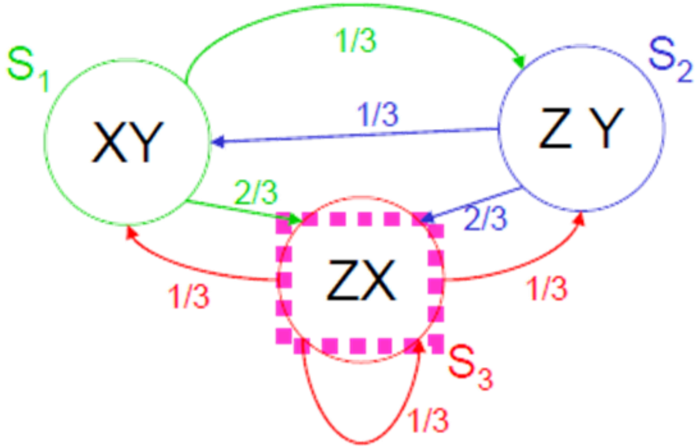
$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$
$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$
$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

Example (cont.)

- 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

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



N = 3

M = 3

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$

$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$

$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

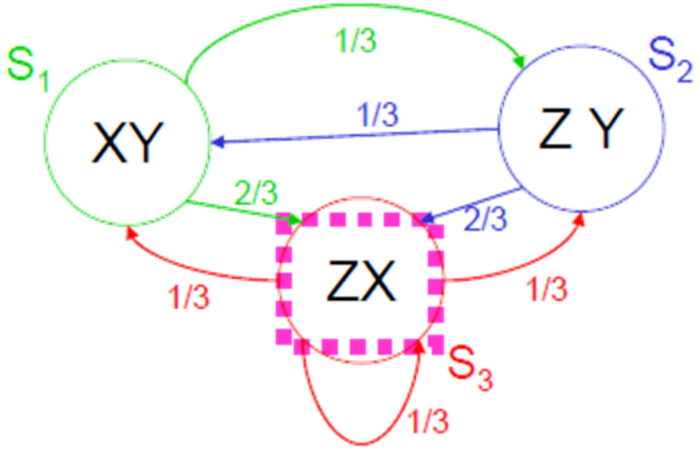
$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$

$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$

$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

Example (cont.)

- 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:



$N = 3$
$M = 3$

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$
$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$
$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

$b_1(X) = ½$ $b_1(Y) = ½$ $b_1(Z) = 0$
$b_2(X) = 0$ $b_2(Y) = ½$ $b_2(Z) = ½$
$b_3(X) = ½$ $b_3(Y) = 0$ $b_3(Z) = ½$

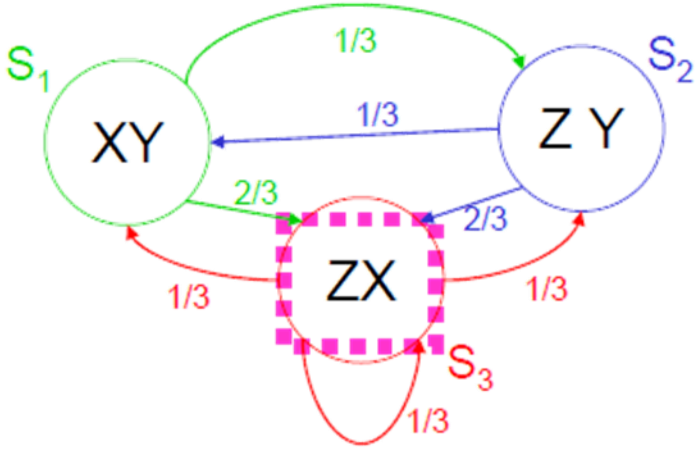
q_0=	S_1	O_0=	X
q_1=	S_3	O_1=	X
q_2=	S_3	O_2=	Z

Example (cont.)

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

This is what the observer has to work with…

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



$N = 3$

$M = 3$

$\pi_1 = ½$ $\pi_2 = ½$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = ⅓$ $a_{13} = ⅔$

$a_{12} = ⅓$ $a_{22} = 0$ $a_{13} = ⅔$

$a_{13} = ⅓$ $a_{32} = ⅓$ $a_{13} = ⅓$

$b_1 (X) = ½$ $b_1 (Y) = ½$ $b_1 (Z) = 0$

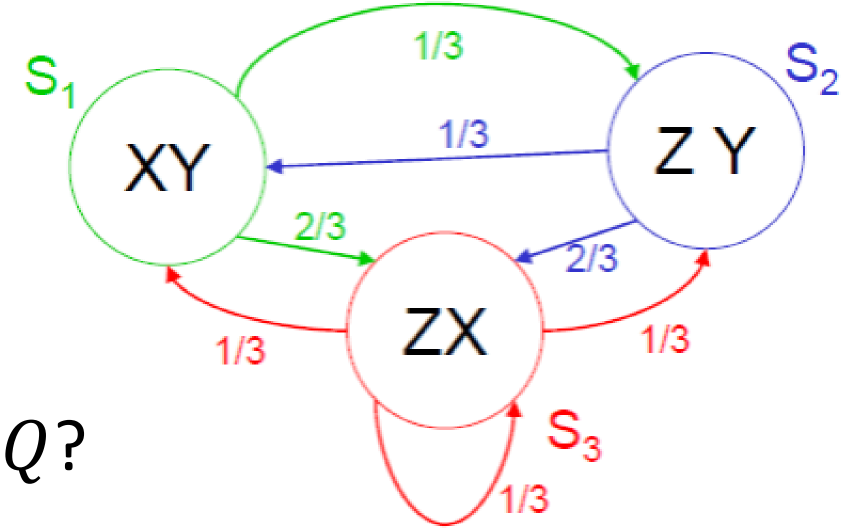
$b_2 (X) = 0$ $b_2 (Y) = ½$ $b_2 (Z) = ½$

$b_3 (X) = ½$ $b_3 (Y) = 0$ $b_3 (Z) = ½$

Probability of a series of observations

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

- 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?

Probability of a series of observations (cont.)

- $P(Q)$ for an arbitrary path Q

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

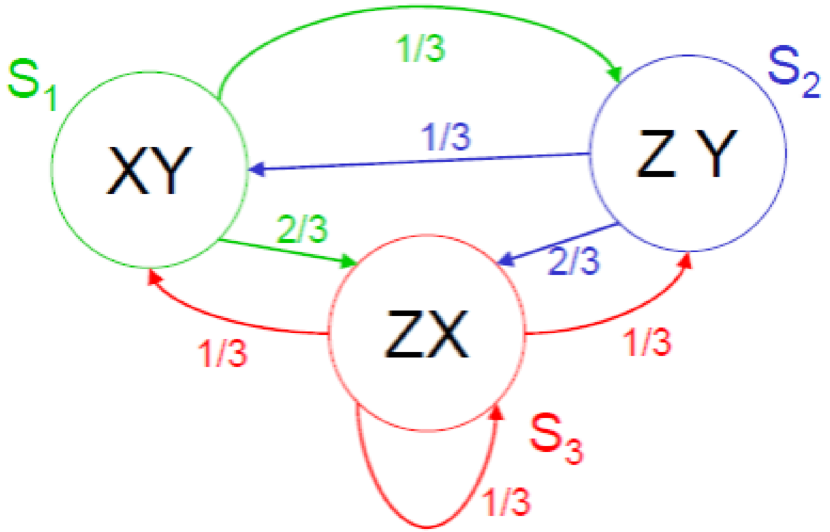
$= P(q_1)\ P(q_2,q_3|q_1)$ (chain rule)

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

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

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

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



Probability of a series of observations (cont.)

- $P(O|Q)$ for an arbitrary path Q

P(O|Q)

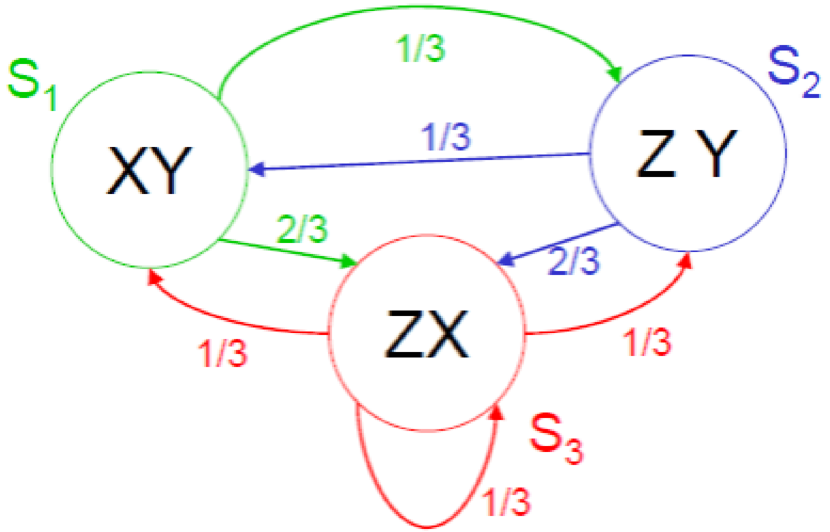
$= P(O_1\ O_2\ O_3\ |q_1\ q_2\ q_3\ )$

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

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

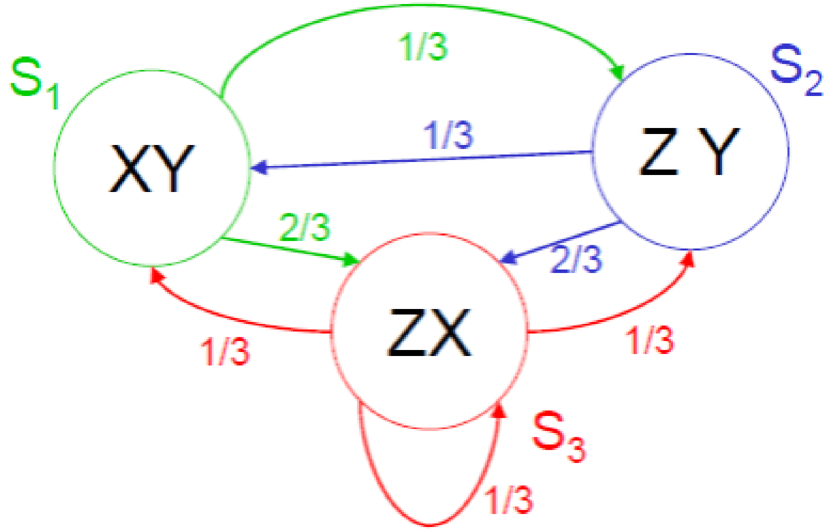
$= P(X|\ S_1)\ P(X|\ S_3)\ P(Z|\ S_3) =$

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



Probability of a series of observations (cont.)

- Computation complexity of the slow stupid answer:
  - $P(O)$ would require 27 $P(Q)$ and 27 $P(O|Q)$
  - A sequence of 20 observations would need 3^{20}=3.5 billion $P(Q)$ and 3.5 billion $P(O|Q)$
- So we have to find some smarter answer



Probability of a series of observations (cont.)

- Smart answer (based on dynamic programming)
- Given observations $O_1 O_2 \dots O_T$
- Define: $\alpha_t(i) = P(O_1\ O_2 \dots O_t \wedge q_t = S_i \mid \lambda)$ 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 the example, what is $\alpha_2(3)$?

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

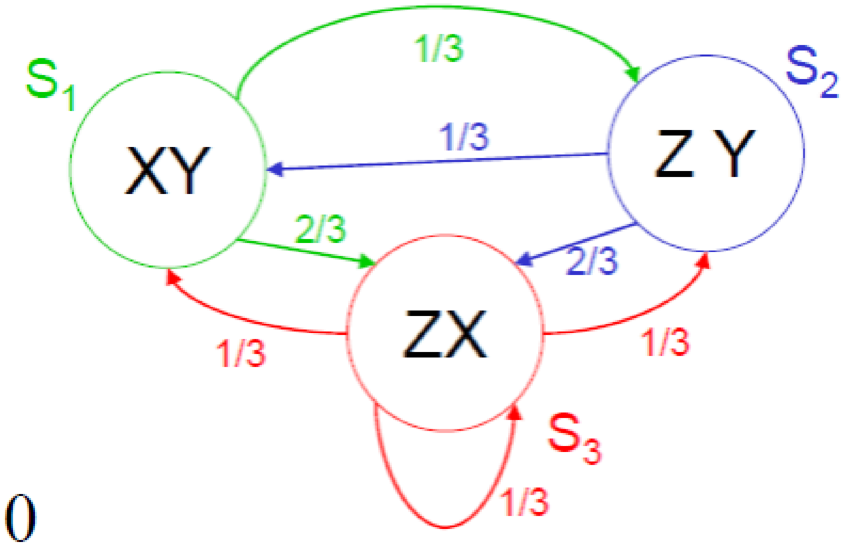
$$
\begin{aligned}
\alpha_1(i) &= \mathrm{P}(O_1 \wedge q_1 = S_i) \\
&= \mathrm{P}(q_1 = S_i)\mathrm{P}(O_1 | q_1 = S_i) \\
&= \qquad\qquad\qquad\qquad \text{what?} \\
\alpha_{t+1}(j) &= \mathrm{P}(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j) \\
&= \sum_{i=1}^{N} \mathrm{P}(O_1 O_2 \ldots O_t \wedge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

$\alpha_t(i)$ in the 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 see $O_1 O_2 O_3 = XXZ$

$$\alpha_1(1) = \frac{1}{4} \qquad \alpha_1(2) = 0 \qquad \alpha_1(3) = 0$$

$$\alpha_2(1) = 0 \qquad \alpha_2(2) = 0 \qquad \alpha_2(3) = \frac{1}{12}$$

$$\alpha_3(1) = 0 \qquad \alpha_3(2) = \frac{1}{72} \qquad \alpha_3(3) = \frac{1}{72}$$

Easy question

- We can cheaply compute

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

- (How) can we cheaply compute

$$P(O_1O_2\ldots O_t) \quad ?$$

- (How) can we cheaply compute

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

Easy question (cont.)

- We can cheaply compute

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

- (How) can we cheaply compute

$$P(O_1O_2\ldots O_t) \quad ?$$

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

- (How) can we cheaply compute

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

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

Recall: Hidden Markov models

- The robot with noisy sensors is a good example

- Question 1: (Evaluation) State estimation:
  - what is $P(q_t = S_i | O_1, \ldots, O_t)$
- Question 2: (Inference) Most probable path:
  - Given $O_1, \ldots, O_t$, what is the most probable path of the states? And what is the probability?
- Question 3: (Leaning) Learning HMMs:
  - Given $O_1, \ldots, O_t$, what is the maximum likelihood HMM that could have produced this string of observations?
  - MLE

Most probable path (MPP) given observations

What's most probable path given $O_1 O_2 \ldots O_T$, i.e.

What is $\underset{Q}{\operatorname{argmax}} \; P(Q | O_1 O_2 \ldots O_T)$?

Slow, stupid answer :

$$\underset{Q}{\operatorname{argmax}} \; P(Q | O_1 O_2 \ldots O_T)$$

$$= \underset{Q}{\operatorname{argmax}} \; \frac{P(O_1 O_2 \ldots O_T | Q) P(Q)}{P(O_1 O_2 \ldots O_T)}$$

$$= \underset{Q}{\operatorname{argmax}} \; P(O_1 O_2 \ldots O_T | Q) P(Q)$$

Efficient MPP computation

- We're going to compute the following variables

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

- It's the probability of the path of length $t-1$ with the maximum chance of doing all these things OCCURING and ENDING UP IN STATE S_i and PRODUCING OUTPUT $O_1 \ldots O_t$

- DEFINE: $mpp_t(i)$ = that path
- So: $\delta_t(i)$= Prob($mpp_t(i)$)

The Viterbi algorithm

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

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

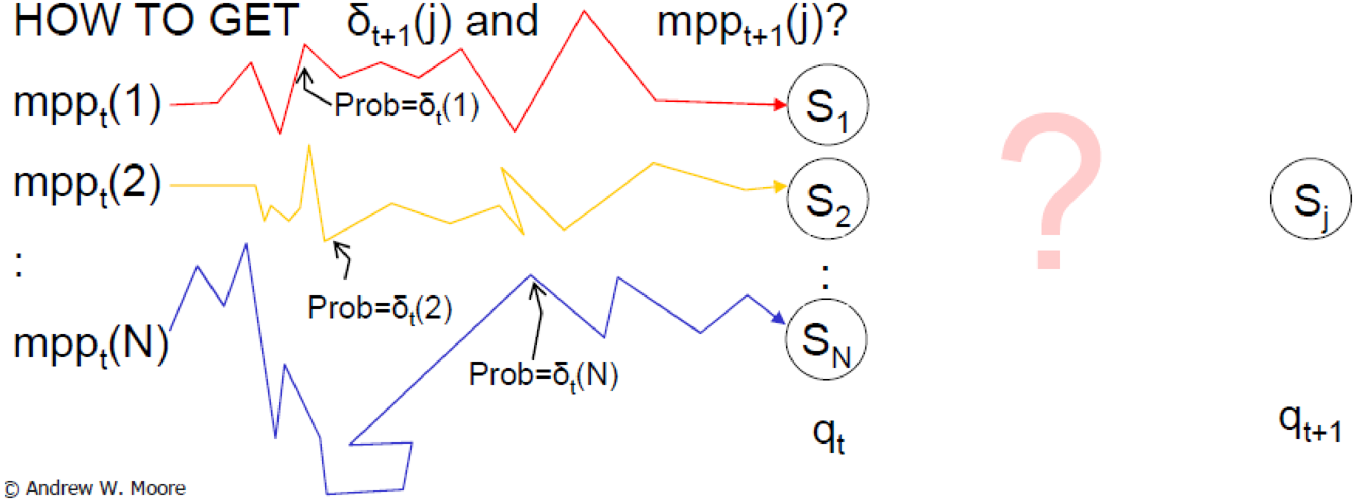
$$\delta_1(i) = \max_{\text{one choice}} P(q_1 = S_i \wedge O_1)$$

$$= P(q_1 = S_i) P(O_1 | q_1 = S_i)$$

$$= \pi_i b_i(O_1)$$

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

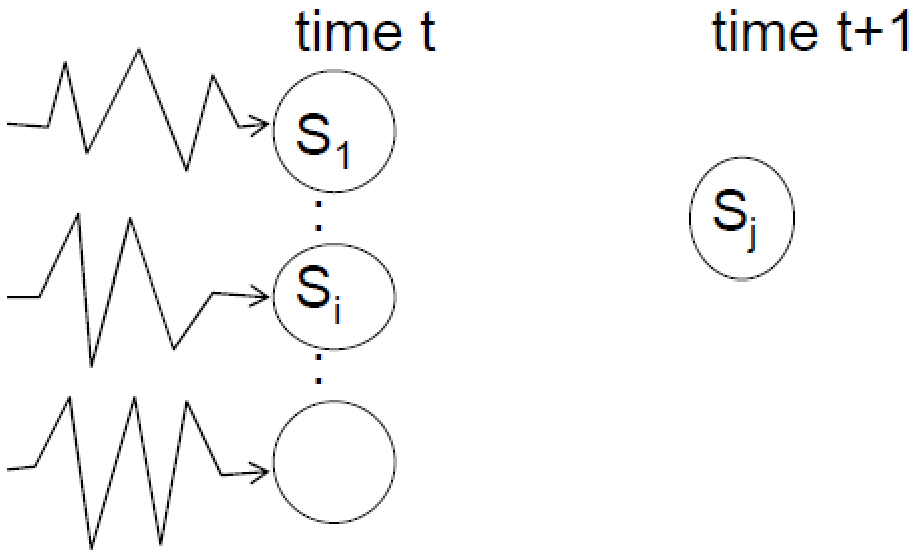


The Viterbi algorithm (cont.)

The most prob path with last two states S_i S_j

is

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

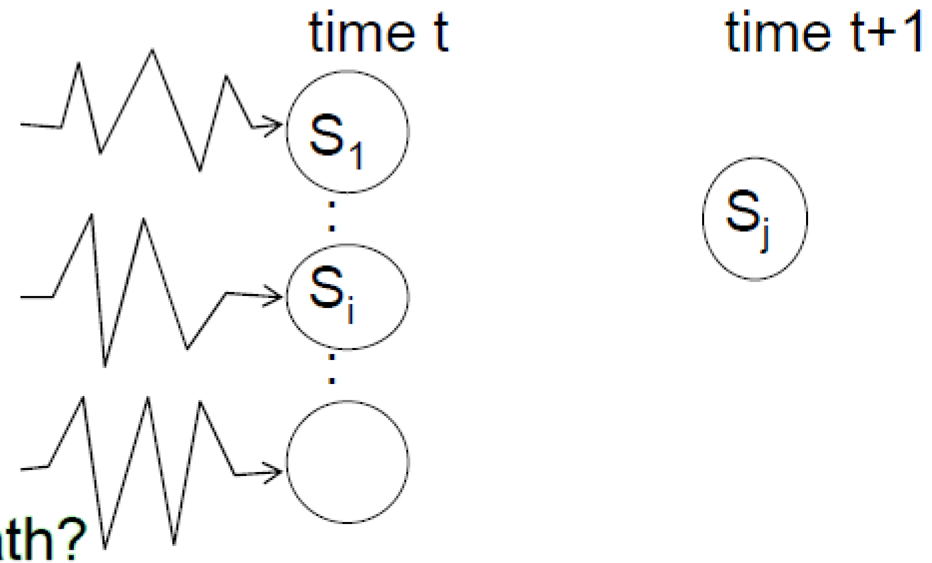


The Viterbi algorithm (cont.)

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?

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

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

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

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

The Viterbi algorithm (cont.)

- Summary

What is the prob of that path?

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

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

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

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

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

Recall: Hidden Markov models

- The robot with noisy sensors is a good example

- Question 1: (Evaluation) State estimation:
  - what is $P(q_t = S_i | O_1, \dots, O_t)$
- Question 2: (Inference) Most probable path:
  - Given $O_1, \dots, O_t$, what is the most probable path of the states? And what is the probability?
- Question 3: (Leaning) Learning HMMs:
  - Given $O_1, \dots, O_t$, what is the maximum likelihood HMM that could have produced this string of observations?
  - MLE

Inferring an HMM

- Remember, we've been doing things like

$$P(O_1\ O_2\ ..\ O_T \mid \lambda\ )$$

- That "λ" is the notation for our HMM parameters
- Now we want to estimate λ from the observations
- AS USUAL: We could use

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

(ii) BAYES

Work out $P(\ \lambda \mid O_1\ ..\ O_T\ )$

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

Max likelihood HMM estimation

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

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

$\gamma_t(i)$ and $\varepsilon_t(i,j)$ 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

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

Max likelihood HMM estimation

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

$$= \text{Estimate of Prob}\left(\text{Next state S}_\text{j} \middle| \text{This state S}_\text{i}\right)$$

We can re - estimate

$$\text{a}_{\text{ij}} \leftarrow \frac{\sum \varepsilon_t(i,j)}{\sum \gamma_t(i)}$$

We can also re - estimate

$$\text{b}_\text{j}(O_k) \leftarrow \cdots \qquad \text{(See Rabiner)}$$

Max likelihood HMM estimation

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

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

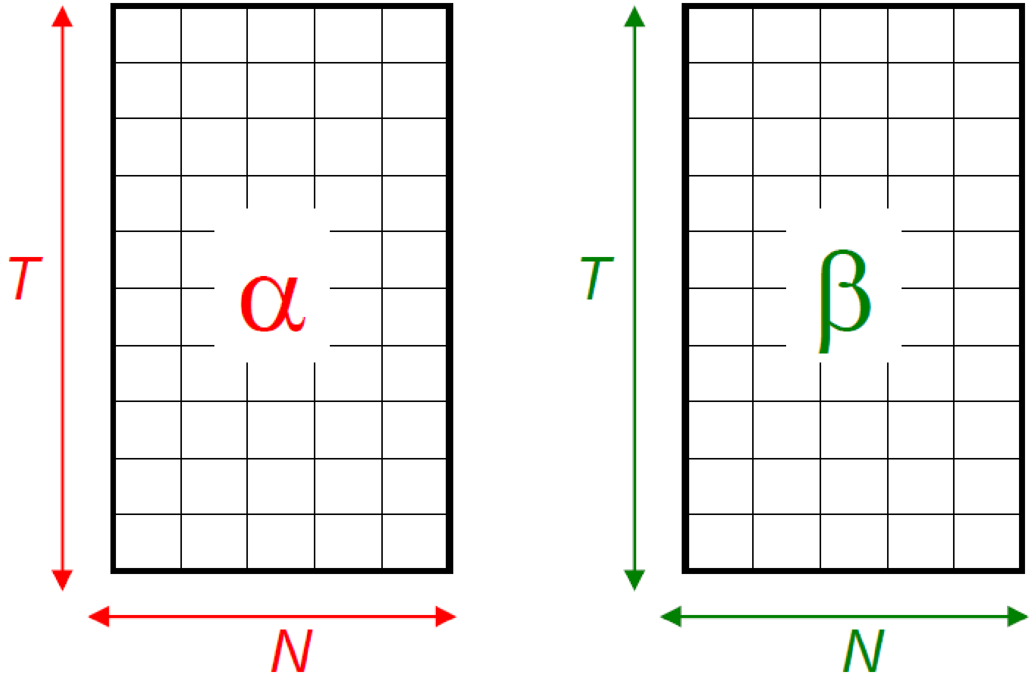
$$= \frac{\sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}{\sum_{k=1}^{N} \sum_{t=1}^{T} P(q_{t+1} = s_k, q_t = s_i \mid \lambda^{\text{old}}, O_1, O_2, \cdots O_T)}$$

$$= \frac{S_{ij}}{\sum_{k=1}^{N} S_{ik}} \quad \text{where} \quad S_{ij} = \sum_{t=1}^{T} P(q_{t+1} = s_j, q_t = s_i, O_1, \cdots O_T \mid \lambda^{\text{old}})$$

$$= a_{ij} \sum_{t=1}^{T} \alpha_t(i) \beta_{t+1}(j) b_j(O_{t+1})$$

Max likelihood HMM estimation

We want $a_{ij}^{\text{new}} = S_{ij} \Big/ \sum_{k=1}^{N} S_{ik}$ 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 $i \to j$

- If we knew the quantities such as
  - Expected number of times in state i
  - Expected number of transitions $i \to j$
- We could compute the MAX LIKELIHOOD estimate of

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

EM for HMMs

1. Get your observations $O_1 \ldots O_T$
2. Guess your first λ estimate $\lambda(0)$, $k=0$
3. $k = k+1$
4. Given $O_1 \ldots O_T$, $\lambda(k)$ compute
   $\gamma_t(i)$, $\varepsilon_t(i,j)$ $\forall 1 \le t \le T$, $\forall 1 \le i \le N$, $\forall 1 \le j \le N$
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.**

EM for HMMs

- 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