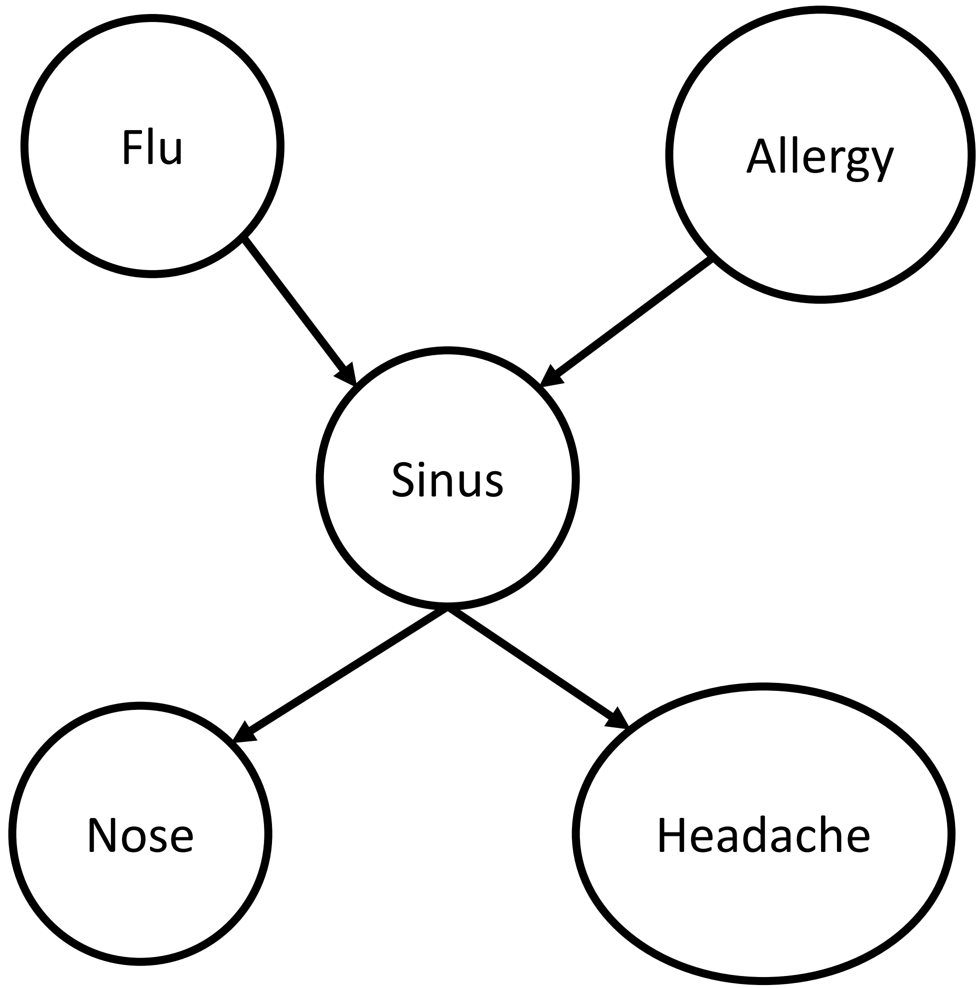
Artificial Intelligence

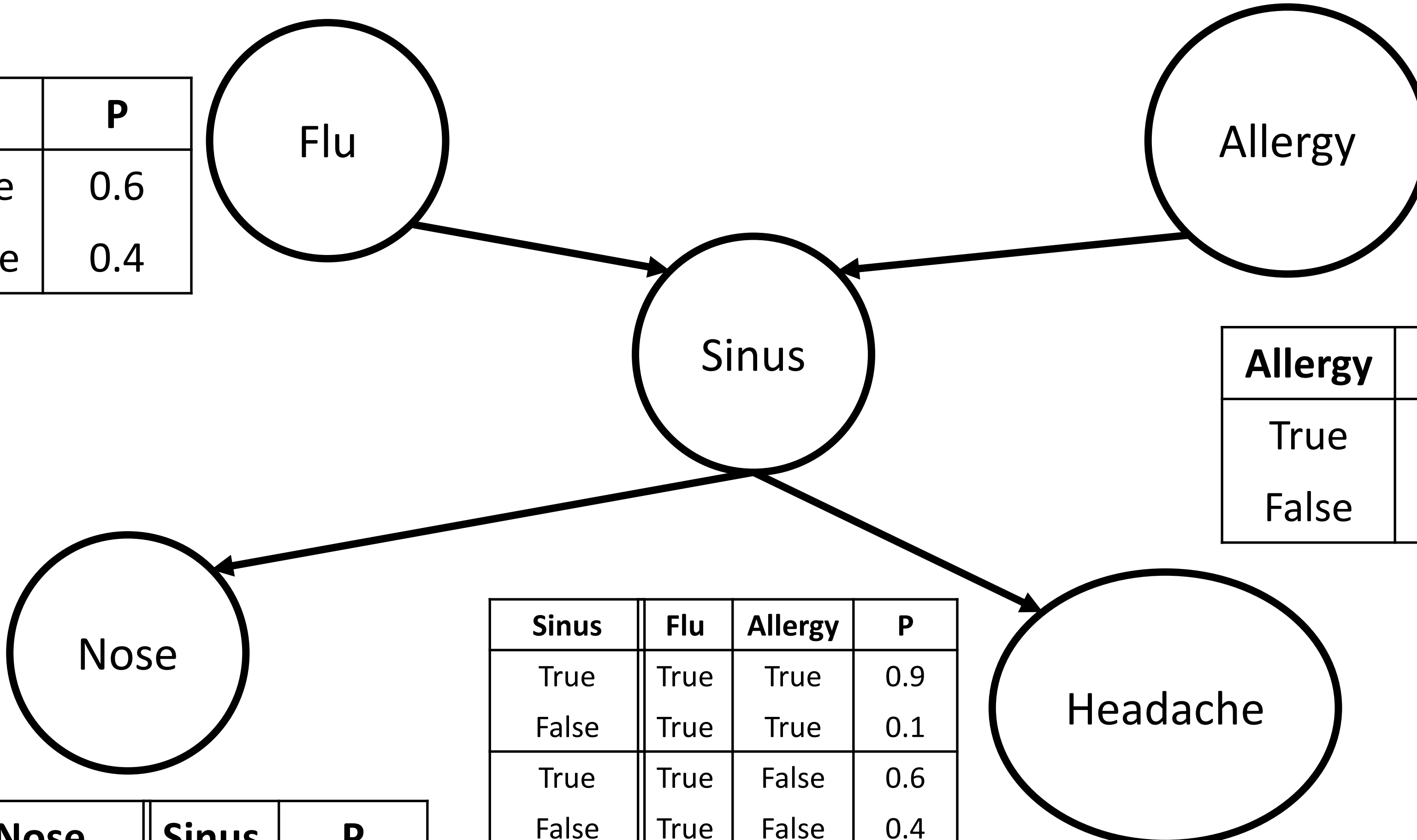
Steve James

Hidden Markov Models

Recall



Recall: BN



Flu	P
True	0.6
False	0.4

Allergy	P
True	0.2
False	0.8

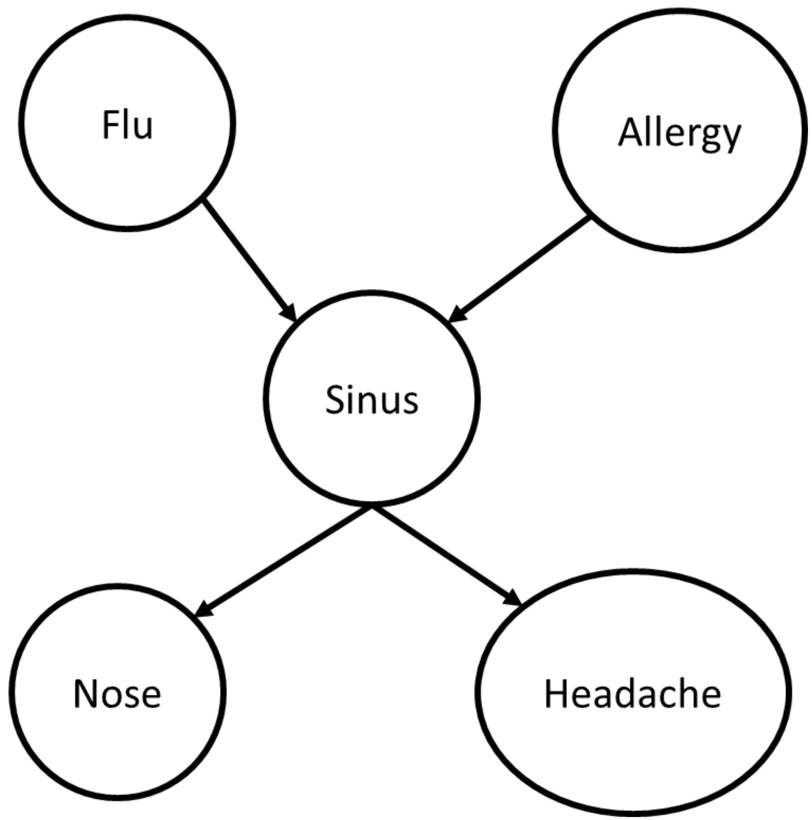
Sinus	Flu	Allergy	P
True	True	True	0.9
False	True	True	0.1
True	True	False	0.6
False	True	False	0.4
True	False	True	0.2
False	False	True	0.8
True	False	False	0.4
False	False	False	0.6

Nose	Sinus	P
True	True	0.8
False	True	0.2
True	False	0.3
False	False	0.7

Headache	Sinus	P
True	True	0.6
False	True	0.4
True	False	0.5
False	False	0.5

Inference

- What is $P(flu = True|headache = True)$?



Time

- Bayesian networks (so far) contain no notion of time

- In many applications, how signal changes over time is critical:
  - Target tracking
  - Patient monitoring
  - Speech recognition
  - Gesture recognition

State

- In prob theory, we talked about atomic events
  - All possible outcomes
  - Mutually exclusive

- In time series, we have state:
  - System is in a state at time t
  - Describes system completely
  - Over time, transition from state to state

Example

- Weather today can be
  – Hot
  – Cold
  – Chilly
  – Freezing
- Weather has four states
- At each point in time, system is in one (and only one) state

Example

t=1	**t=2**	**t=3**	**…**	**t=n**
Freezing	**Freezing**	Freezing		Freezing
Chilly	Chilly	**Chilly**		Chilly
Hot	Hot	Hot		**Hot**

State at time t

Markov Assumption

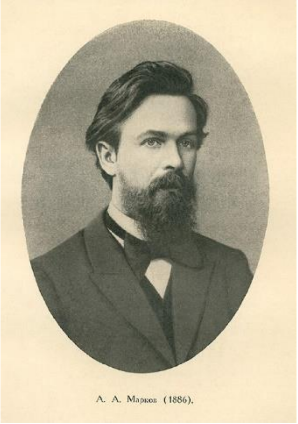
- We are probabilistic modelers, so we'd like to model:

$$P(S_t | S_{t-1}, S_{t-2}, \ldots, S_0)$$

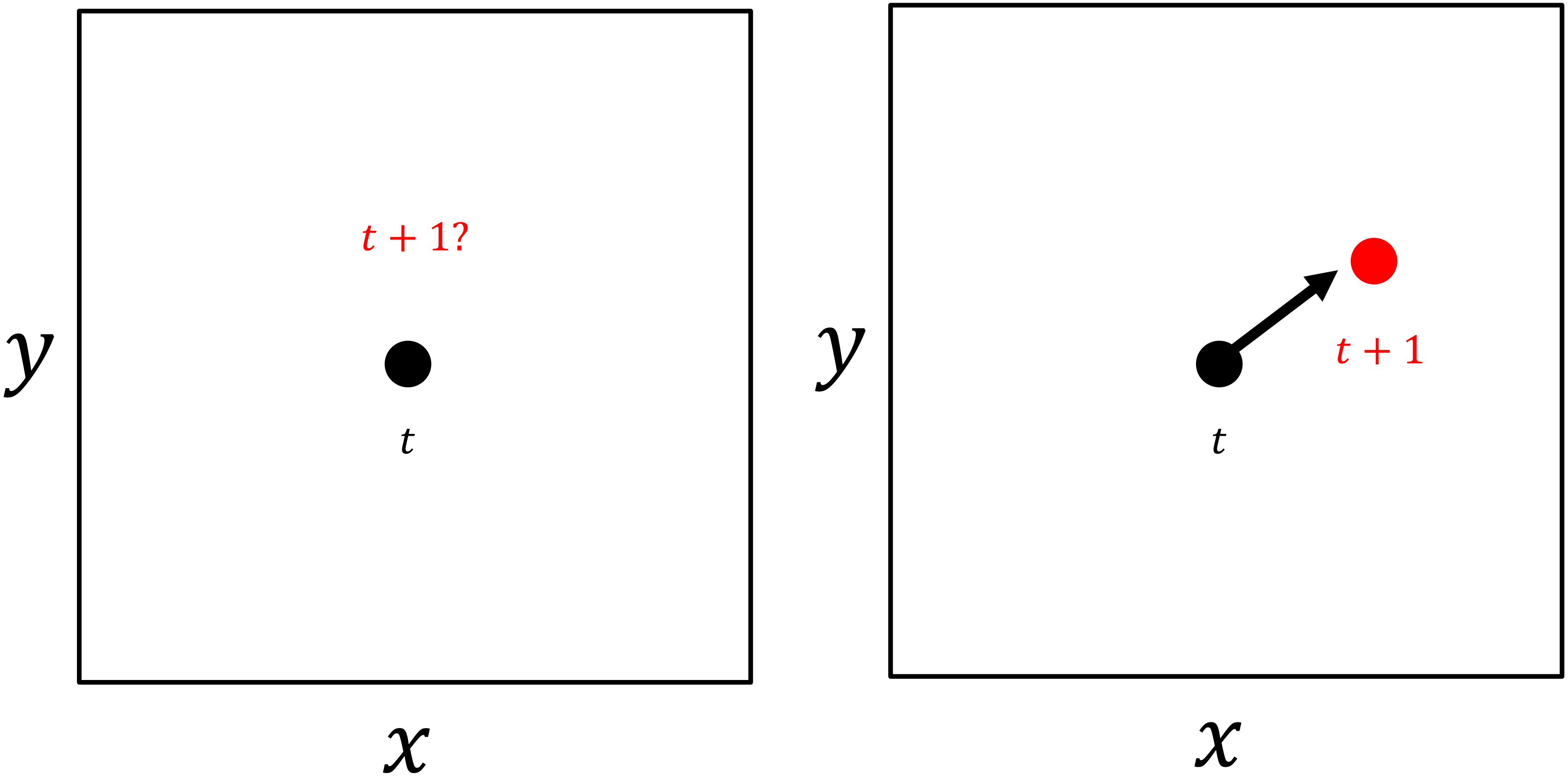
- A state has the Markov property when we can write this as:

$$P(S_t | S_{t-1})$$

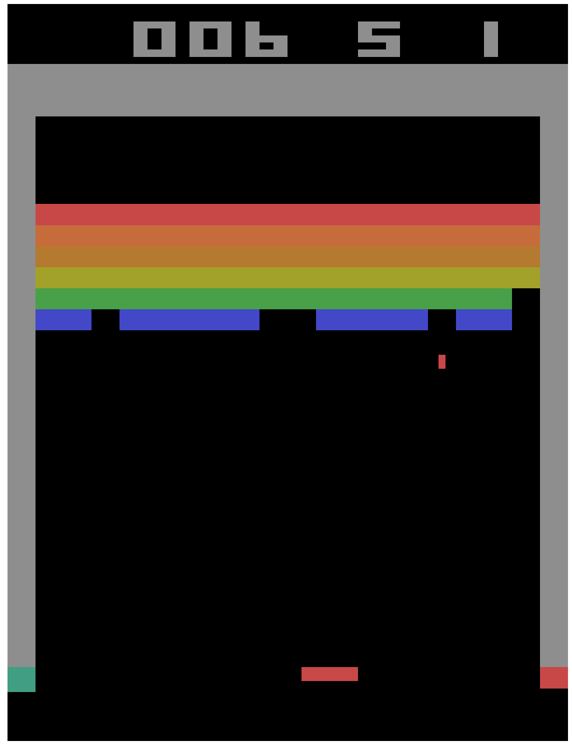
- Special kind of independence assumption:
  - *Future independent of past given present*

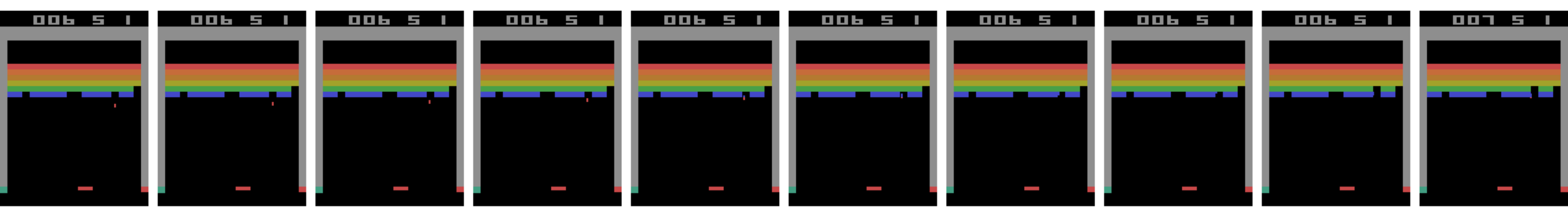


Example



Memorising



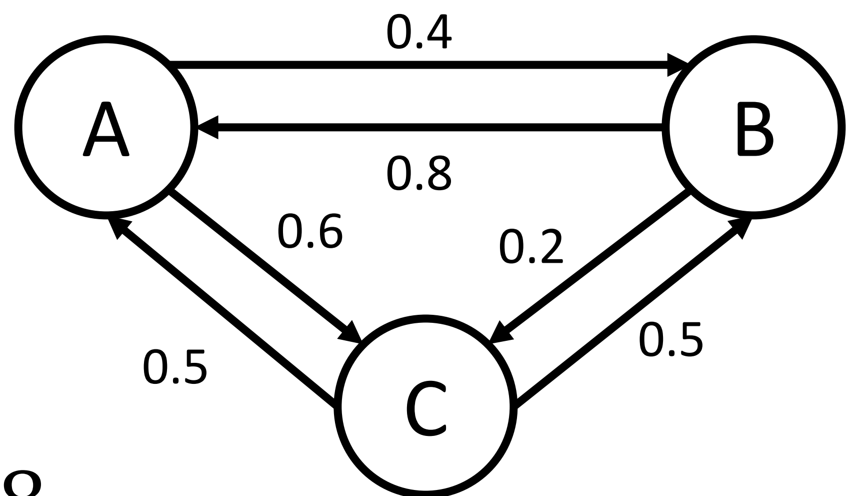


Markov Assumption

- Model that has this is the Markov model
- Sequence of states thus generated is a Markov chain

- Definition of a state:
  - Sufficient statistic for history
  - $P(S_t|S_{t-1},\ldots,S_0) = P(S_t|S_{t-1})$
- Can describe transition probabilities with matrix
  - $P(S_i|S_j)$
  - Steady state probabilities
  - Convergence rates

State machines



$$P(A|B) = 0.8$$
$$P(A|C) = 0.5$$
$$P(B|A) = 0.4$$
$$P(B|C) = 0.5$$
$$P(C|A) = 0.6$$
$$P(C|B) = 0.2$$

Time implicit

	A	**B**	**C**
A	0.0	0.8	0.5
B	0.4	0.0	0.5
C	0.6	0.2	0.0

State machines

- Assumptions
  - Markov assumption
  - Transition probabilities don’t change with time
  - Event space doesn’t change with time
  - Time moves in discrete increments

Hidden state

- State machines are great, but:
  - Often state is not observed directly
  - State is latent or hidden

State: forehand

State:$[playerx, playery, enemyx, enemyy]$

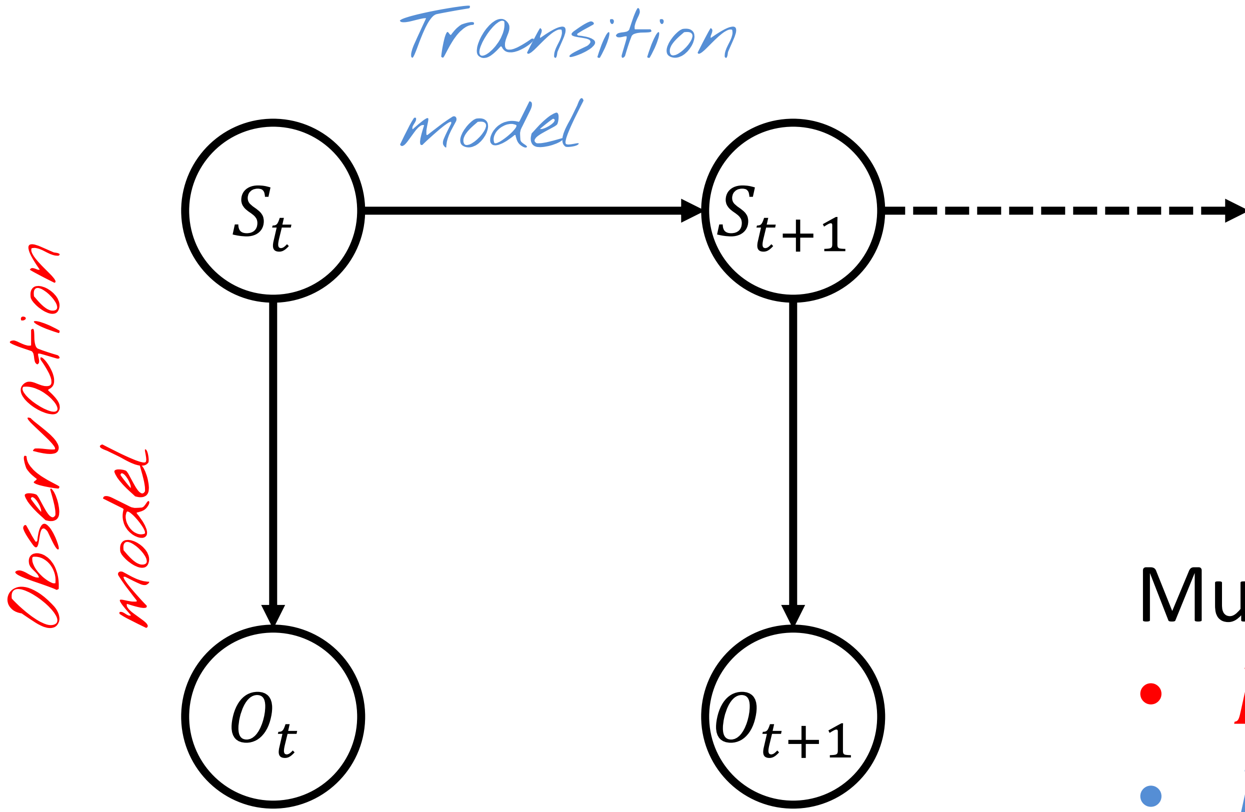
- Instead you see an observation
  - This contains info about hidden state

Examples

State	**Observation**
Word	Phoneme
Chemical state	Colour, smell, etc
Flu?	Runny nose
Cardiac arrest?	Pulse

Sensors!

Hidden Markov models



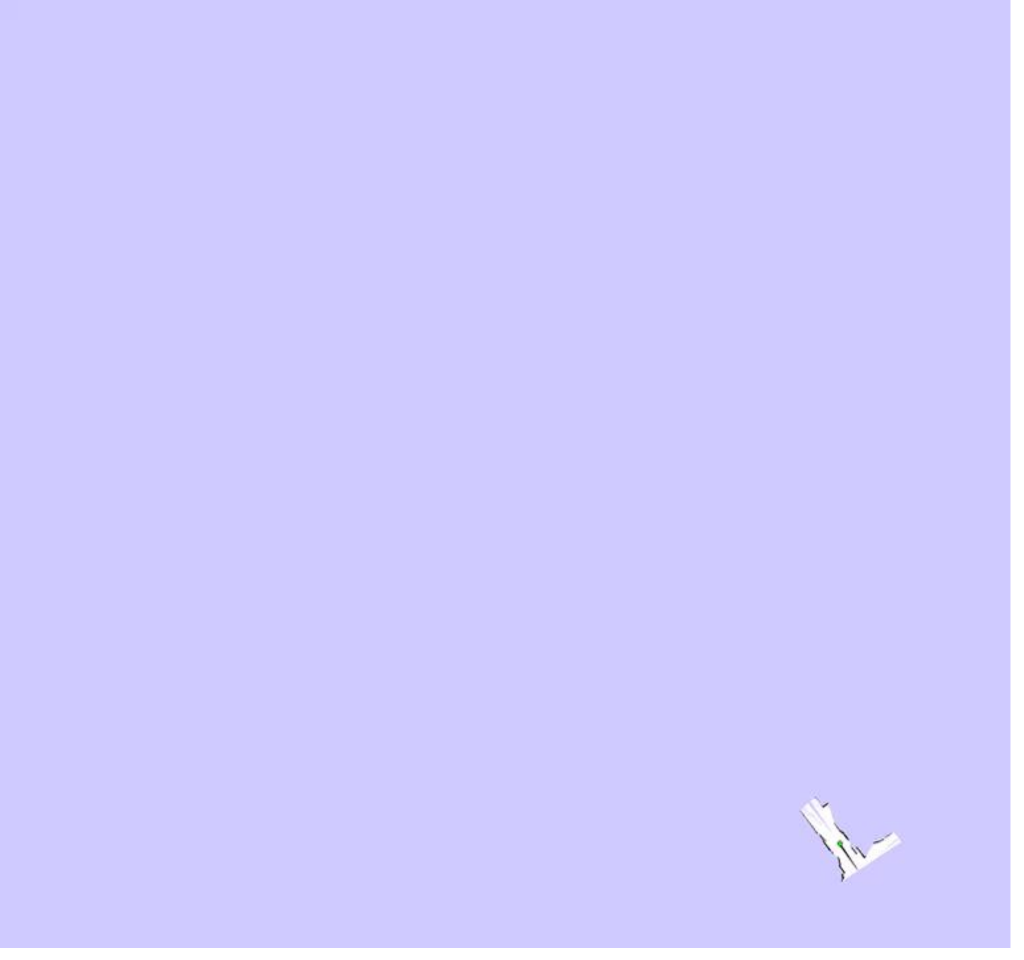
Must store

- $P(O|S)$
- $P(S_{t+1}|S_t)$

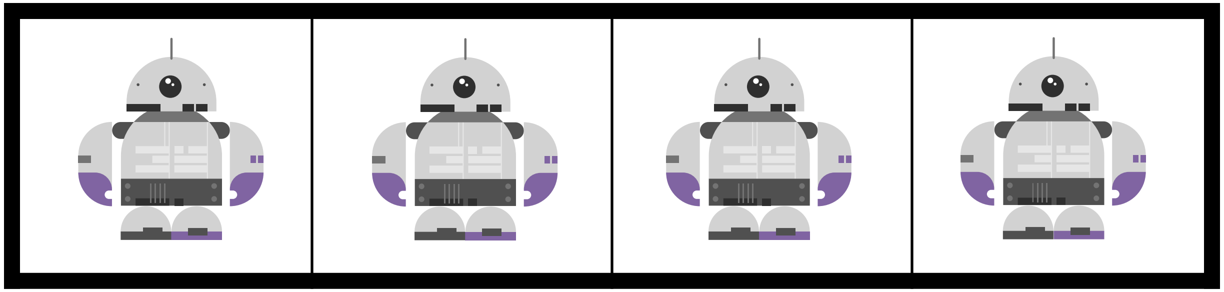
HMMs

- Monitoring/filtering
  - $P(S_t | O_0, \dots, O_t)$
  - e.g. estimate patient disease state
- Prediction
  - $P(S_t | O_0, \dots, O_k), k < t$
  - e.g. Given first two phonemes, what word?
- Smoothing
  - $P(S_t | O_0, \dots, O_k), k > t$
  - What happened back there?
- Most likely path
  - $P(S_0, \dots, S_t | O_0, \dots, O_t)$
  - How did I get here?

Most likely path



Example: robot localisation

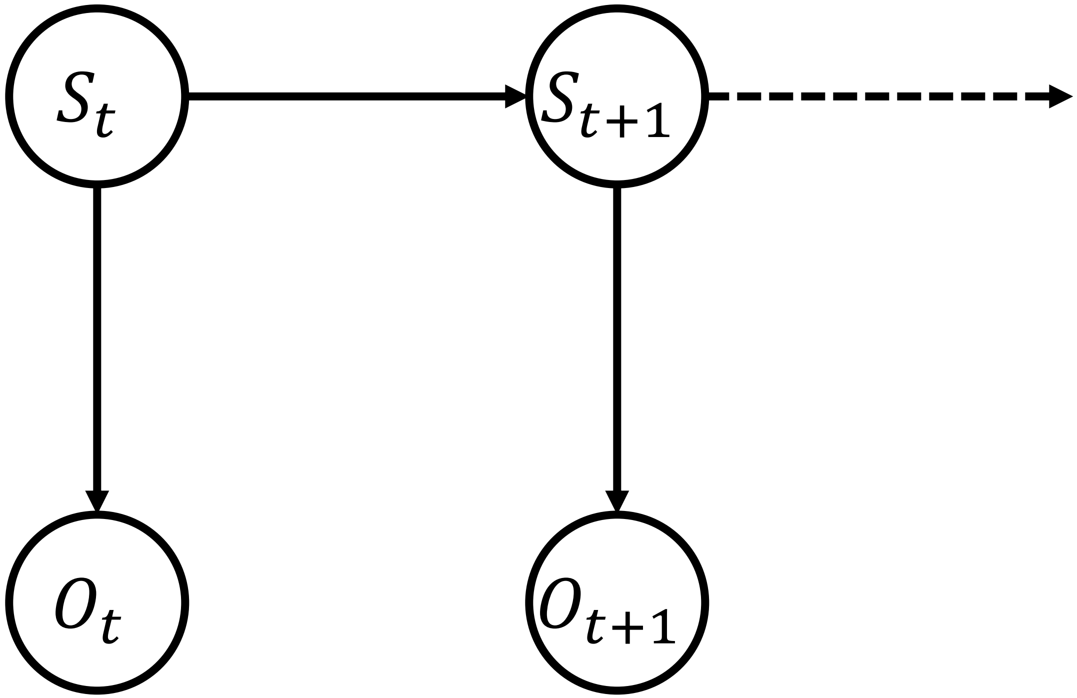


- We start off not knowing where robot is
- Uniform prior?
- Robot sense: obstacles up and down. Update!
- Robot moves right: updates distribution
- Obstacles up and down – update distribution

What happened?

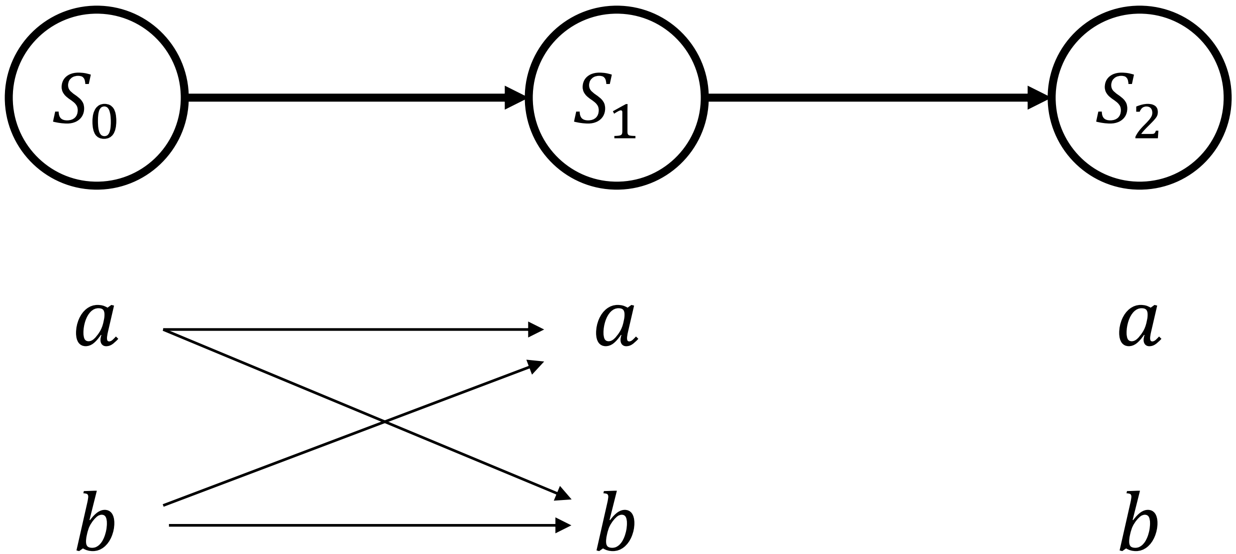
- Instance of robot tracking – filtering
- Could also:
  - Predict (where will robot be in 3 steps?)
  - Smooth (where was the robot?)
  - Most likely path (what was the robot's path?)
- All questions are about HMM's state at various times

How?



- Let's look at $P(S_t)$ – no observations
  - Assume we have CPTs

Prediction



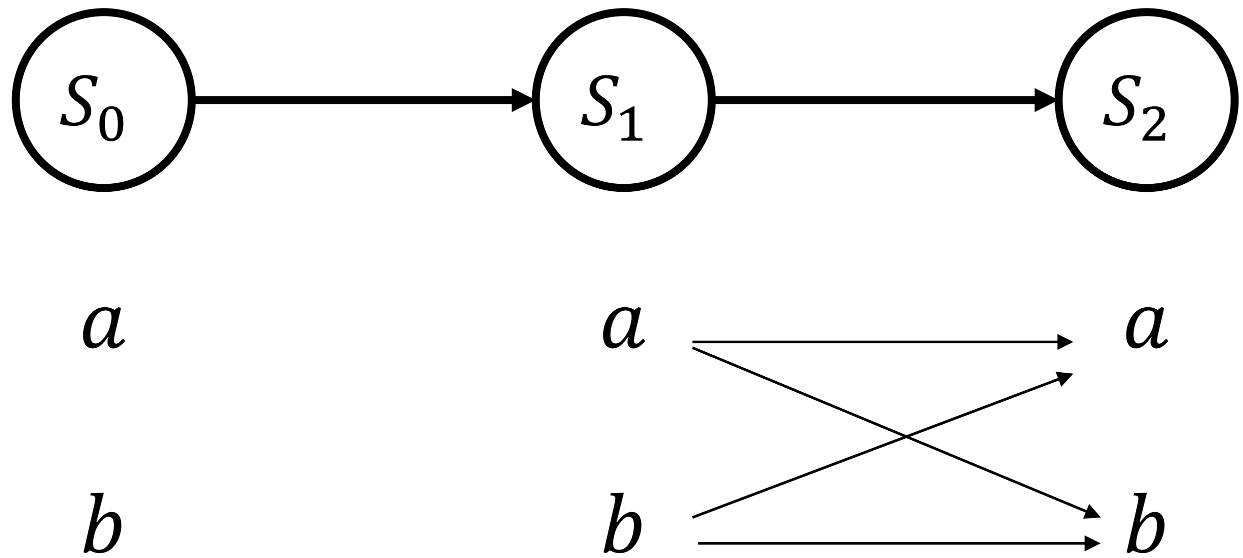
$P(S_0)$

(prior)

$P(S_1 = a) =$
$P(S_0 = a)P(a|a) +$
$P(S_0 = b)P(a|b)$

$P(S_1 = b) =$
$P(S_0 = a)P(b|a) +$
$P(S_0 = b)P(b|b)$

Prediction



$P(S_0)$

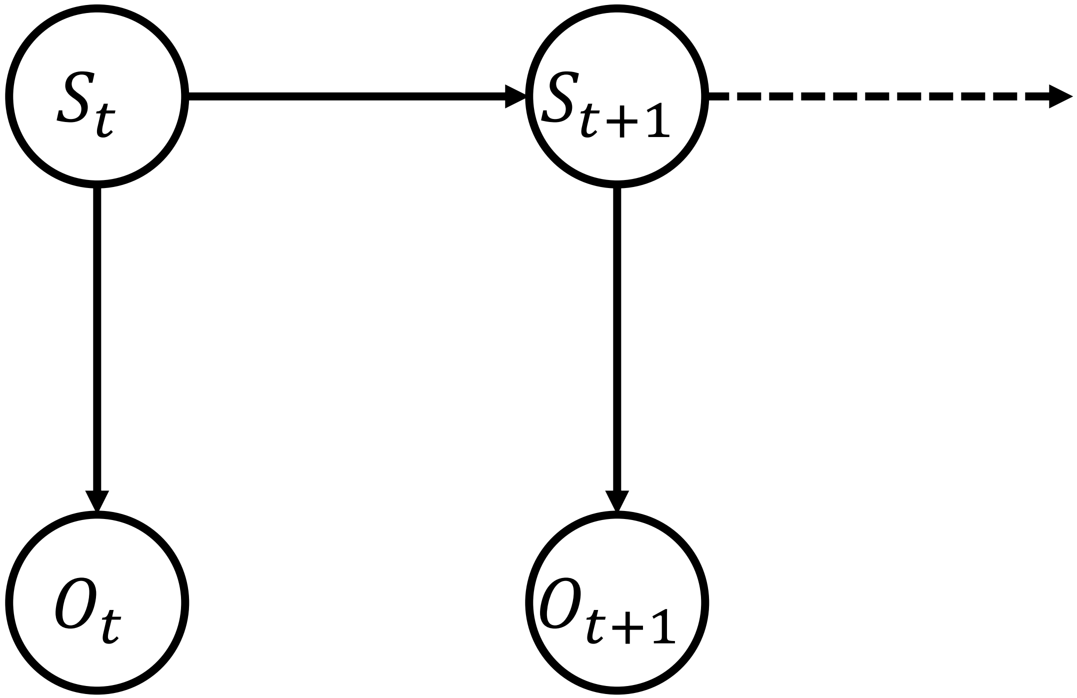
(prior)

$P(S_1)$

$$P(S_2 = a) = P(S_1 = a)P(a|a) + P(S_1 = b)P(a|b)$$

$$P(S_2 = b) = P(S_1 = a)P(b|a) + P(S_1 = b)P(b|b)$$

Filtering



$$\underset{S_t}{\operatorname{argmax}} P(S_t | O_0, \ldots, O_t)$$

Filtering

- Want $P(S_t|O_0, \dots, O_t)$
- Let's start with $P(S_t, O_0, \dots, O_t)$

$$P(S_t, O_0, \dots, O_t) = \sum_i P(S_t, {\color{red}S_{t-1} = s_i}, O_0, \dots, O_t)$$

$$= \sum_i P(O_t|S_t) P(S_t|S_{t-1} = s_i) P(S_{t-1} = s_i, O_0, \dots, O_{t-1})$$

$$= P(O_t|S_t) \sum_i P(S_t|S_{t-1} = s_i) P(S_{t-1} = s_i, O_0, \dots, O_{t-1})$$

Forward algorithm

- Let $F(k,0) = P(S_0 = s_k)P(O_0|S_0 = s_k)$

For $t = 1, \dots T$:

For k in possible states:

$$F(k,t) = P(O_t|S_t = s_k)\sum_i P(s_k|s_i)F(i,t-1)$$

- $F(k,T)$ is $P(S_T = s_k, O_0, \dots, O_T)$
  - Just take max state for $F(k,T)$
  - Can normalise to get $P(S_T|O_0, \dots, O_T)$

Smoothing

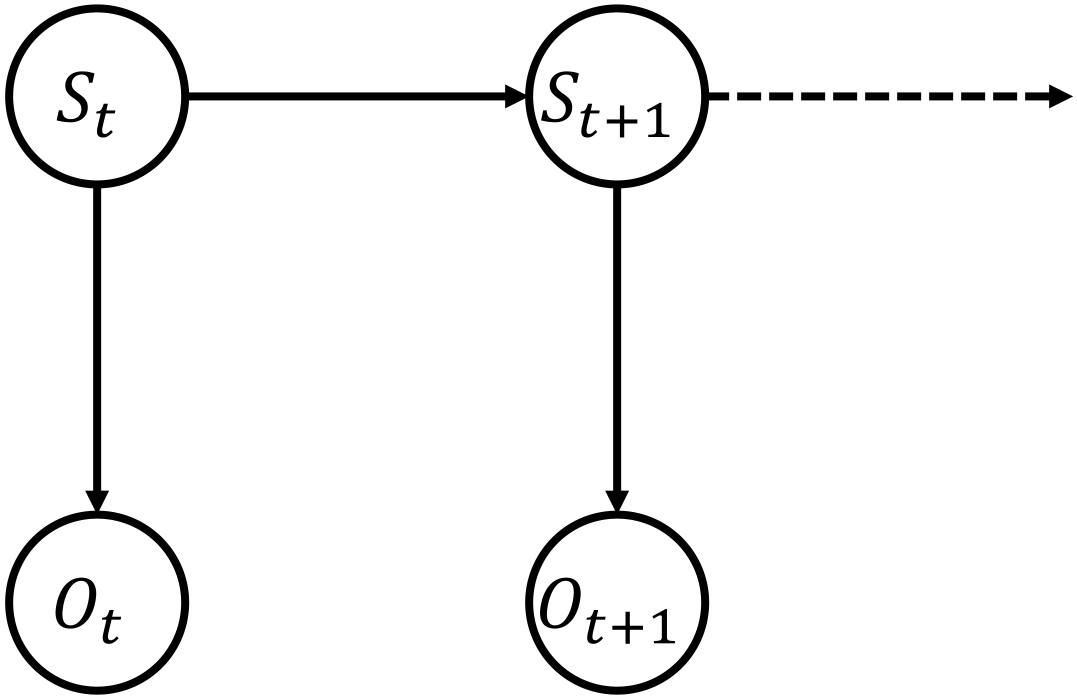
- $P(S_t|O_0, \ldots, O_k), k > t$ – given data of length k, find $P(S_t)$ for earlier t

- Bayes rule:
  - $P(S_t|O_0, \ldots, O_k) \propto P(O_t, \ldots, O_k|S_t)P(S_t|O_0, \ldots, O_t)$

Forward algorithm

- Compute using backward pass:
  - $P(O_i, \ldots, O_k|S_i)$ computed using similar recursion
  - Forward-backward algorithm

Most likely path



$$\underset{S_0 \ldots S_t}{\operatorname{argmax}} P(S_0, \ldots, S_t | O_0, \ldots, O_t)$$

Viterbi algorithm

- Similar logic to highest probability state, but:
  - We seek a path, not a state
  - Single highest probability path
  - Therefore look for highest probability of *(ancestor probability times observation probability)*
  - Maintain link matrix to read path backwards

- Similar dynamic programming algorithm, but replace sum with max

Viterbi algorithm

Most likely path $S_0, \ldots, S_T$

$V_{i,k}$: probability of max prob path ending in state s_k including observations up to O_i (at time $\mathrm{t} = i$)

$L_{i,k}$: most likely predecessor of state s_k at time i

For each state s_k:

$$V_{0,k} = P(O_0|s_k)P(s_k)$$

$$L_{0,k} = 0$$

for $i = 1 \ldots T$:

for each k:

$$V_{i,k} = P(O_i|s_k)\max_x P(s_k|s_x)\, V_{i-1,x}$$

$$L_{i,k} = \mathrm{argmax}_x\, P(s_k|s_x)V_{i-1,x}$$

observation model

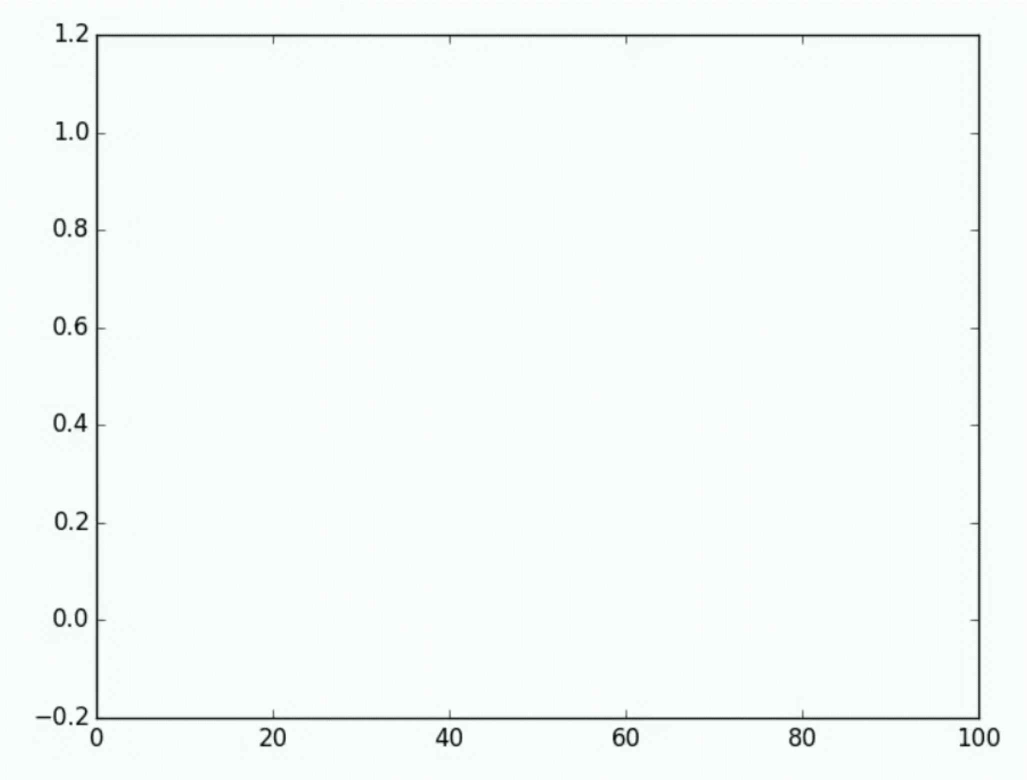
transition model

probability of path

Most likely ancestor

Common form

- Very common form:
  - Noisy observations of true state



Viterbi

- “The algorithm has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications, and 802.11 wireless LANs” (Wikipedia)