Hidden Markov Models

Andrew W. Moore

Carnegie Mellon University

www.cs.cmu.edu/~awm/tutorials

with extensions from

Norbert Fuhr

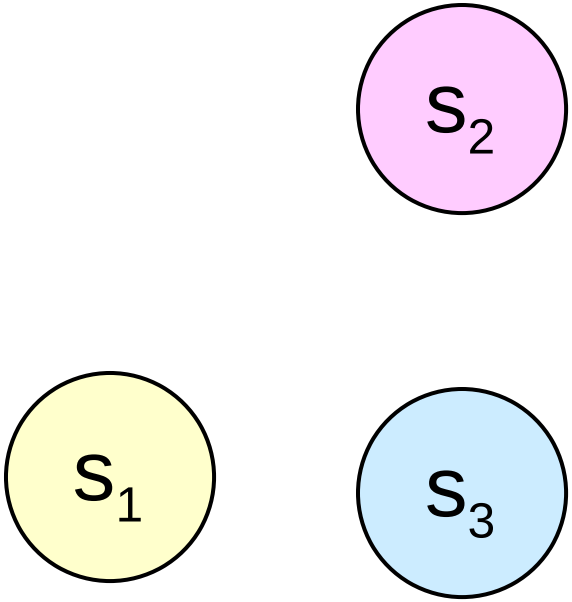
University of Duisburg-Essen

Copyright © 2001-2003, Andrew W. Moore Nov 29th, 2001

A Markov System

Has N states, called s_1, s_2 .. s_N

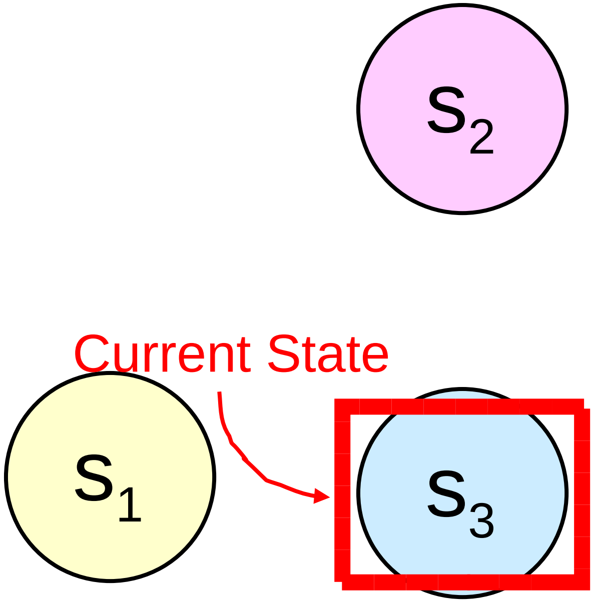
There are discrete timesteps, $t=0$, $t=1$, …



$N = 3$

$t=0$

A Markov System



$N = 3$

$t=0$

$q_t=q_0=s_3$

Has N states, called s_1, s_2 .. s_N

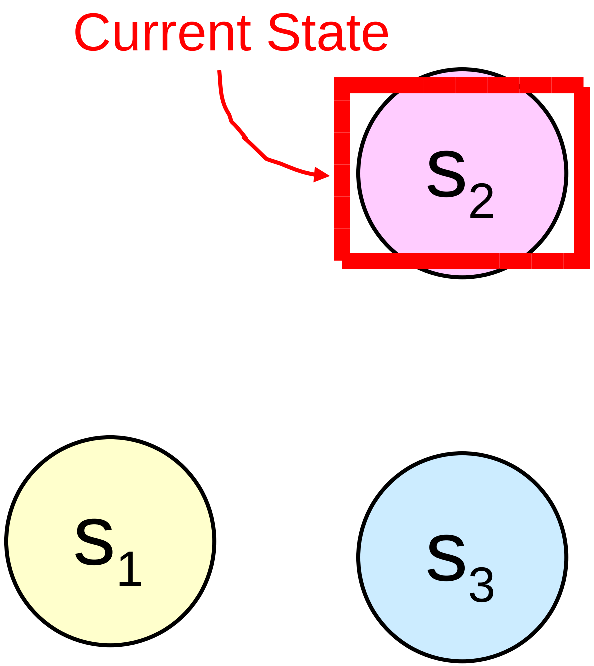
There are discrete timesteps, $t=0, t=1, \ldots$

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

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

Copyright © 2001-2003, 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, $t=0, t=1, …$

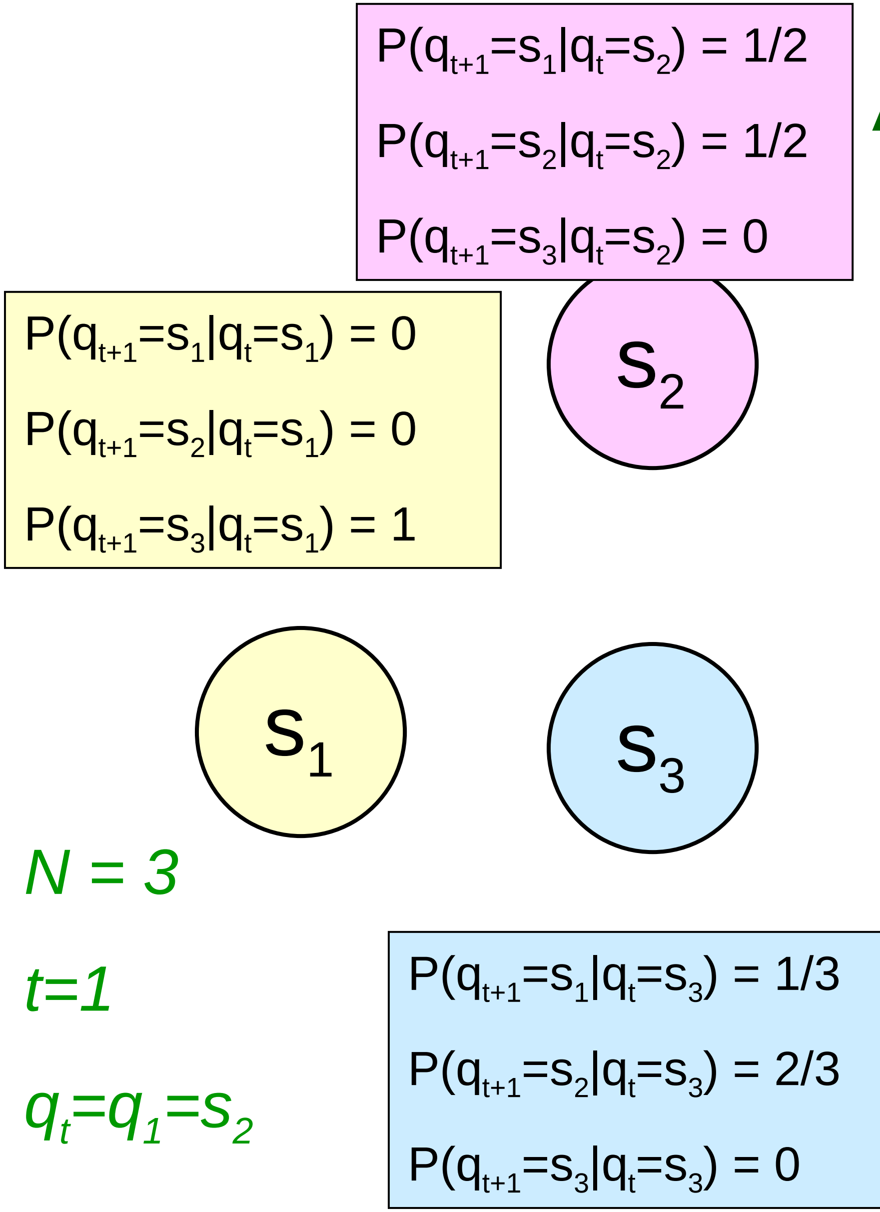
On the t'th timestep the system is in exactly one of the available states. Call it q_t

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

Between each timestep, the next state is chosen randomly.

Copyright © 2001-2003, 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: $q_t \in \{s_1, s_2 .. s_N \}$

Between each timestep, the next state is chosen randomly.

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

Copyright © 2001-2003, Andrew W. Moore

A Markov System

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

Between each timestep, the next state is chosen randomly.

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

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

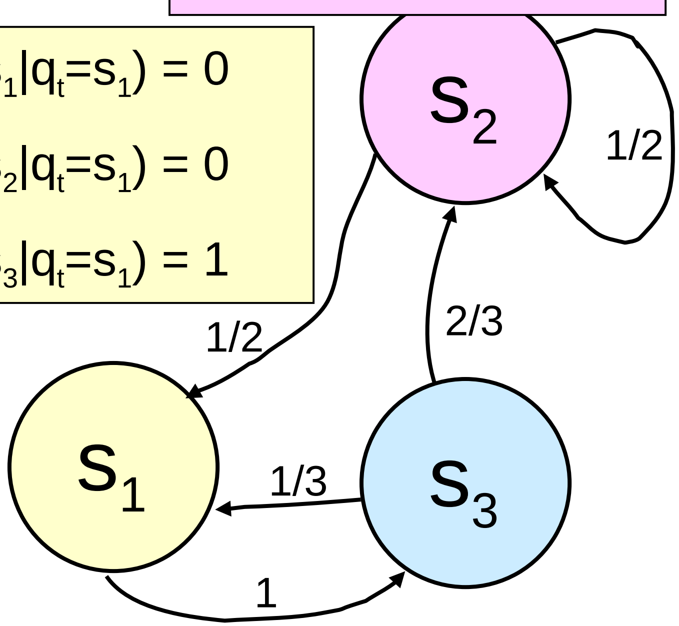
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$

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

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

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

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



$N = 3$

$t=1$

$q_t=q_1=s_2$

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

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

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

Often notated with arcs between states

Markov Property

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

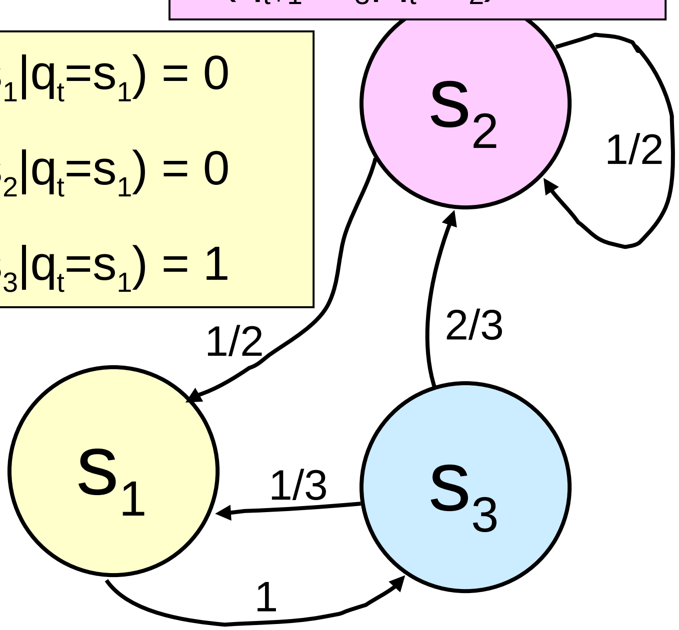
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$

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

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

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

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



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 $\{ q_{t-1}, q_{t-2}, \ldots q_1, q_0 \}$ given q_t.

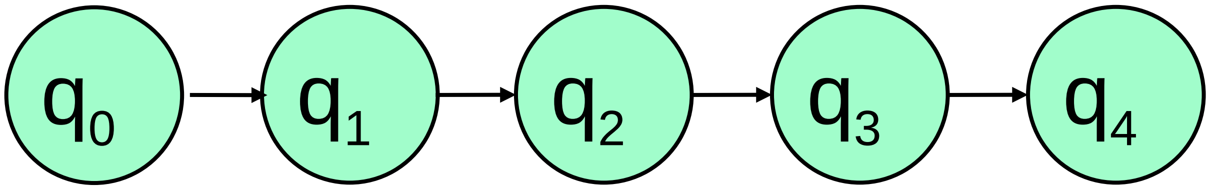
In other words:

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

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

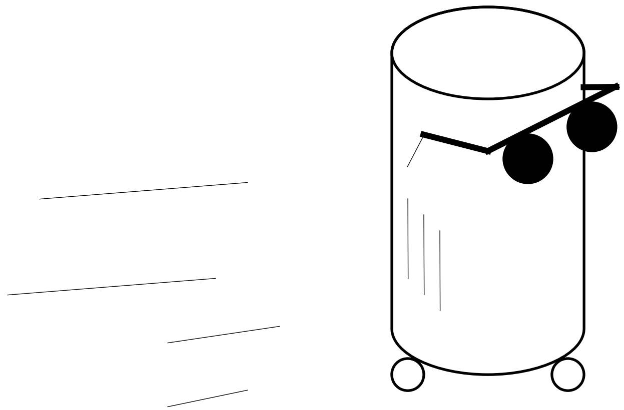
Copyright © 2001-2003, Andrew W. Moore

Markov Property: Representation

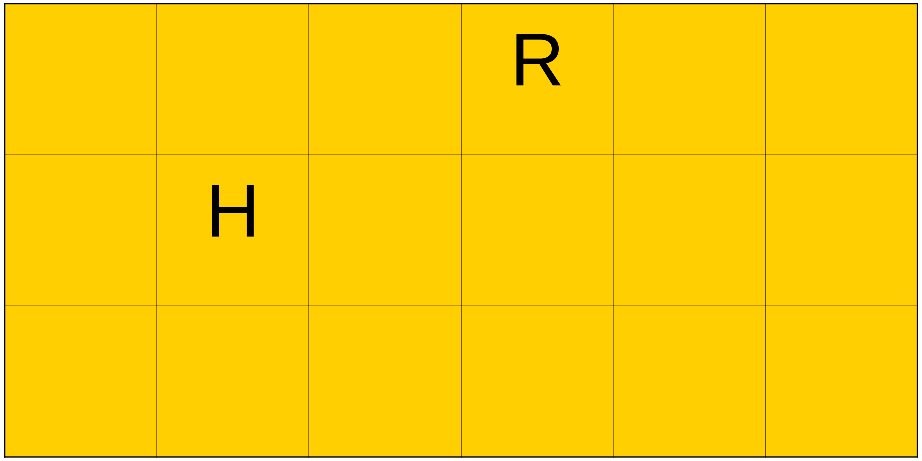


Copyright © 2001-2003, Andrew W. Moore

A Blind Robot



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



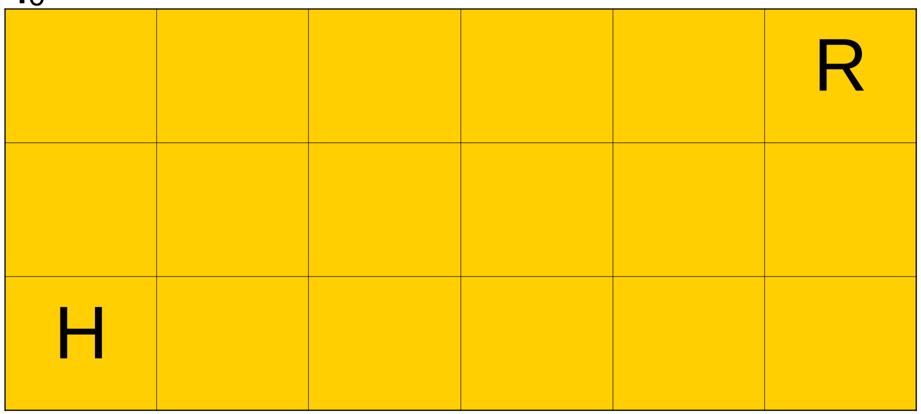
STATE q = $\left\langle \begin{array}{l}\text{Location of Robot,}\\ \text{Location of Human}\end{array} \right\rangle$

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

Copyright © 2001-2003, Andrew W. Moore

Dynamics of System

q_0 =



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 © 2001-2003, 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

Copyright © 2001-2003, Andrew W. Moore

What is $P(q_t = s)$? Too Slow

Step 1: Work out how to compute P(Q) for any path $Q = q_0\, q_1\, q_2\, q_3 \,..\, q_t$

Given we know the start state q_0

$$P(q_0\, q_1 \,..\, q_t) = P(q_0\, q_1 \,..\, q_{t-1})\, P(q_t | q_0\, q_1 \,..\, q_{t-1})$$

$$= P(q_0\, q_1 \,..\, q_{t-1})\, P(q_t | q_{t-1})$$

WHY?

$$= P(q_1|q_0)P(q_2|q_1)\ldots P(q_t|q_{t-1})$$

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 © 2001-2003, 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

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

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

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

- For each state s_i, define

  $p_t(i)$ = 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

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

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

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

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

Remember,

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

Copyright © 2001-2003, 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

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

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

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

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

- 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}				

Copyright © 2001-2003, 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

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

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

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

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

- 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 © 2001-2003, Andrew W. Moore

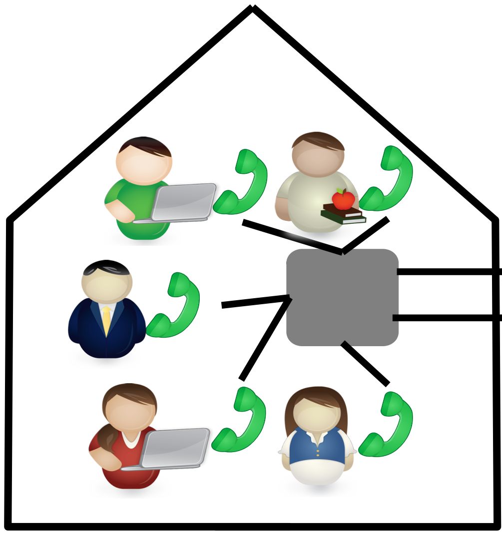
What is $P(q_t = s)$?

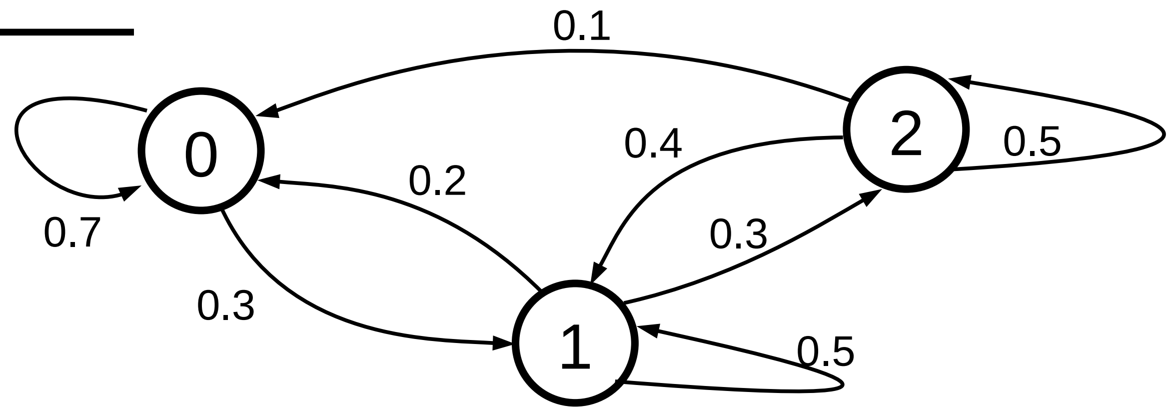
- $p_t(i)$ = Prob. state is s_i at time t = $P(q_t = s_i)$
- $\boldsymbol{p}_t = (p_t(1), \ldots p_t(N))$: vector or state probabilities
- $\boldsymbol{A} = (a_{ij})$: Matrix of transition probabilities

$$\forall j \quad p_{t+1}(j) = P(q_{t+1} = s_j) = \sum_{i=1}^{N} a_{ij} p_t(i)$$

$$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t \boldsymbol{A} = \boldsymbol{p}_0 \boldsymbol{A}^{t+1}$$

Steady state probability



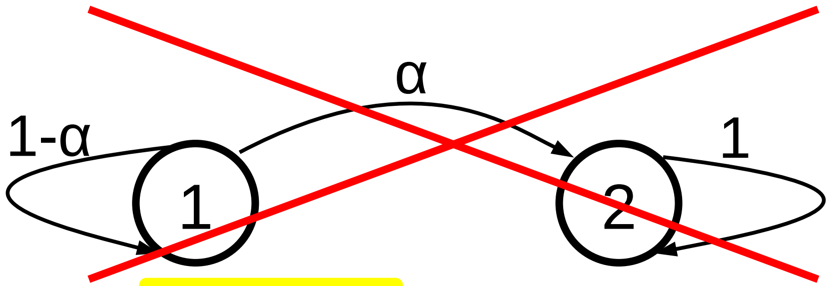


- What is the „average“ distribution over the states?
- -> steady state probability

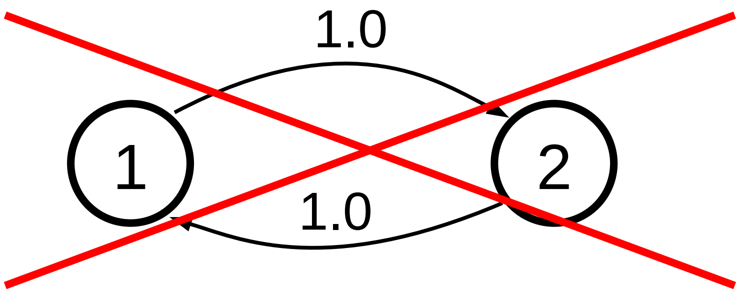
Copyright © 2001-2003, Andrew W. Moore

Steady State in MMs

- There exists a steady state, if
  - Markov model is irreducible, and



  - Markov model is aperiodic



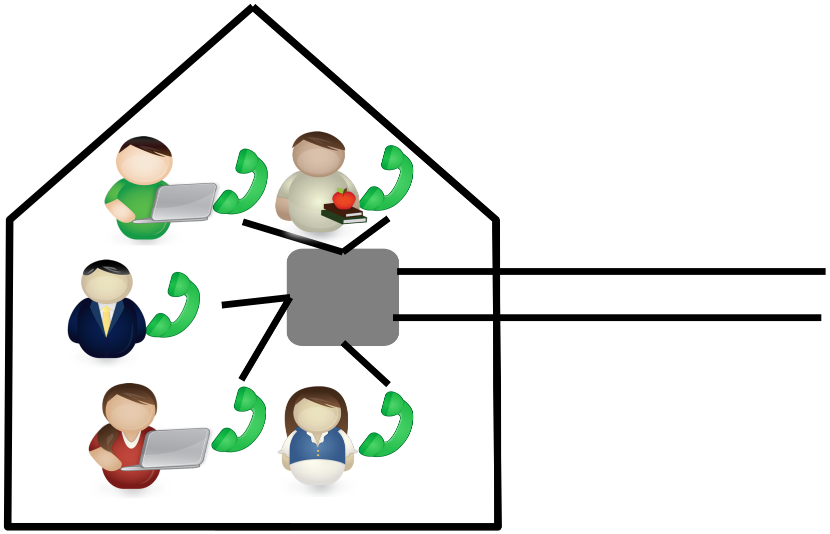
- Then there exists a steady state distribution $\boldsymbol{p}_t$ that is independent of t, which we denote as $\boldsymbol{\pi}$

Computing the Steady State Distribution

- $\boldsymbol{\pi = \pi A}$
  - → $\boldsymbol{\pi}$ is left eigenvector of $\boldsymbol{A}$
  - → solve linear equation system
- Alternative method:
  - Start with arbitrary vector $\boldsymbol{\pi_0}$
  - compute iteratively $\boldsymbol{\pi_1 = \pi_0 A}$, $\boldsymbol{\pi_2 = \pi_1 A}$, …
    $\boldsymbol{\pi_t = \pi_0 A^t}$
  - → „power method“

Copyright © 2001-2003, Andrew W. Moore

Application of Power Method



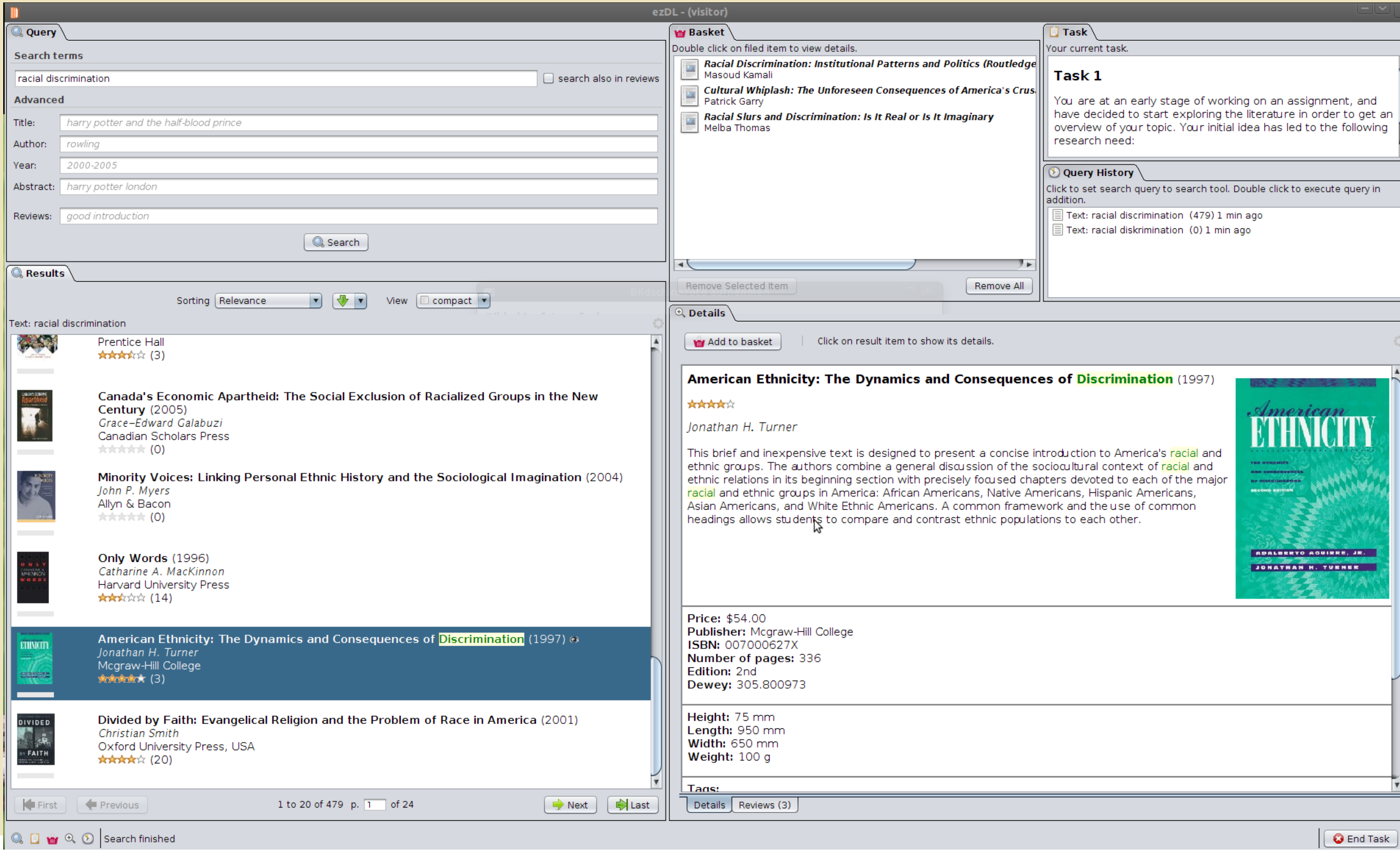
$$\begin{pmatrix} 0.7 & 0.3 & 0 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.4 & 0.5 \end{pmatrix}$$

	π_0	π_1	π_2
$\vec{\pi}$	1.00	0.00	0.00
$\vec{\pi}.A$	0.70	0.30	0.00
$\vec{\pi}.A^2$	0.55	0.36	0.09
$\vec{\pi}.A^3$	0.47	0.38	0.15
$\vec{\pi}.A^4$	0.42	0.39	0.19
$\vec{\pi}.A^5$	0.39	0.40	0.21
$\vec{\pi}.A^6$	0.37	0.40	0.23
$\vec{\pi}.A^7$	0.36	0.40	0.23
$\vec{\pi}.A^8$	0.36	0.40	0.24
$\vec{\pi}.A^9$	0.36	0.40	0.24
$\vec{\pi}.A^{10}$	0.35	0.40	0.24
$\vec{\pi}.A^{11}$	0.35	0.41	0.24
$\vec{\pi}.A^{12}$	0.35	0.41	0.24

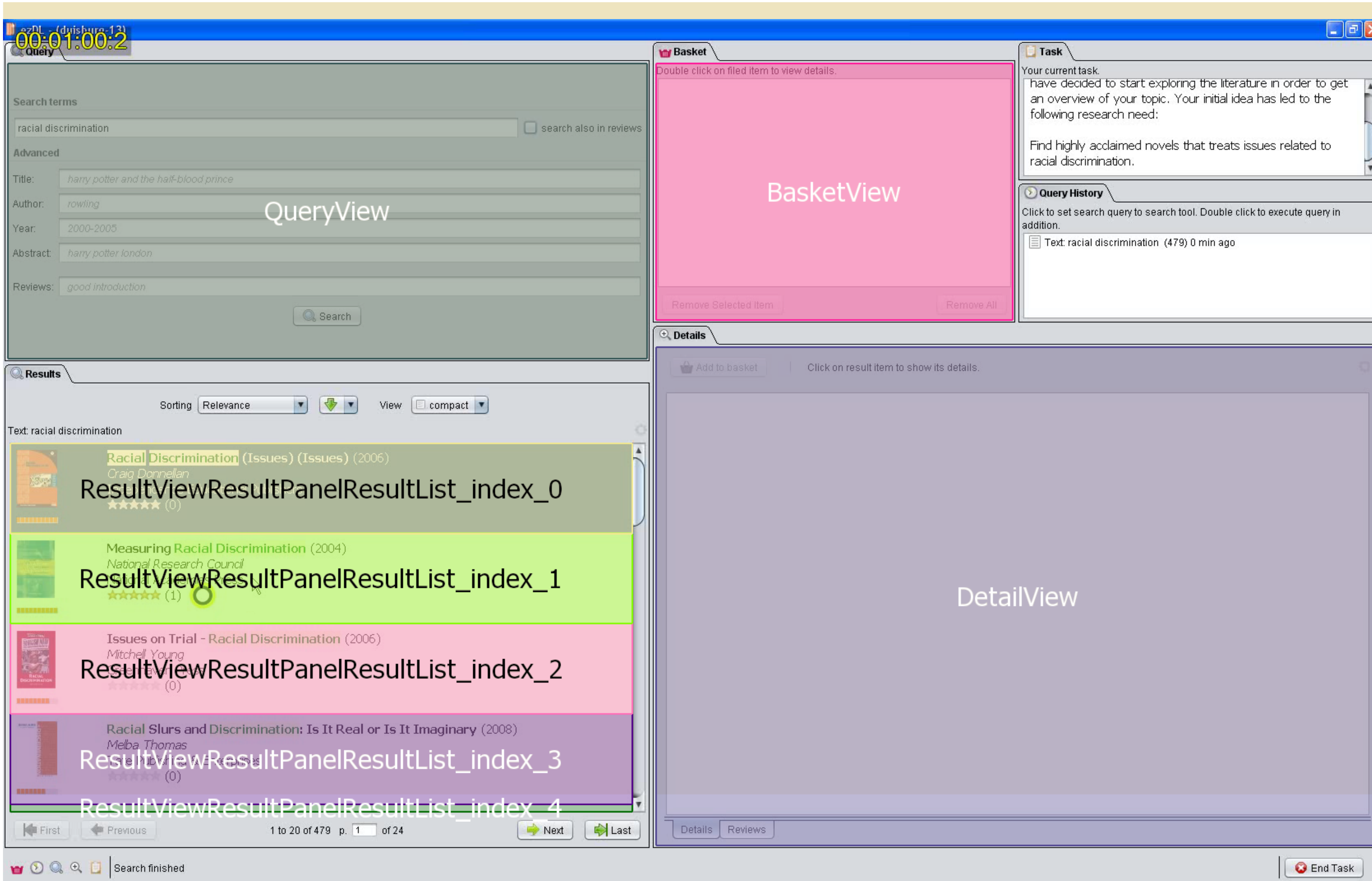
Copyright © 2001-2003, Andrew W. Moore

Markov Modelling of Book Search

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

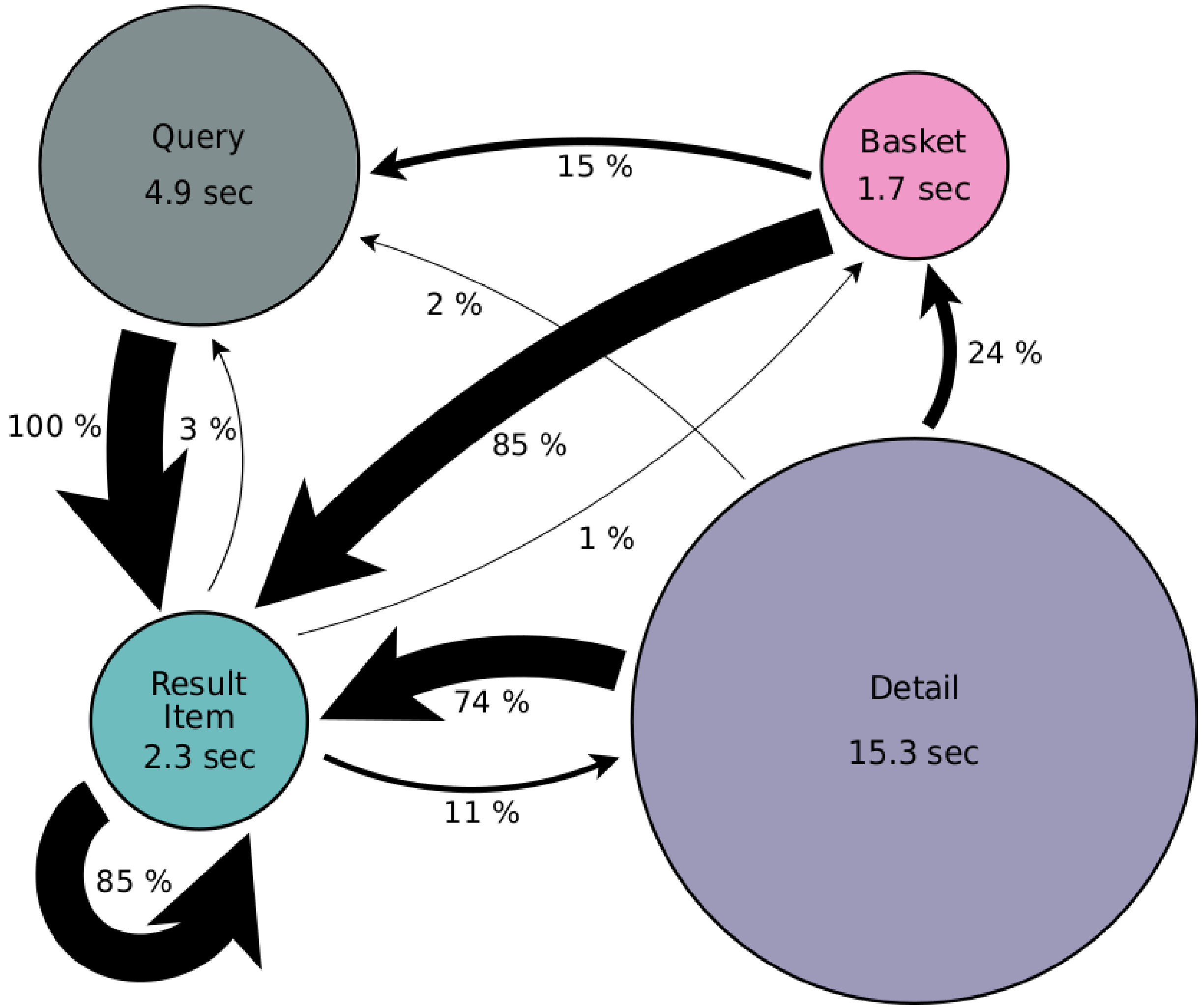


Eye Tracking: Areas of Interest



Markov Modeling

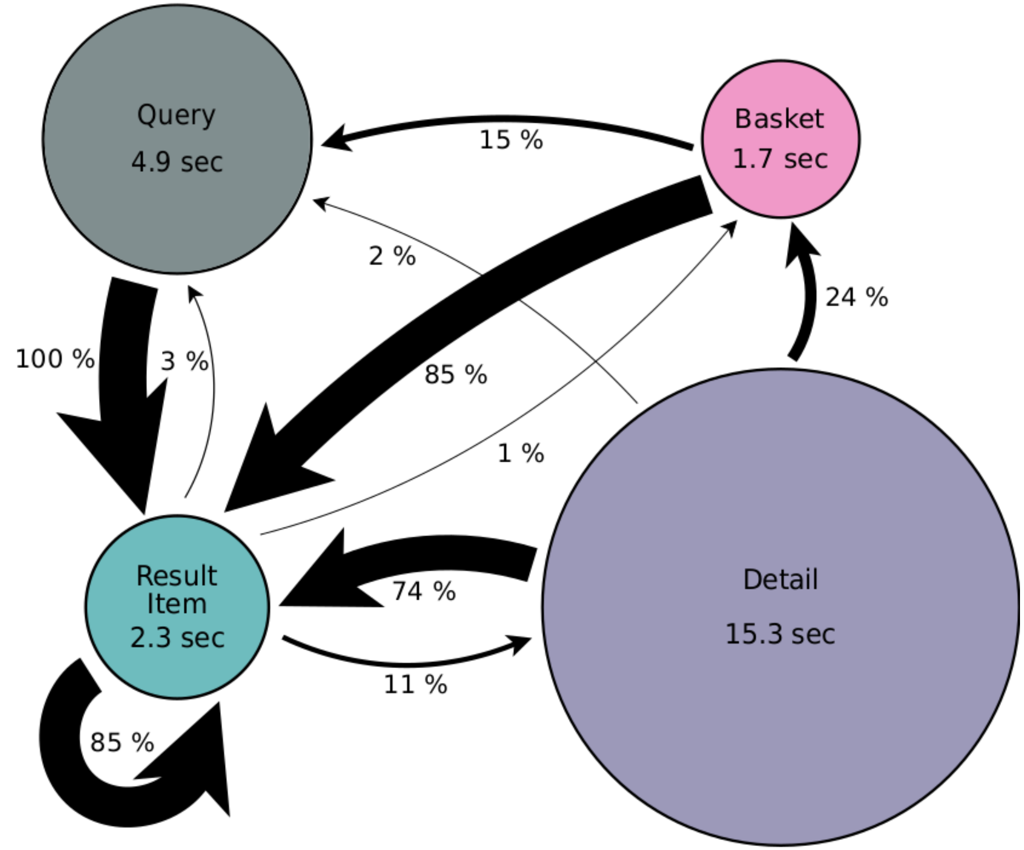
- Book search task
- 12 user sessions
- Consider eye tracking + system logs



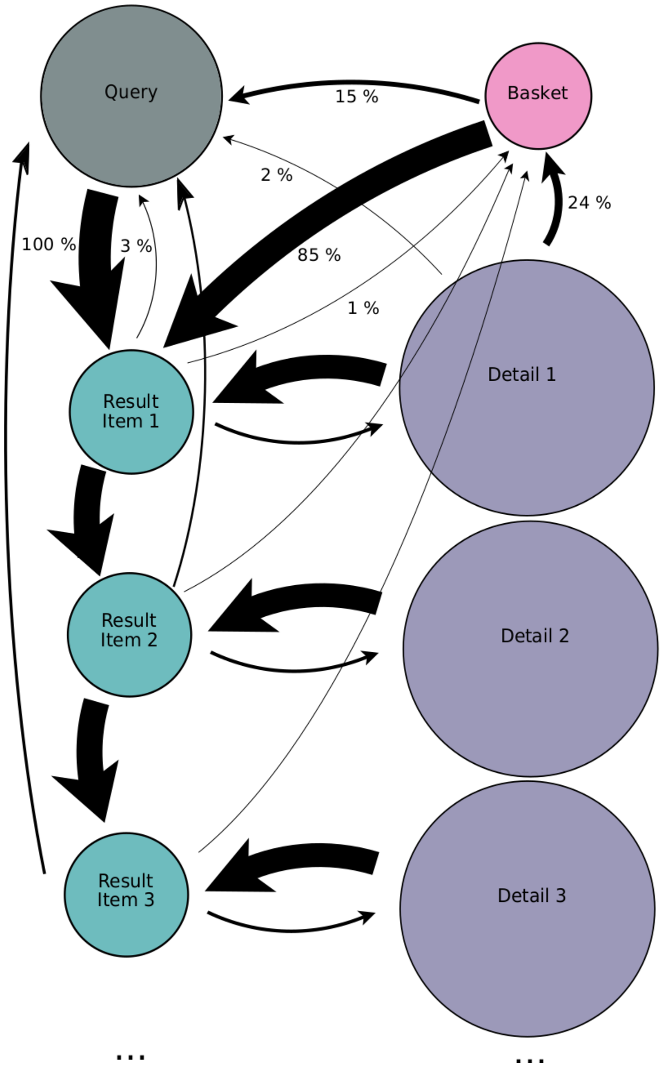
Time to Basket (Relevance)

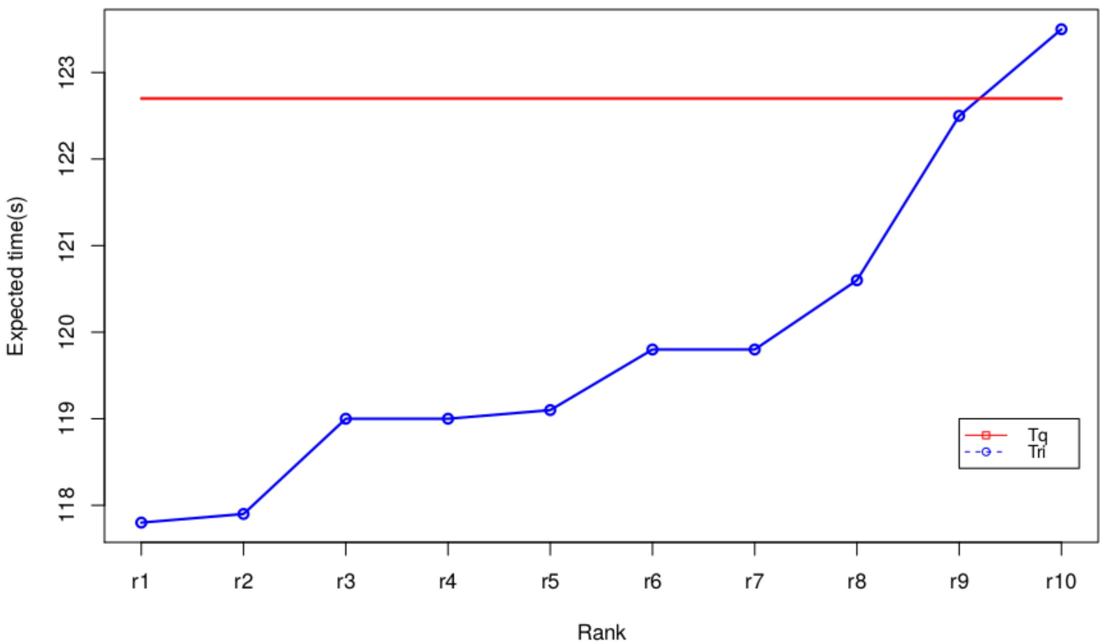
$$T_q = t_q + p_{qr} T_r$$
$$T_r = t_r + p_{rq} T_q + p_{rr} T_r + p_{rd} T_d$$
$$T_d = t_d + p_{dq} T_q + p_{dr} T_r$$

$$T_q = 122.7\, s$$
$$T_r = 117.8\, s$$
$$T_d = 104.9\, s$$



Guidance: Modeling Retrieval Ranks

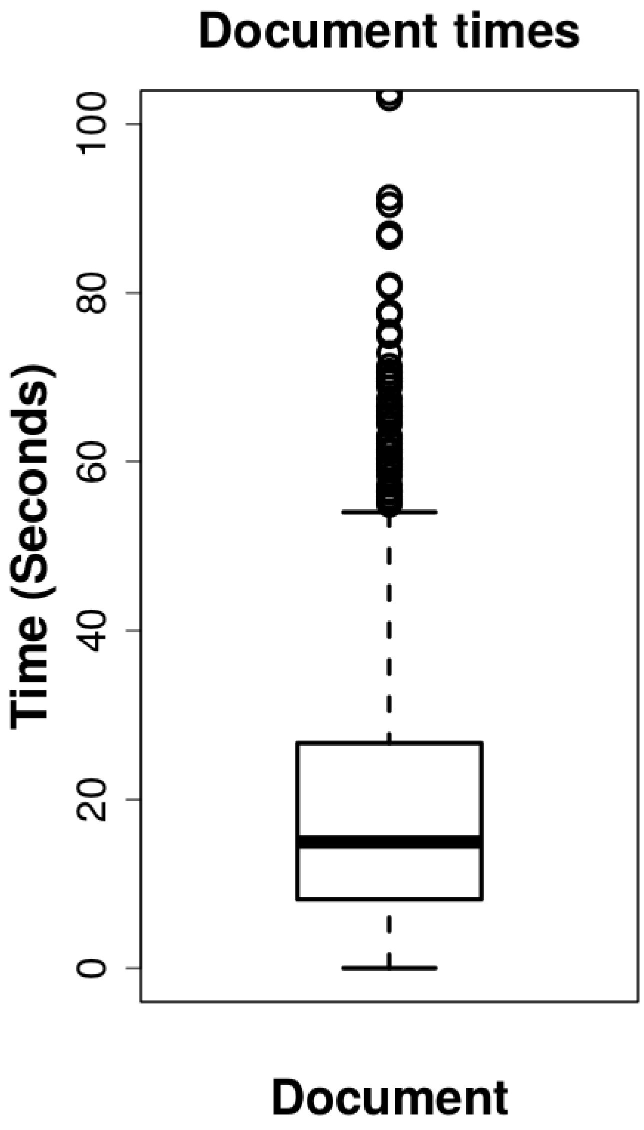




- After rank 9, it is better to reformulate query

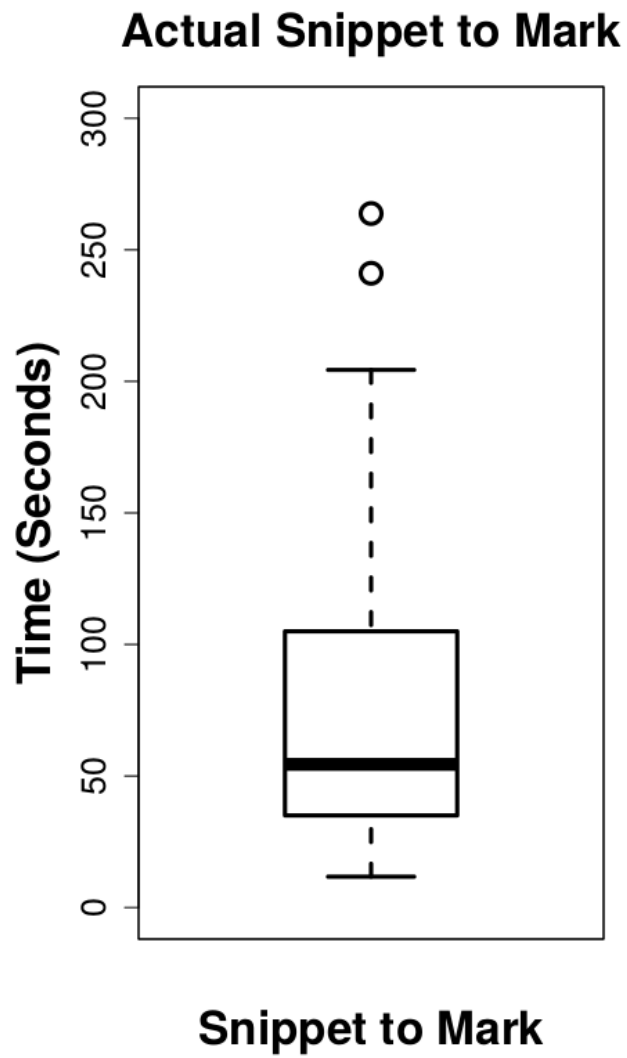
Search Time Prediction

- Analysis of 36 User sessions
- TREC tasks
- Find as many relevant items as possible
- Data from Maxwell & Azzopardi 2014
- Document times capped at 3.5 StdDev



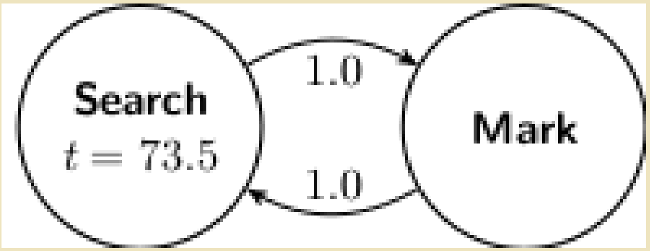
Search Time Prediction

- Regard time from first snippet (after query or mark) to mark

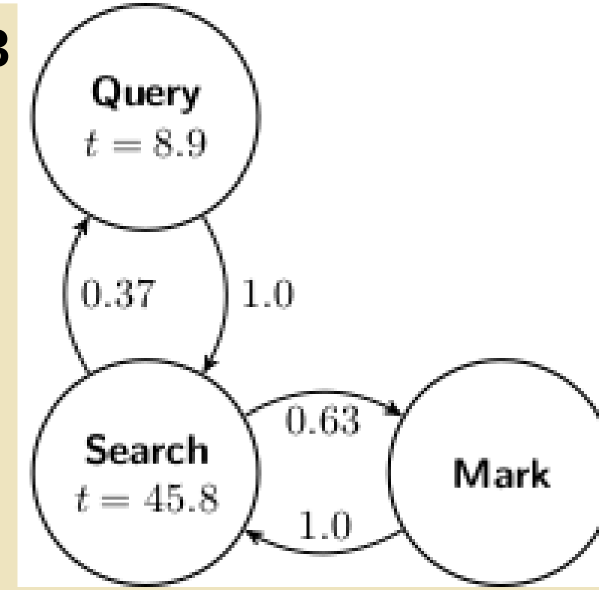


Model Variants

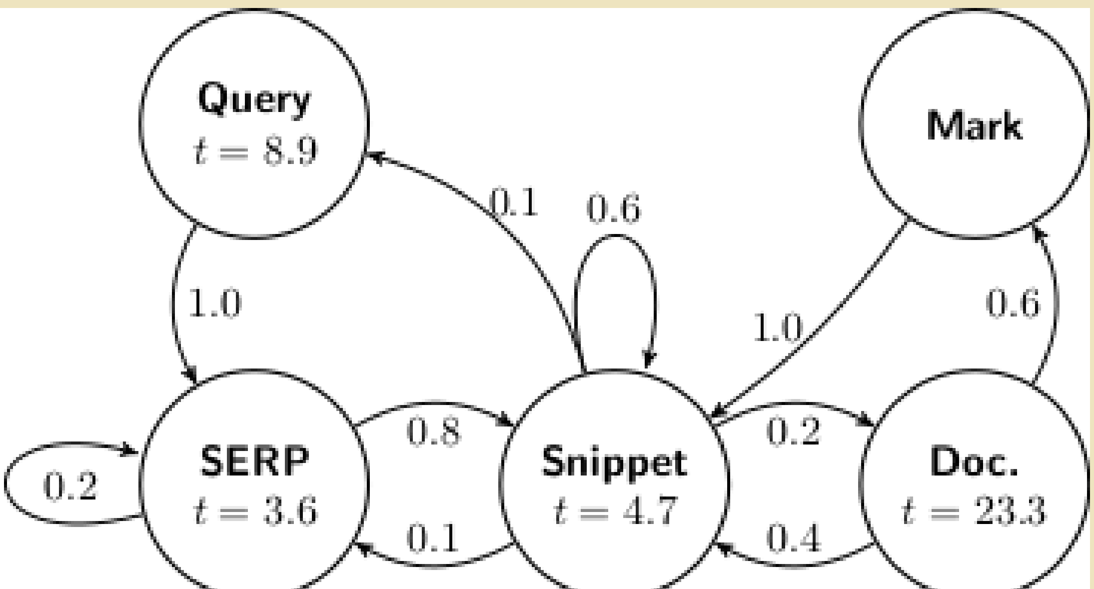
MM2



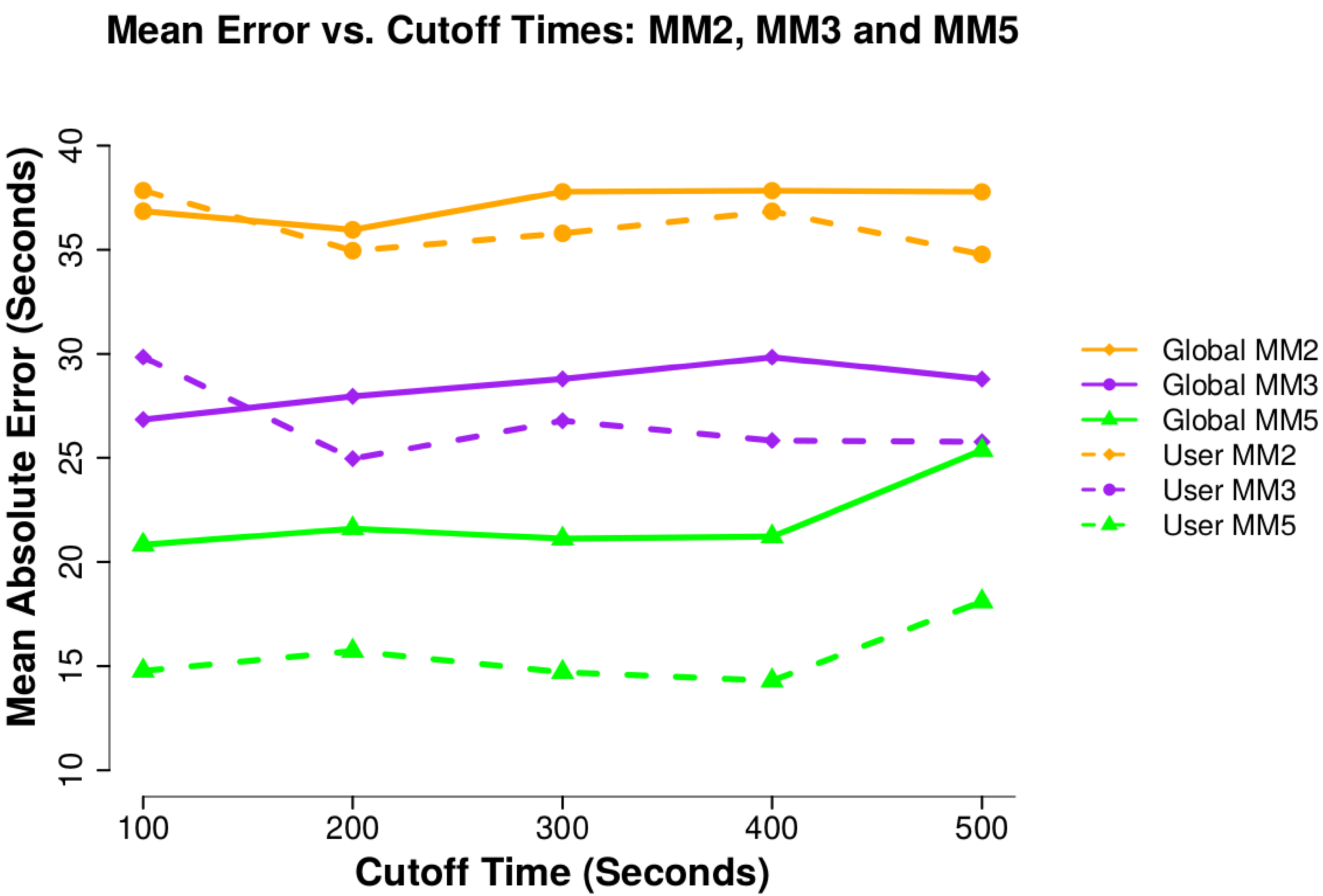
MM3



MM5



Quality of Predictions



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

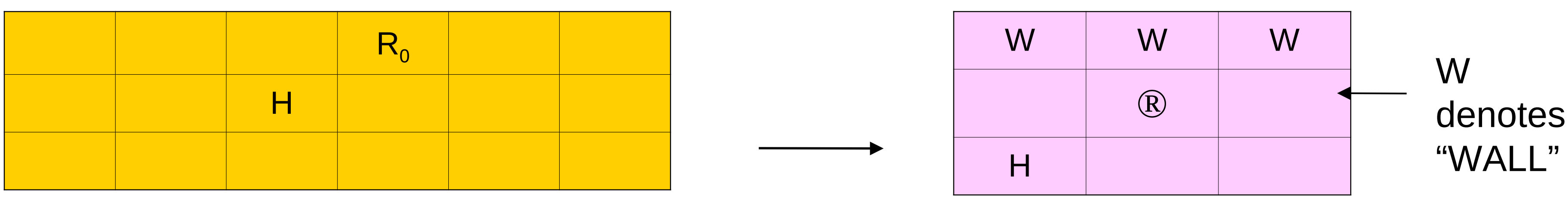
If robot has some sensors, but incomplete state information …

Hidden Markov Models are applicable!

Main Body of Lecture

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 © 2001-2003, Andrew W. Moore

Noisy Hidden State

- Example: <u>Noisy proximity sensors.</u> (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: <u>Noisy Proximity sensors.</u> (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 $\{q_{t-1}, q_{t-2}, \ldots q_1, q_0, O_{t-1}, O_{t-2}, \ldots O_1, O_0\}$ given q_t.

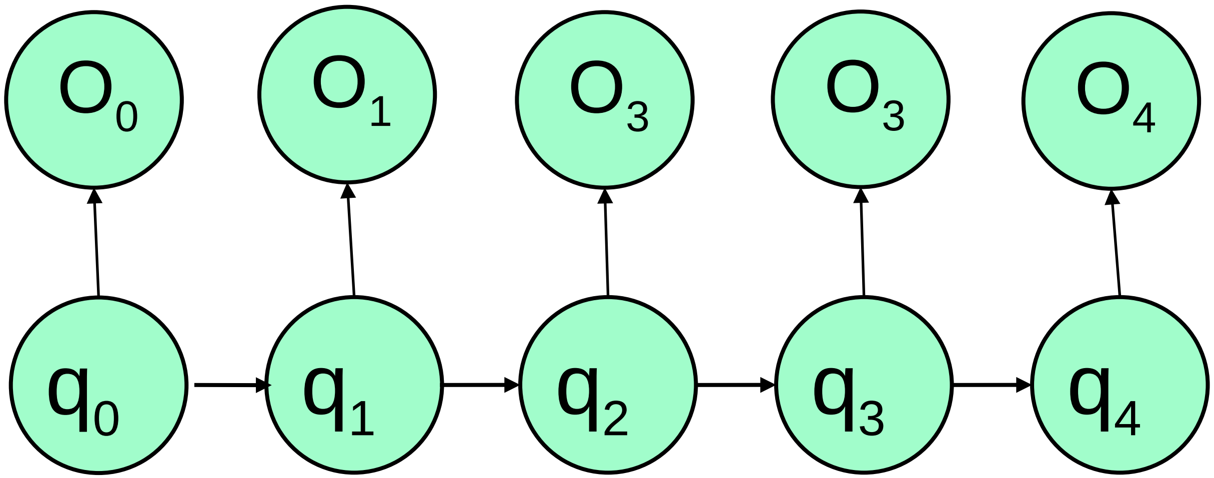
In other words:

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

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

Copyright © 2001-2003, Andrew W. Moore

Noisy Hidden State: Representation



Hidden Markov Models

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

- Question 1: State Estimation

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

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

- Question 2: Most Probable Path

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

  And what is that probability?

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

- Question 3: Learning HMMs:

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

  Very very useful. Uses the E.M. Algorithm

Copyright © 2001-2003, Andrew W. Moore

Are H.M.M.s Useful?

You bet !!

- Robot planning + sensing when there's uncertainty
- Speech Recognition/Understanding
  Phonemes $\rightarrow$ Words, Signal $\rightarrow$ phonemes
- Consumer decision modeling
- Economics & Finance.
- Many others …

Copyright © 2001-2003, Andrew W. Moore

HMM Notation (from Rabiner's Survey)

*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.

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

HMM Formal Definition

An HMM, λ, is a 5-tuple consisting of

- N the number of states
- M the number of possible observations
- $\{\pi_1, \pi_2, .. \pi_N\}$ 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

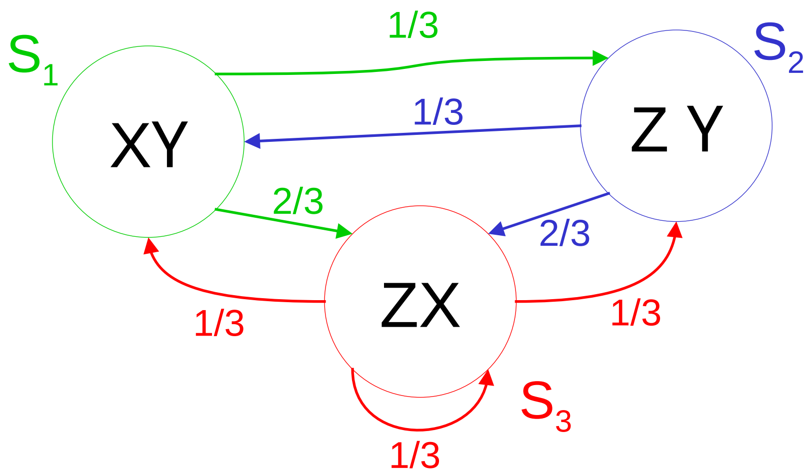
$P(O_t=k \mid q_t=S_i)=b_i(k)$

Copyright © 2001-2003, 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.



$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$

$a_{21} = 1/3$ $a_{22} = 0$ $a_{23} = 2/3$

$a_{31} = 1/3$ $a_{32} = 1/3$ $a_{33} = 1/3$

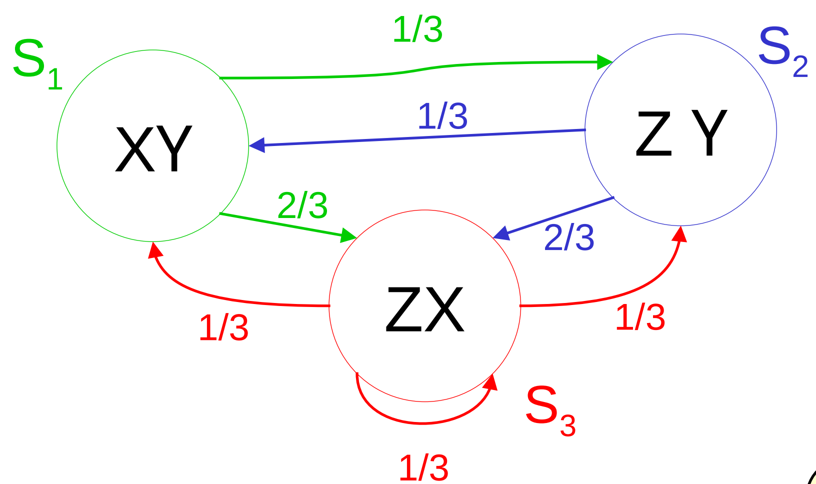
$b_1(X) = 1/2$ $b_1(Y) = 1/2$ $b_1(Z) = 0$

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

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

Copyright © 2001-2003, Andrew W. Moore

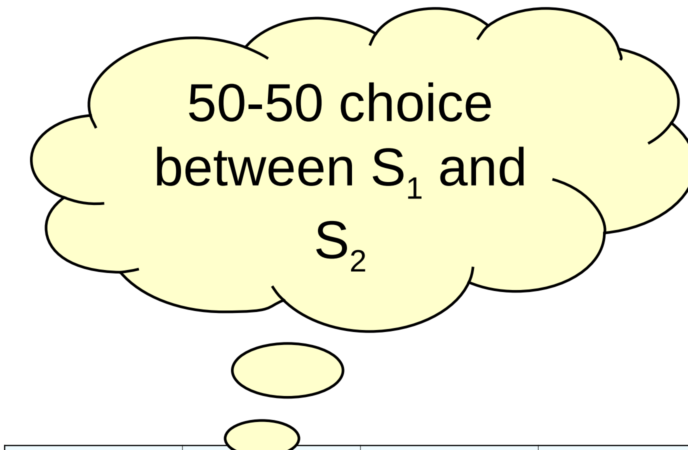
Here's an HMM



Start randomly in state 1 or 2

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

Let's generate a sequence of observations:



$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\quad \pi_2 = 1/2$ $\quad \pi_3 = 0$

$a_{11} = 0$ $\quad a_{12} = 1/3$ $\quad a_{13} = 2/3$

$a_{21} = 1/3$ $\quad a_{22} = 0$ $\quad a_{23} = 2/3$

$a_{31} = 1/3$ $\quad a_{32} = 1/3$ $\quad a_{33} = 1/3$

$b_1(X) = 1/2$ $\quad b_1(Y) = 1/2$ $\quad b_1(Z) = 0$

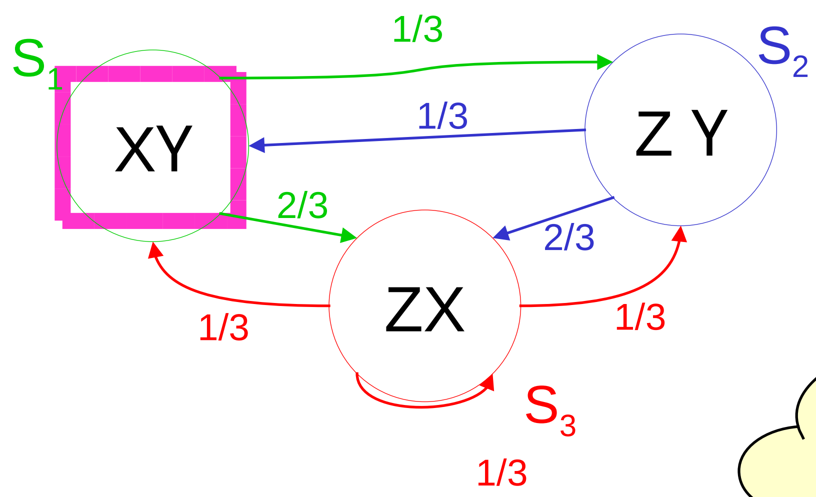
$b_2(X) = 0$ $\quad b_2(Y) = 1/2$ $\quad b_2(Z) = 1/2$

$b_3(X) = 1/2$ $\quad b_3(Y) = 0$ $\quad b_3(Z) = 1/2$

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

Copyright © 2001-2003, Andrew W. Moore

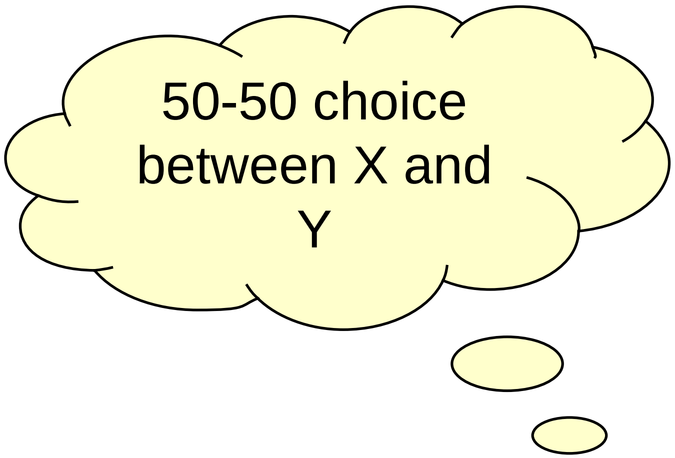
Here's an HMM



Start randomly in state 1 or 2

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

Let's generate a sequence of observations:



$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\quad\pi_2 = 1/2$ $\quad\pi_3 = 0$

$a_{11} = 0$ $\quad a_{12} = 1/3$ $\quad a_{13} = 2/3$

$a_{21} = 1/3$ $\quad a_{22} = 0$ $\quad a_{23} = 2/3$

$a_{31} = 1/3$ $\quad a_{32} = 1/3$ $\quad a_{33} = 1/3$

$b_1(X) = 1/2$ $\quad b_1(Y) = 1/2$ $\quad b_1(Z) = 0$

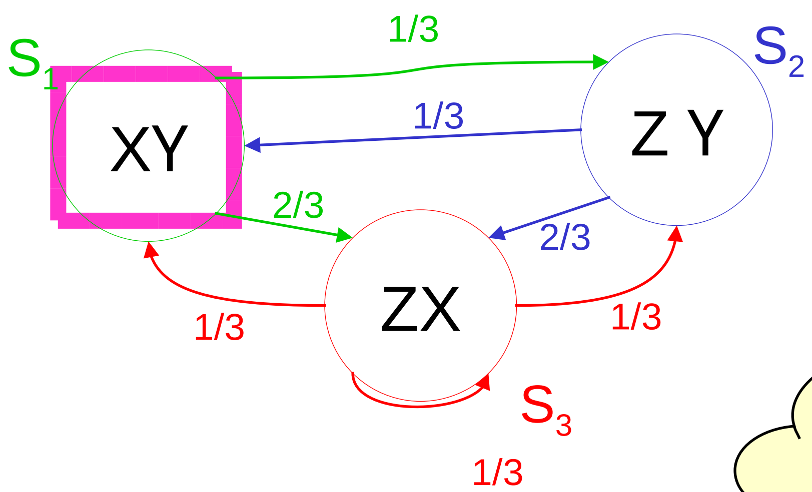
$b_2(X) = 0$ $\quad b_2(Y) = 1/2$ $\quad b_2(Z) = 1/2$

$b_3(X) = 1/2$ $\quad b_3(Y) = 0$ $\quad b_3(Z) = 1/2$

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

Copyright © 2001-2003, 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

$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$

$a_{21} = 1/3$ $a_{22} = 0$ $a_{23} = 2/3$

$a_{31} = 1/3$ $a_{32} = 1/3$ $a_{33} = 1/3$

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

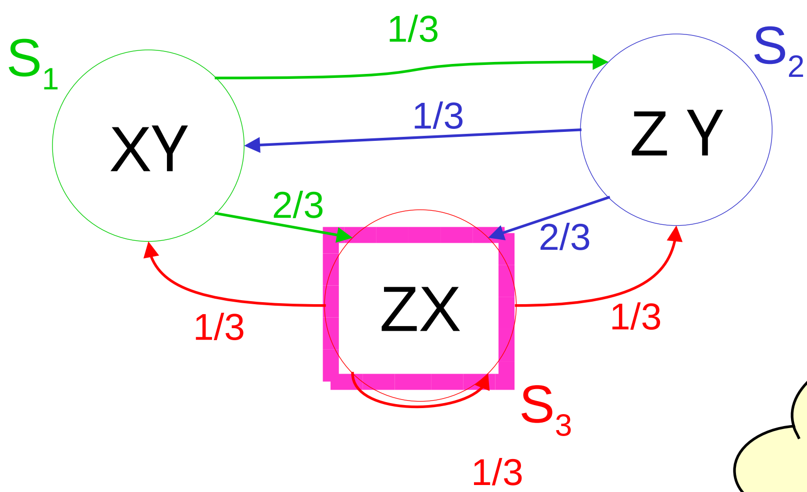
$b_2(X) = 0$ $b_2(Y) = 1/2$ $b_2(Z) = 1/2$

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

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

Copyright © 2001-2003, 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

$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\quad \pi_2 = 1/2$ $\quad \pi_3 = 0$

$a_{11} = 0$ $\quad a_{12} = 1/3$ $\quad a_{13} = 2/3$

$a_{21} = 1/3$ $\quad a_{22} = 0$ $\quad a_{23} = 2/3$

$a_{31} = 1/3$ $\quad a_{32} = 1/3$ $\quad a_{33} = 1/3$

$b_1(X) = 1/2$ $\quad b_1(Y) = 1/2$ $\quad b_1(Z) = 0$

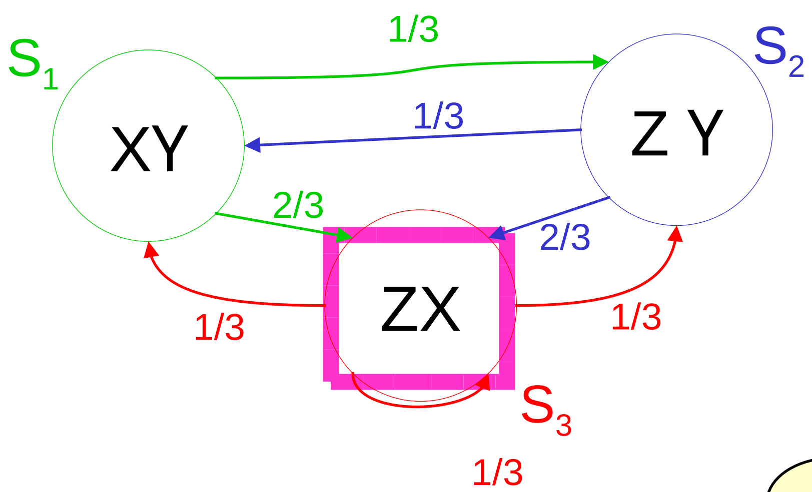
$b_2(X) = 0$ $\quad b_2(Y) = 1/2$ $\quad b_2(Z) = 1/2$

$b_3(X) = 1/2$ $\quad b_3(Y) = 0$ $\quad b_3(Z) = 1/2$

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

Copyright © 2001-2003, Andrew W. Moore

Here's an HMM



$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$

$a_{21} = 1/3$ $a_{22} = 0$ $a_{23} = 2/3$

$a_{31} = 1/3$ $a_{32} = 1/3$ $a_{33} = 1/3$

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

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

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

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:

Each of the three next states is equally likely

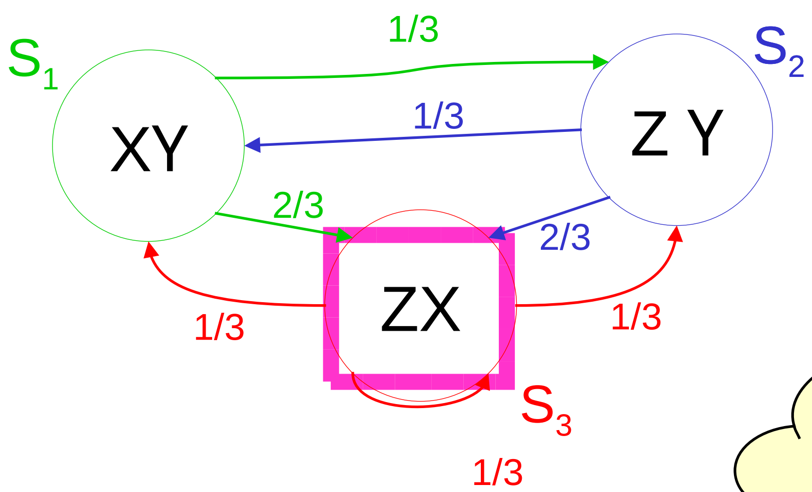
$q_0=$	S_1	$O_0=$	X
$q_1=$	S_3	$O_1=$	X
$q_2=$	__	$O_2=$	__

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:



50-50 choice between Z and X

$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$

$a_{21} = 1/3$ $a_{22} = 0$ $a_{23} = 2/3$

$a_{31} = 1/3$ $a_{32} = 1/3$ $a_{33} = 1/3$

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

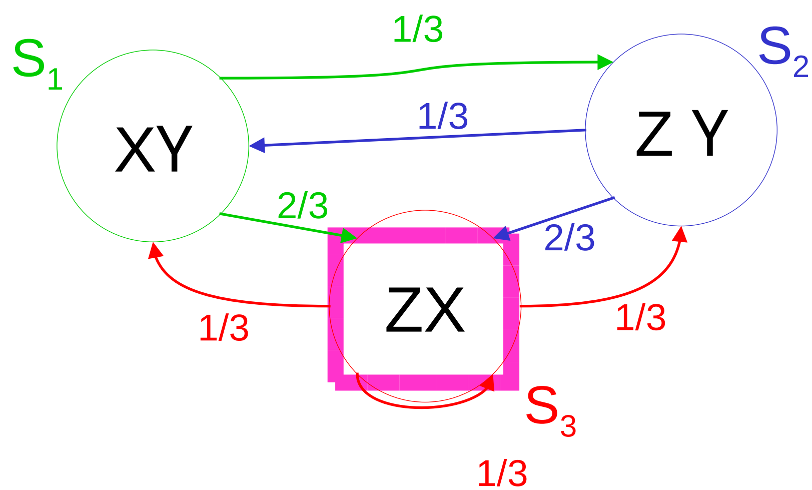
$b_2(X) = 0$ $b_2(Y) = 1/2$ $b_2(Z) = 1/2$

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

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

Copyright © 2001-2003, Andrew W. Moore

Here's an HMM



Start randomly in state 1 or 2

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

Let's generate a sequence of observations:

$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\pi_2 = 1/2$ $\pi_3 = 0$

$a_{11} = 0$ $a_{12} = 1/3$ $a_{13} = 2/3$

$a_{21} = 1/3$ $a_{22} = 0$ $a_{23} = 2/3$

$a_{31} = 1/3$ $a_{32} = 1/3$ $a_{33} = 1/3$

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

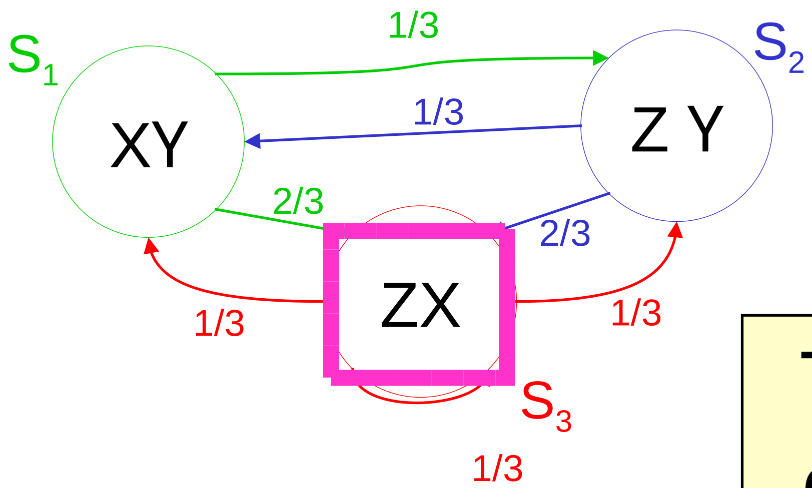
$b_2(X) = 0$ $b_2(Y) = 1/2$ $b_2(Z) = 1/2$

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

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

Copyright © 2001-2003, 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…

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

$N = 3$

$M = 3$

$\pi_1 = 1/2$ $\quad$ $\pi_2 = 1/2$ $\quad$ $\pi_3 = 0$

$a_{11} = 0$ $\quad$ $a_{12} = 1/3$ $\quad$ $a_{13} = 2/3$

$a_{21} = 1/3$ $\quad$ $a_{22} = 0$ $\quad$ $a_{23} = 2/3$

$a_{31} = 1/3$ $\quad$ $a_{32} = 1/3$ $\quad$ $a_{33} = 1/3$

$b_1(X) = 1/2$ $\quad$ $b_1(Y) = 1/2$ $\quad$ $b_1(Z) = 0$

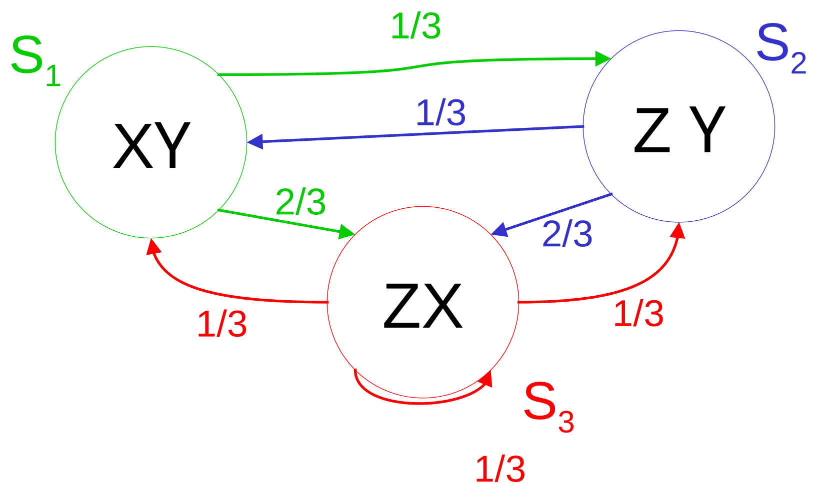
$b_2(X) = 0$ $\quad$ $b_2(Y) = 1/2$ $\quad$ $b_2(Z) = 1/2$

$b_3(X) = 1/2$ $\quad$ $b_3(Y) = 0$ $\quad$ $b_3(Z) = 1/2$

Copyright © 2001-2003, Andrew W. Moore

Prob. 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:

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

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

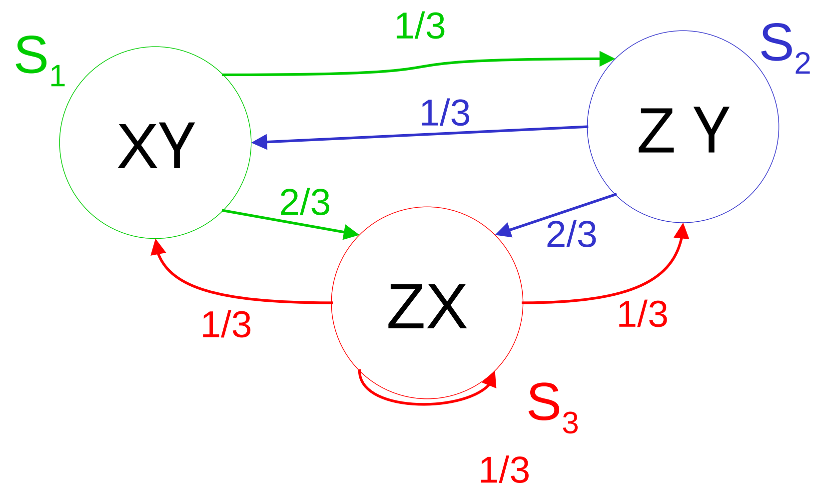
How do we compute P(Q) for an arbitrary path Q?

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

Copyright © 2001-2003, Andrew W. Moore

Prob. of a series of observations

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



Slow, stupid way:

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

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

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

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

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

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

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

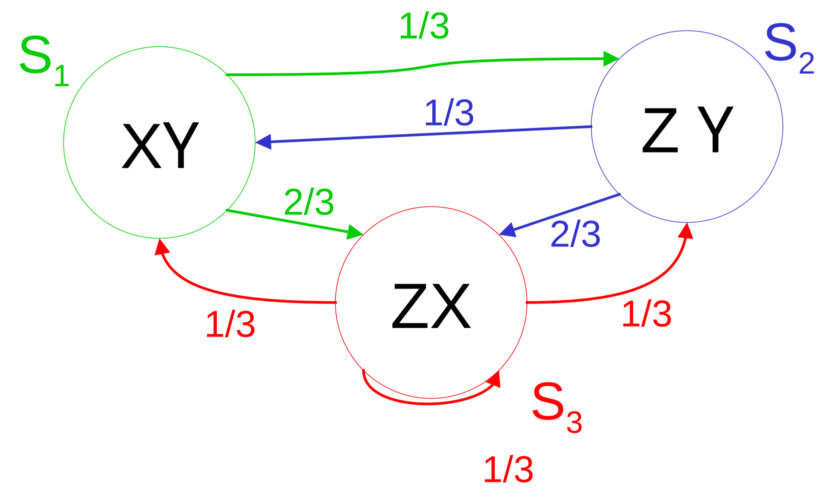
$=P(q_1)\ P(q_2|q_1)\ P(q_3|\ q_2)$ (why?)

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

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

Prob. of a series of observations

What is P(**O**) = P(O_1 O_2 O_3) =
P(O=X ∧ O_2=X ∧ O_3 = Z)?



Slow, stupid way:

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

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

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

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

P(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:

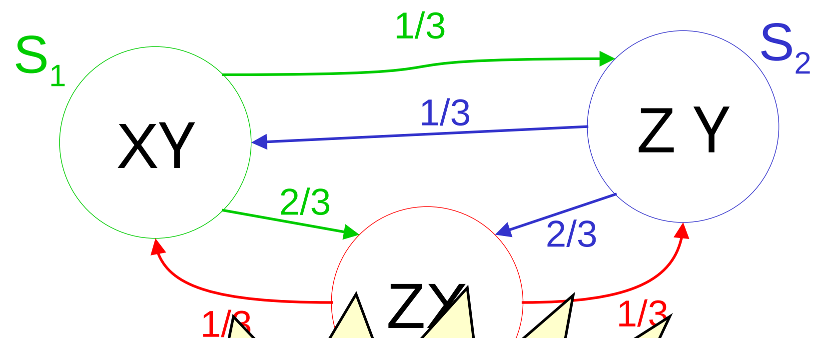
= P(X| S_1) P(X| S_3) P(Z| S_3) =

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

Copyright © 2001-2003, Andrew W. Moore

Prob. of a series of observations

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



Slow, stupid way:

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

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

How do we compute P(Q) fo
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…

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

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

Define

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

$\alpha_t(i)$ = 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 © 2001-2003, Andrew W. Moore

α_t(i): easy to define recursively

$\alpha_t(i) = P(O_1\ O_2\ \dots\ O_T\ \wedge\ q_t = S_i \mid \lambda)$

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

$$
\begin{aligned}
\alpha_1(i) &= \mathrm{P}\left(O_1 \wedge q_1 = S_i\right) \\
&= \mathrm{P}\left(q_1 = S_i\right)\mathrm{P}\left(O_1 \middle| q_1 = S_i\right) \\
&= \pi_i b_i(O_1) \\
\alpha_{t+1}(j) &= \mathrm{P}\left(O_1 O_2 \dots O_t O_{t+1} \wedge q_{t+1} = S_j\right) \\
&=
\end{aligned}
$$

Copyright © 2001-2003, Andrew W. Moore

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

$\alpha_t(i) = P(O_1\ O_2\ \ldots\ O_T\ \wedge q_t = S_i\ |\ \lambda)$

$$\alpha_1(i) = P(O_1 \wedge q_1 = S_i)$$

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

$$= \pi_i b_i(O_1)$$

$$\alpha_{t+1}(j) = P(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j)$$

$$= \sum_{i=1}^{N} P(O_1 O_2 \ldots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j)$$

Copyright © 2001-2003, Andrew W. Moore

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

$\alpha_t(i) = P(O_1\ O_2\ \ldots\ O_T\ \wedge q_t = S_i \mid \lambda)$

$$
\begin{aligned}
\alpha_1(i) &= \mathrm{P}(O_1 \wedge q_1 = S_i) \\
&= \mathrm{P}(q_1 = S_i)\,\mathrm{P}(O_1 \mid q_1 = S_i) \\
&= \pi_i b_i(O_1) \\
\alpha_{t+1}(j) &= \mathrm{P}(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j) \\
&= \sum_{i=1}^{N} \mathrm{P}(O_1 O_2 \ldots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j) \\
&= \sum_{i=1}^{N} \mathrm{P}(O_{t+1}, q_{t+1} = S_j \mid O_1 O_2 \ldots O_t \wedge q_t = S_i)\,\mathrm{P}(O_1 O_2 \ldots O_t \wedge q_t = S_i) \\
&= \sum_{i} \mathrm{P}(O_{t+1}, q_{t+1} = S_j \mid q_t = S_i)\,\alpha_t(i)
\end{aligned}
$$

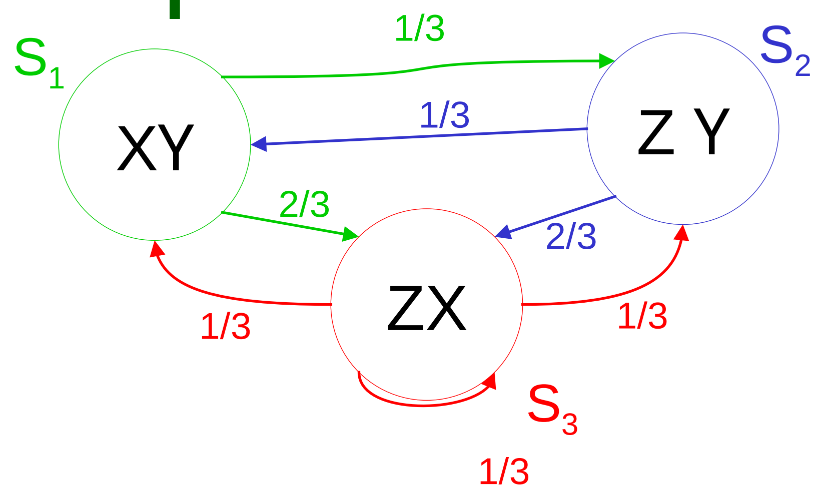
Copyright © 2001-2003, Andrew W. Moore

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

$\alpha_t(i) = P(O_1\ O_2\ \ldots\ O_T\ \wedge q_t = S_i \mid \lambda)$

$$\alpha_1(i) = \mathrm{P}(O_1 \wedge q_1 = S_i)$$
$$= \mathrm{P}(q_1 = S_i)\,\mathrm{P}(O_1 | q_1 = S_i)$$
$$= \pi_i b_i(O_1)$$
$$\alpha_{t+1}(j) = \mathrm{P}(O_1 O_2 \ldots O_t O_{t+1} \wedge q_{t+1} = S_j)$$
$$= \sum_{i=1}^{N} \mathrm{P}(O_1 O_2 \ldots O_t \wedge q_t = S_i \wedge O_{t+1} \wedge q_{t+1} = S_j)$$
$$= \sum_{i=1}^{N} \mathrm{P}(O_{t+1}, q_{t+1} = S_j | O_1 O_2 \ldots O_t \wedge q_t = S_i)\,\mathrm{P}(O_1 O_2 \ldots O_t \wedge q_t = S_i)$$
$$= \sum_{i} \mathrm{P}(O_{t+1}, q_{t+1} = S_j | q_t = S_i)\alpha_t(i)$$
$$= \sum_{i} \mathrm{P}(q_{t+1} = S_j | q_t = S_i)\,\mathrm{P}(O_{t+1} | q_{t+1} = S_j)\alpha_t(i)$$
$$= \sum_{i} a_{ij} b_j(O_{t+1})\alpha_t(i)$$

in our example



$$\alpha_t(i) = P(O_1 O_2 \ldots O_t \wedge q_t = S_i \mid \lambda)$$

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

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

WE SAW O_1 O_2 O_3 = X X Z

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

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

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

Copyright © 2001-2003, 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

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

(How) can we cheaply compute

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

Copyright © 2001-2003, 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

$$P(O_1O_2\ldots O_t) \quad ? \qquad \sum_{i=1}^{N}\alpha_t(i)$$

(How) can we cheaply compute

$$P(q_t=S_i|O_1O_2\ldots O_t) \qquad \frac{\alpha_t(i)}{\sum_{j=1}^{N}\alpha_t(j)}$$

Copyright © 2001-2003, Andrew W. Moore

Most probable path given observations

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

What is $\underset{Q}{\operatorname{argmax}} P\left(Q \mid O_1 O_2 \ldots O_T\right)$?

Slow, stupid answer:

$$\underset{Q}{\operatorname{argmax}} P\left(Q \mid O_1 O_2 \ldots O_T\right)$$

$$= \underset{Q}{\operatorname{argmax}} \frac{P\left(O_1 O_2 \ldots O_T \mid Q\right) P(Q)}{P\left(O_1 O_2 \ldots O_T\right)}$$

$$= \underset{Q}{\operatorname{argmax}} P\left(O_1 O_2 \ldots O_T \mid Q\right) P(Q)$$

Copyright © 2001-2003, 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:

...OCCURING

and

...ENDING UP IN STATE S_i

and

...PRODUCING OUTPUT $O_1 \ldots O_t$

DEFINE: $mpp_t(i)$ = that path

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

The Viterbi Algorithm

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

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

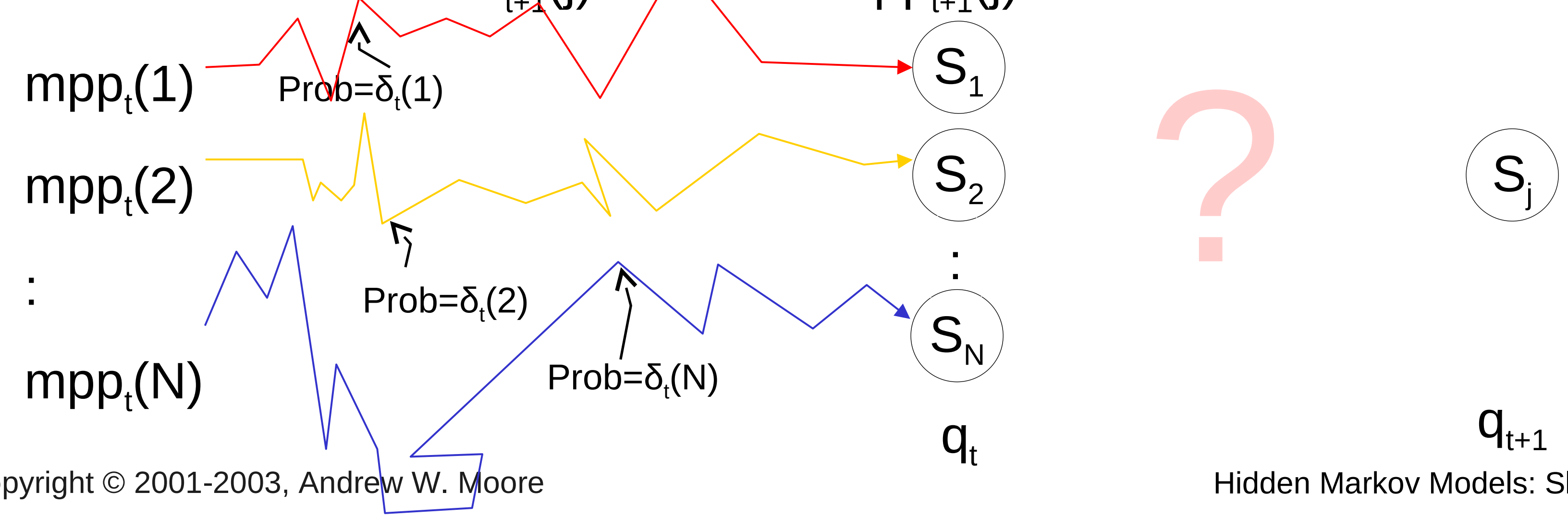
$$\delta_1(i) = \max \mathrm{P}(q_1 = S_i \wedge O_1)$$

$$= \mathrm{P}(q_1 = S_i)\, \mathrm{P}(O_1 | q_1 = S_i)$$

$$= \pi_i b_i(O_1)$$

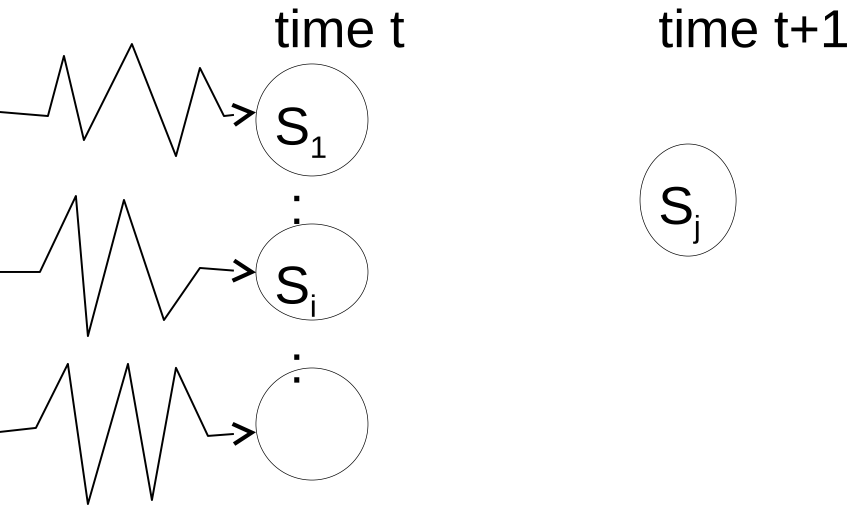
Now, suppose we have all the $\delta_t(i)$'s and $mpp_t(i)$'s for all i.

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



Copyright © 2001-2003, Andrew W. Moore

The Viterbi Algorithm

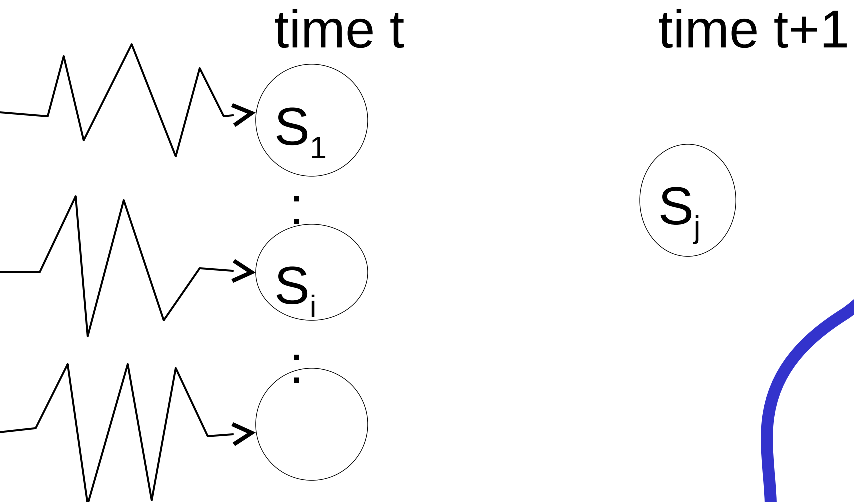


The most prob. path with last two states S_i S_j

is

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

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?

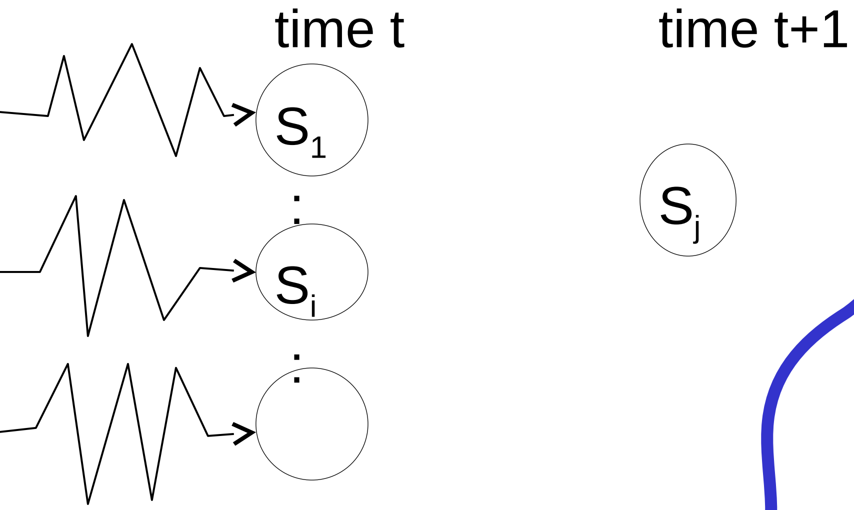
$$\delta_t(i) \times P(S_i \rightarrow S_j \wedge O_{t+1} \mid \lambda)$$

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

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



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?

$$\delta_t(i) \times P(S_i \rightarrow S_j \wedge O \ldots | \lambda)$$

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

SO The most probable ...

S_{i*} as its penultimate state

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

Summary:

$\delta_{t+1}(j) = \delta_t(i^*)\, a_{ij}\, b_j\, (O_{t+1})$
$mpp_{t+1}(j) = mpp_{t+1}(i^*) S_{i^*}$
} 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

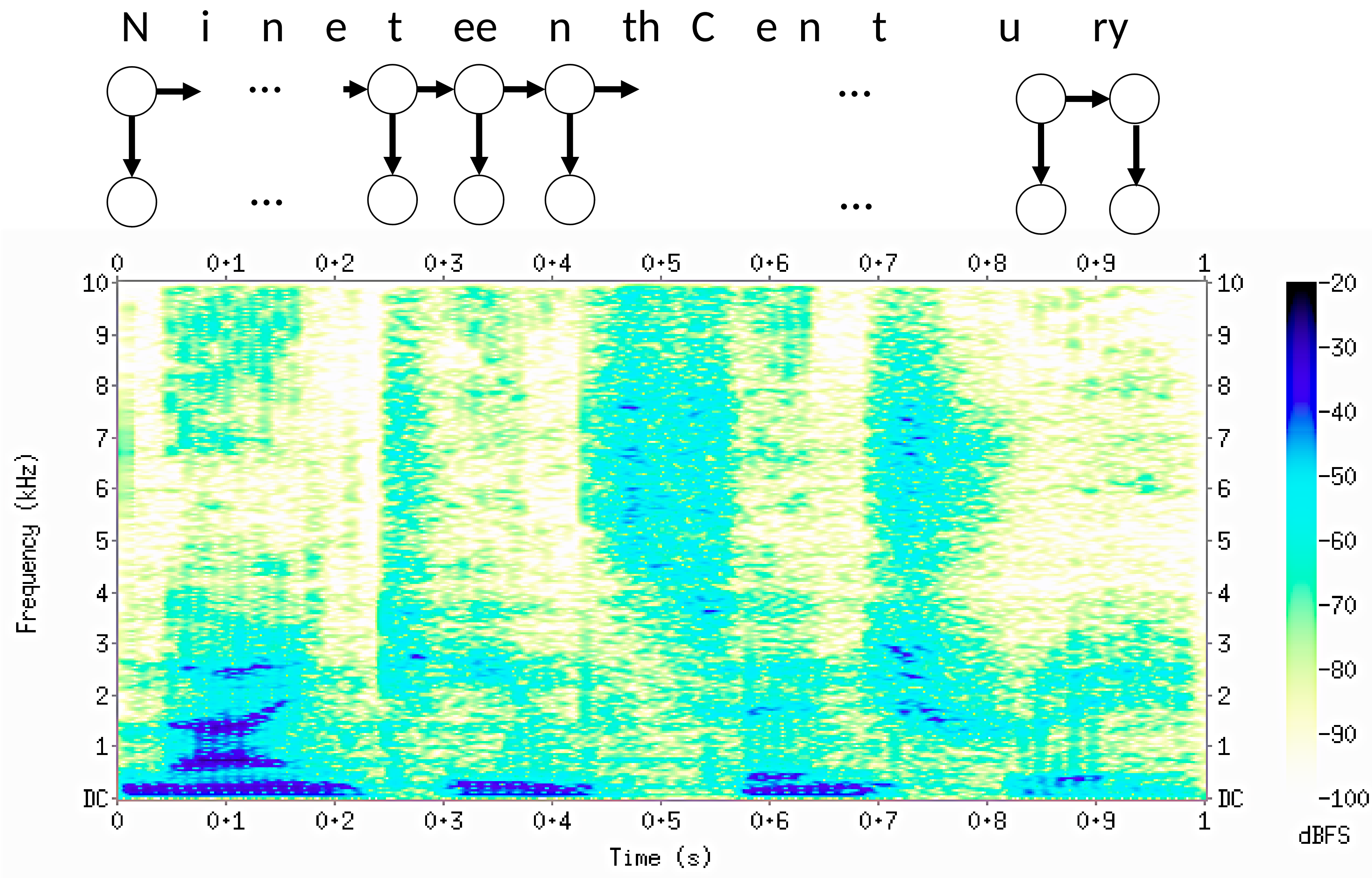
What is the most probable word given this signal?

UTTERLY GROSS SIMPLIFICATION

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

Copyright © 2001-2003, Andrew W. Moore

Example: HMMs for speech recognition



Spectrogram for audio for “nineteenth century” – From Wikipedia

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**T**

Copyright © 2001-2003, Andrew W. Moore

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.

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

AS USUAL: We could use

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

BAYES

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

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

Copyright © 2001-2003, Andrew W. Moore

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

Copyright © 2001-2003, Andrew W. Moore

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

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

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

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

HMM estimation

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

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

We can re-estimate

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

We can also re-estimate

$$b_j(O_k) \leftarrow L$$ (see Rabiner)

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…

Copyright © 2001-2003, Andrew W. Moore

EM 4 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
   $\gamma_t(i)$, $\varepsilon_t(i,j)$ $\quad \forall 1 \le t \le T, \quad \forall 1 \le i \le N, \quad \forall 1 \le j \le N$
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.**

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

Copyright © 2001-2003, Andrew W. Moore

Bad News

- There are lots o

- The local minim
  data.

Trade-off between too few states (inadequately modeling the structure in the data) and too many (fitting the noise).

Thus #states is a regularization parameter.

Blah blah blah… bias variance tradeoff…blah blah…cross-validation…blah blah….AIC, BIC….blah blah (same ol' same ol')

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

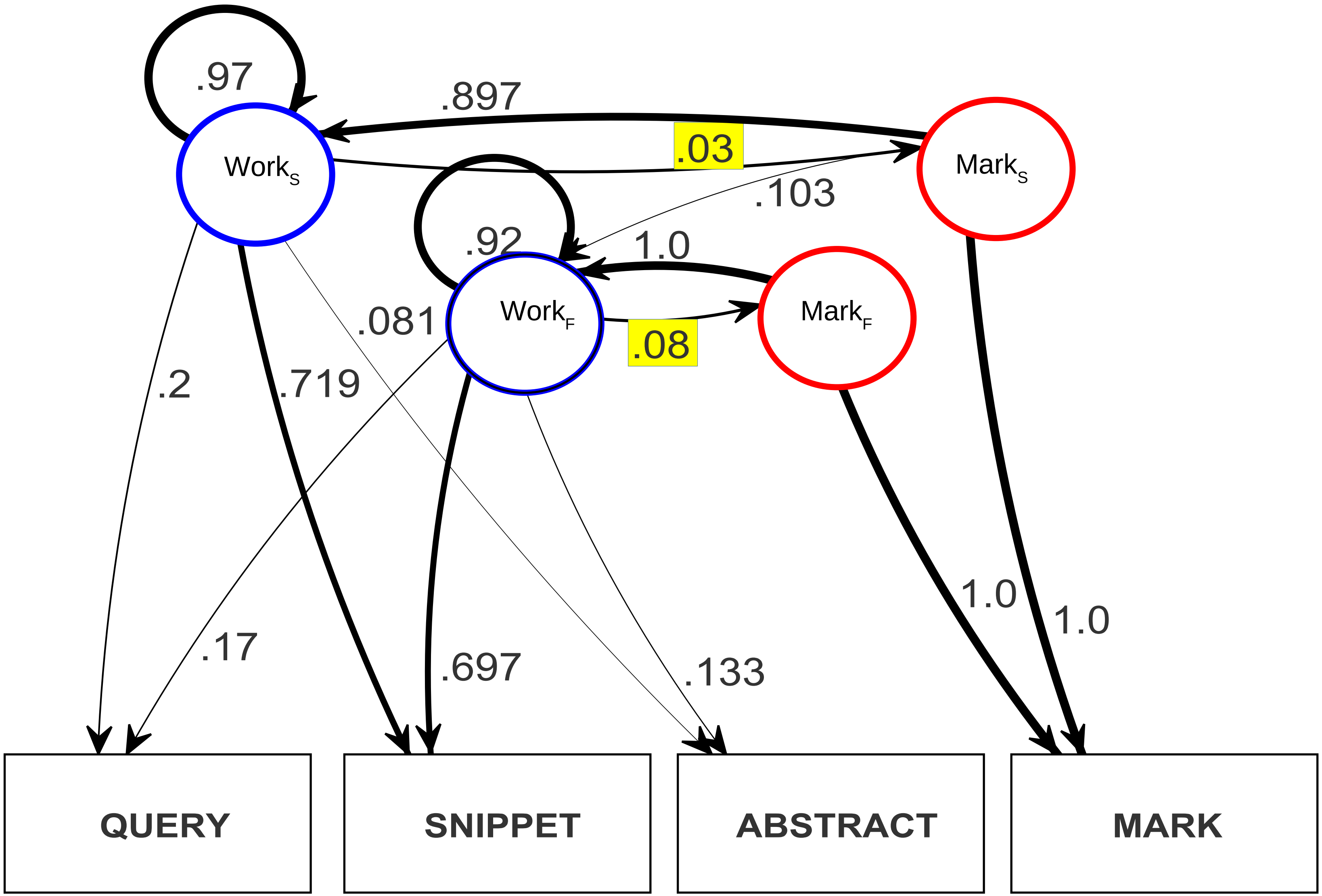
- What is a Markov model, what is an HMM?
- Which questions can be answered with a MM?
- What are the central questions for an HMM?
- What are the basic methods for addressing these problems?
- Give some examples of applications of MMs and HMMs!

Copyright © 2001-2003, Andrew W. Moore

Hidden Markov Models (HMM)

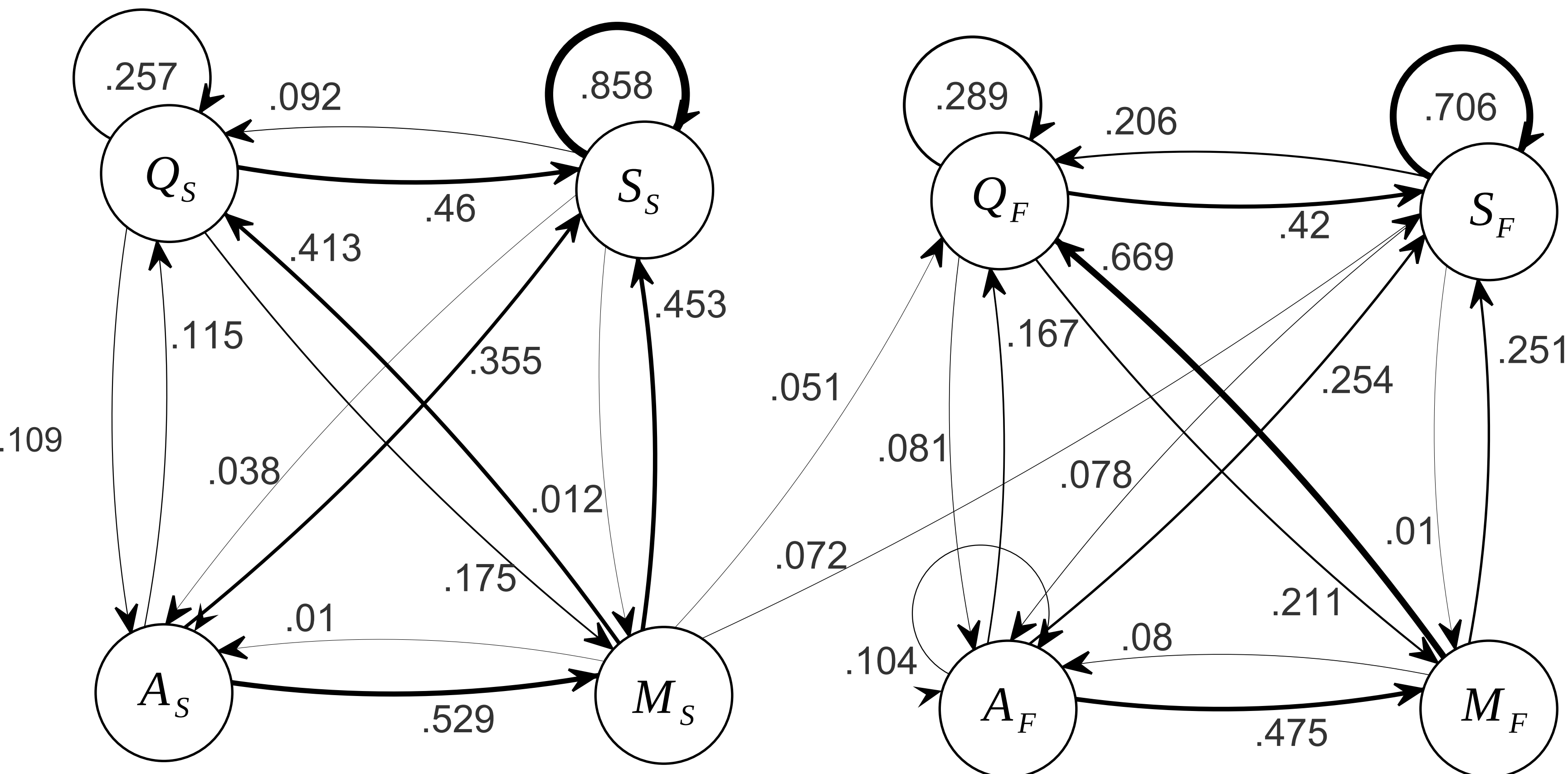
- Goal: detection of search phases
  - adaptive system behavior: ranking, search suggestions...
  - user guidance
- Data
  - Search logs from German Social Science Info Centre (GESIS)
  - Consider only sessions with > 3 documents marked relevant
  - 1642 search sessions

Simple Discrete HMM



Hybrid HMM

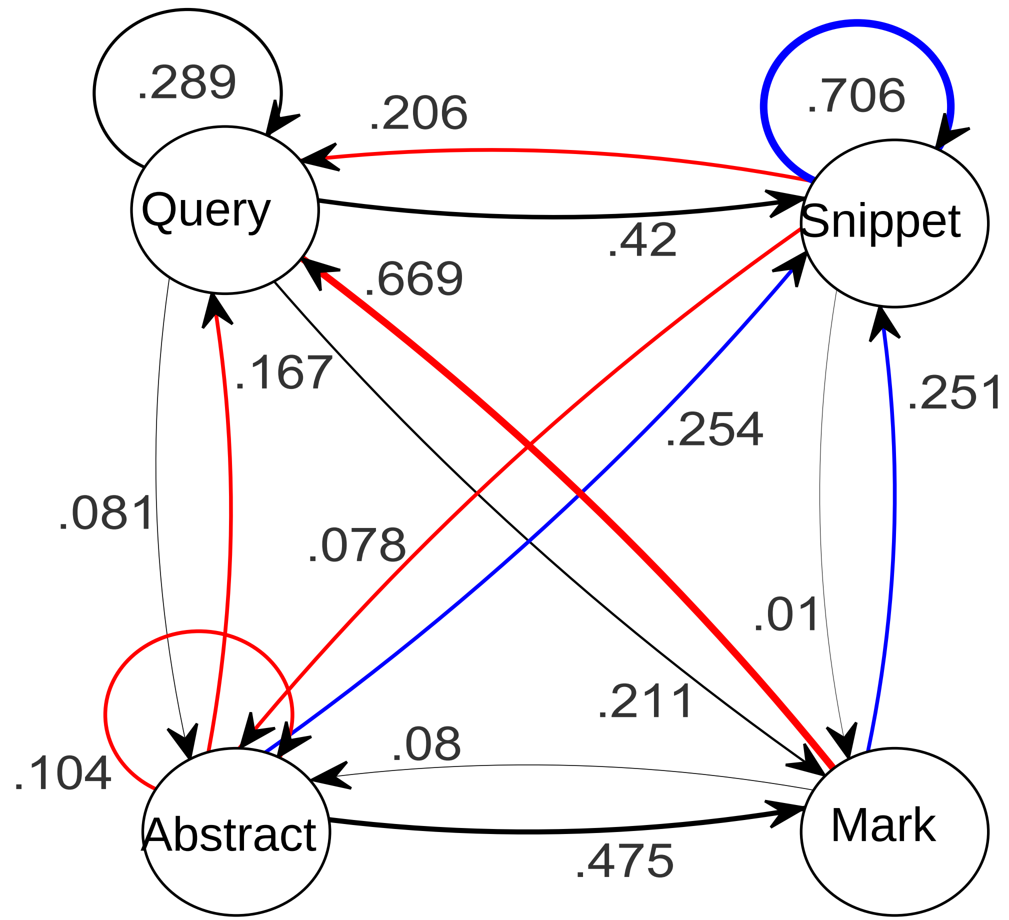
- 4 states per search phase
- Consider discrete signals + action times



Differences Between Phases

In 2nd phase, users

- formulate more queries
- look at fewer snippets
- click more often on a snippet



(H)MMs for Classification

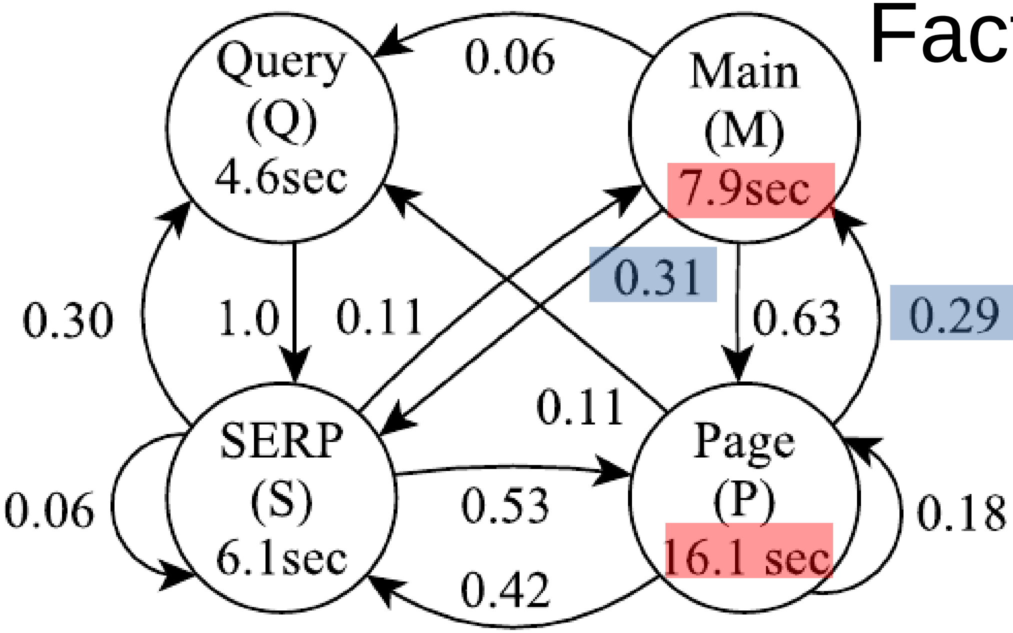
- Given: training sample with observation sequences for different classes c_k
- Train separate HMM λ_k for each class c_k
- Testing: determine most probable class C for given observation sequence ε_i

$$C(\varepsilon_i) = \underset{c_k}{\operatorname{argmax}}\, P(\varepsilon_i \mid \lambda_k)$$

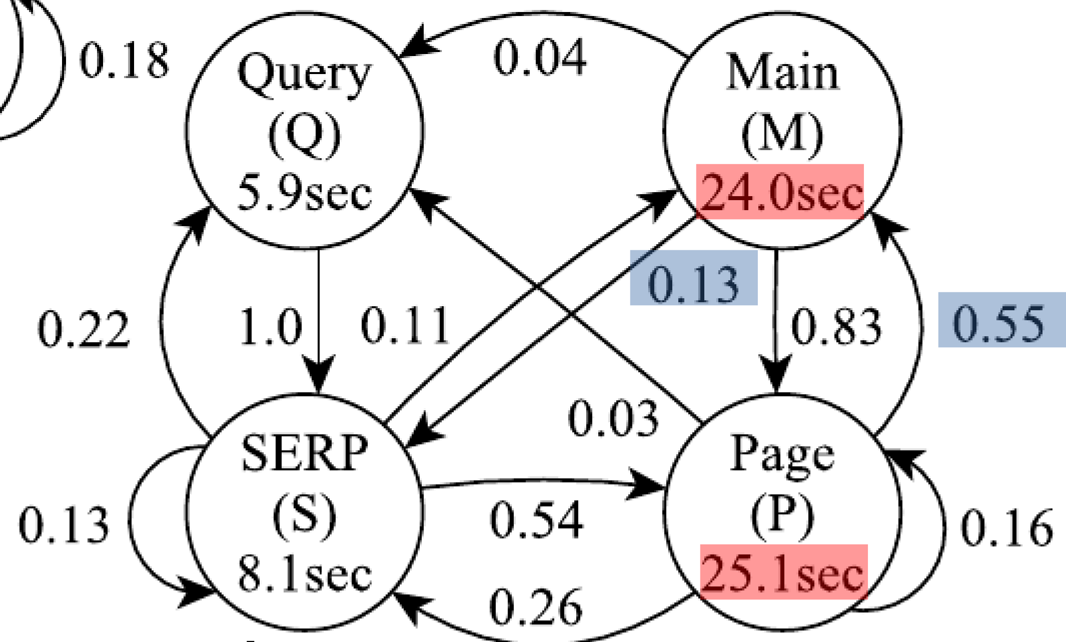
Recognizing Type of Search

UNIVERSITÄT
DUISBURG
ESSEN
Open-Minded

Fact Finding



(Main contains the task description)



Exploratory Search