Hidden Markov Models

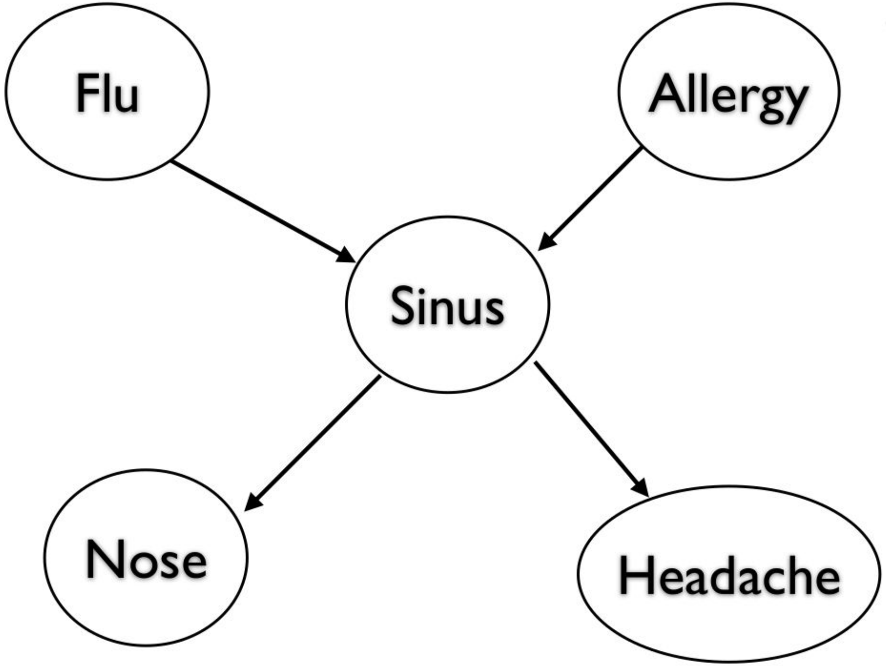
Eric Rosen
eric.andrew.rosen@gmail.com

Slides adapted from

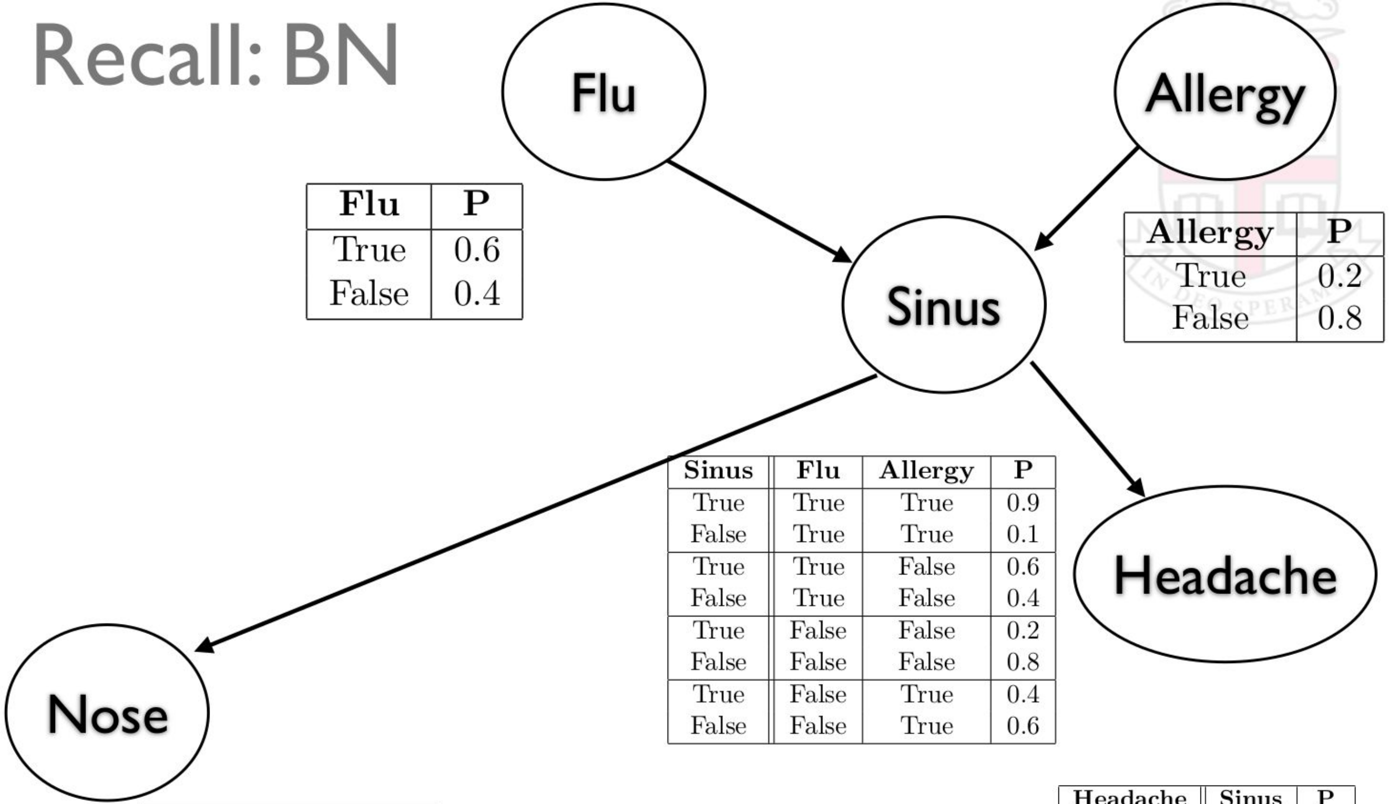
George Konidaris
gdk@cs.brown.edu

Fall 2023

Recall: Bayesian Network



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	False	0.2
False	False	False	0.8
True	False	True	0.4
False	False	True	0.6

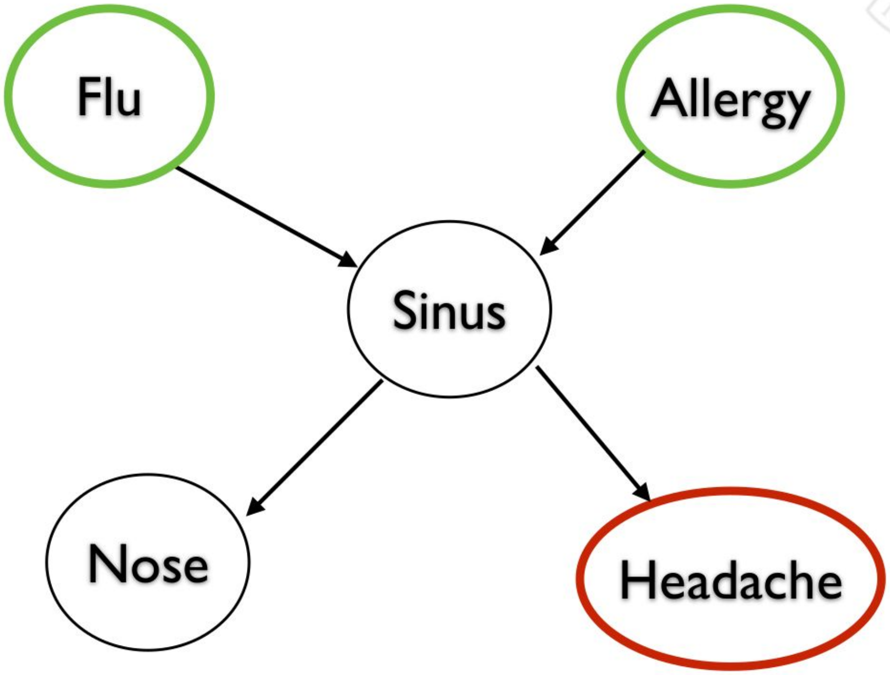
Headache	Sinus	P
True	True	0.6
False	True	0.4
True	False	0.5
False	False	0.5

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

joint: 32 (31) entries

Inference

Given A compute P(B | A).



Time

Bayesian Networks (so far) contain no notion of ***time***.

However, in many applications:

- Target tracking
- Patient monitoring
- Speech recognition
- Gesture recognition

… how a signal changes over time is critical.

States

In probability 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

The weather today can be:

- Hot
- Cold
- Chilly
- Freezing

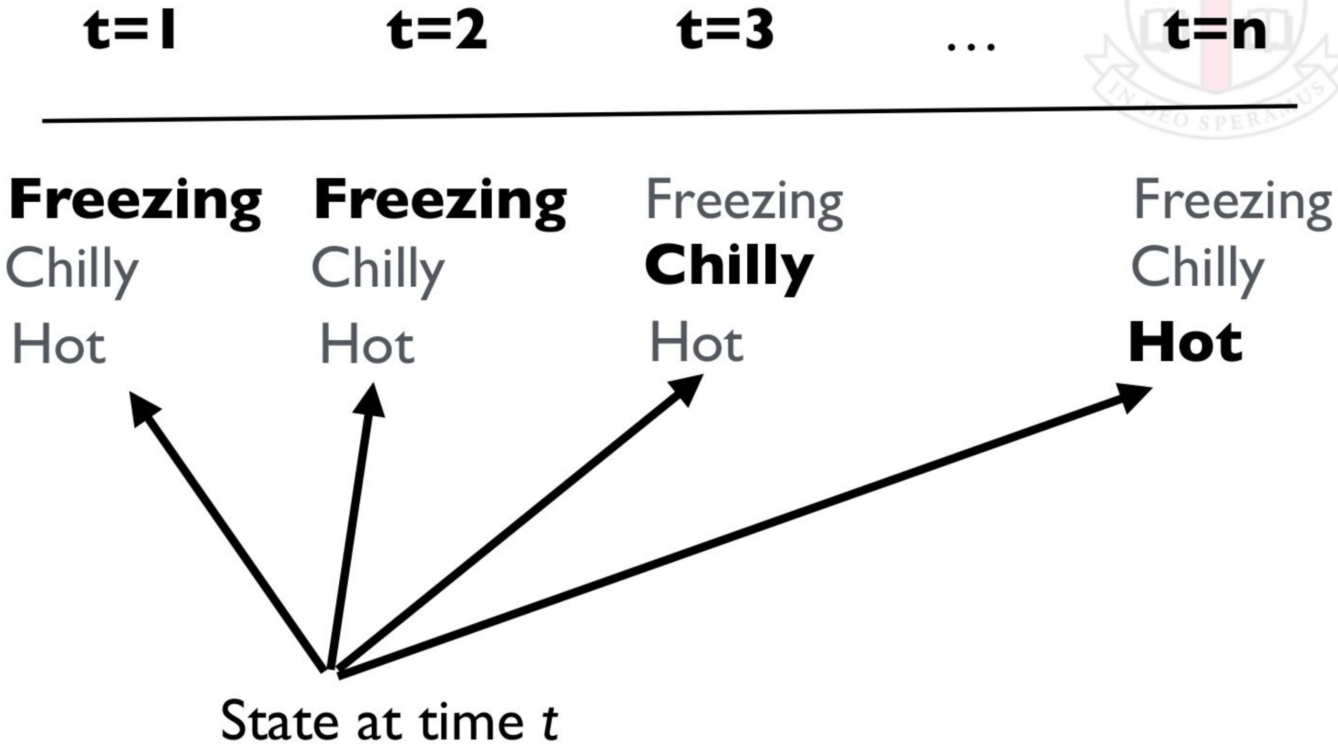
The weather has four *states*.

At each ***point in time***, the system is in ***one (and only one) state***.

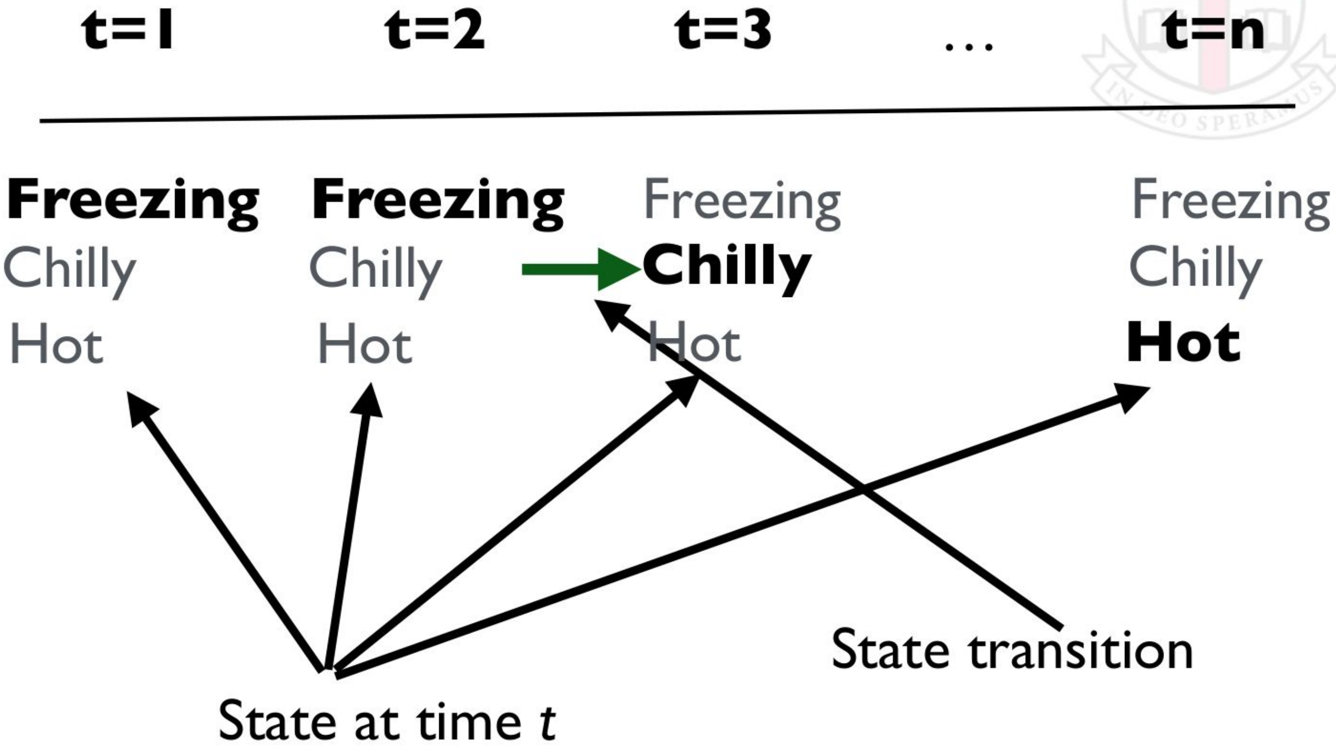
Example

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

Example


t=1
t=2
t=3
…
t=n
Freezing Chilly Hot
Freezing Chilly Hot
Freezing **Chilly** Hot
Freezing Chilly **Hot**
State at time *t*

Example



t=1
t=2
t=3
…
t=n
Freezing
Chilly
Hot
Freezing
Chilly
Hot
Freezing
Chilly
Hot
Freezing
Chilly
Hot
State transition
State at time t

The Markov Assumption

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

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

The Markov Assumption

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

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

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

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

А. А. Марков (1886).

Special kind of independence assumption:

- *Future independent of past given present.*

Markov Assumption

Model that has it is a ***Markov model***.

Sequence of states thus generated is a ***Markov chain***.

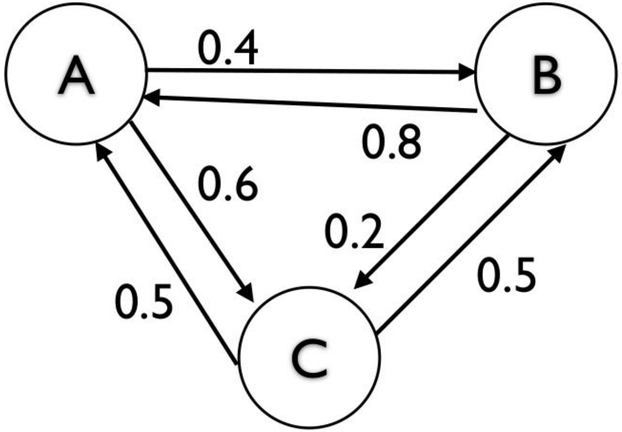
Definition of a state:

- Sufficient statistic for history
- $P(S_t|S_{t-1}, ..., S_0) = P(S_t|S_{t-1})$

Can describe transition probabilities with matrix:

- $P(S_i \mid S_j)$
- Steady state probabilities.
- Convergence rates.

State Machines

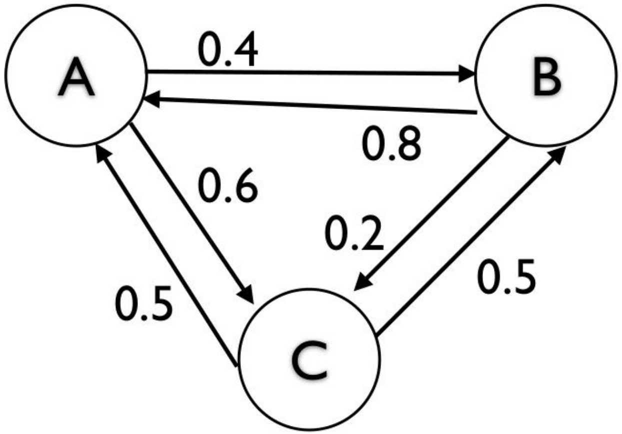


P(A | B) = 0.8
P(A | C) = 0.5
P(B | A) = 0.4
P(B | C) = 0.5
P(C | A) = 0.6
P(C | B) = 0.2

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



states not state vars!

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

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 cool but:

- Often state is not observed directly.
- State is latent, or hidden.

State:
forehand

Instead you see an *observation*, which contains information about the hidden state.

Examples

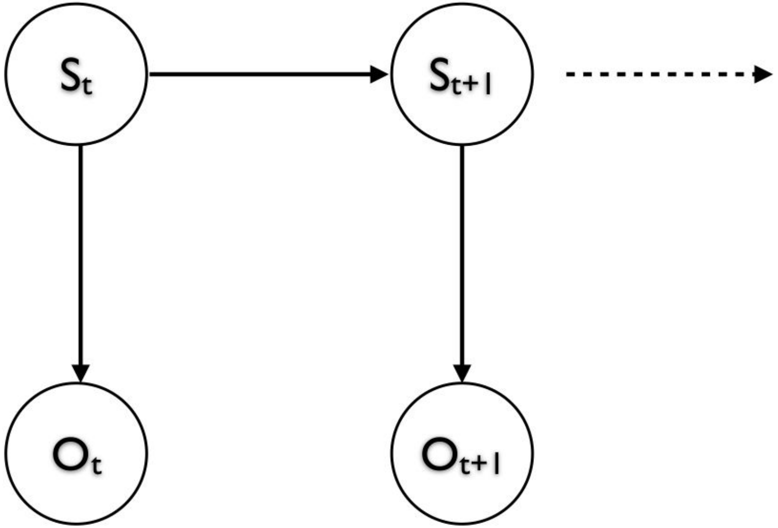
State	Observation
Word	Phoneme
Chemical State	Color, Smell, etc.
Flu?	Runny Nose
Cardiac Arrest?	Pulse

Examples

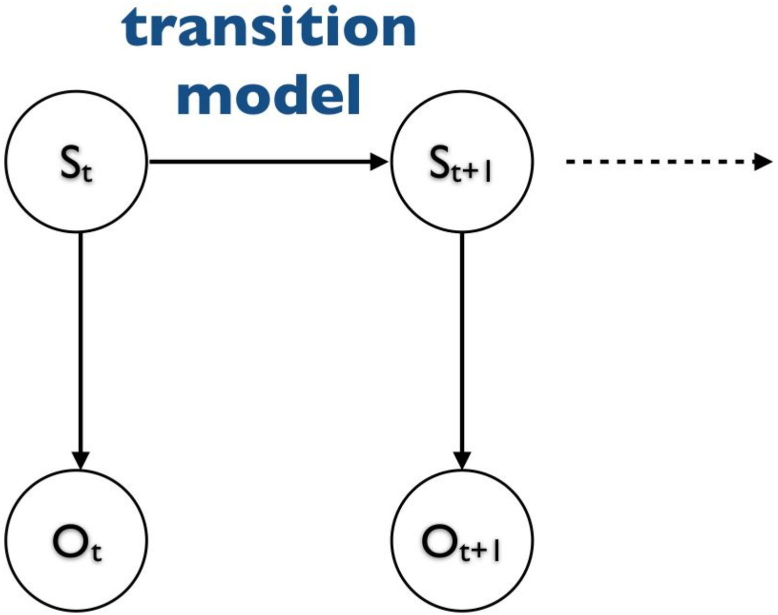
State	Observation
Word	Phoneme
Chemical State	Color, Smell, etc.
Flu?	Runny Nose
Cardiac Arrest?	Pulse

Sensor

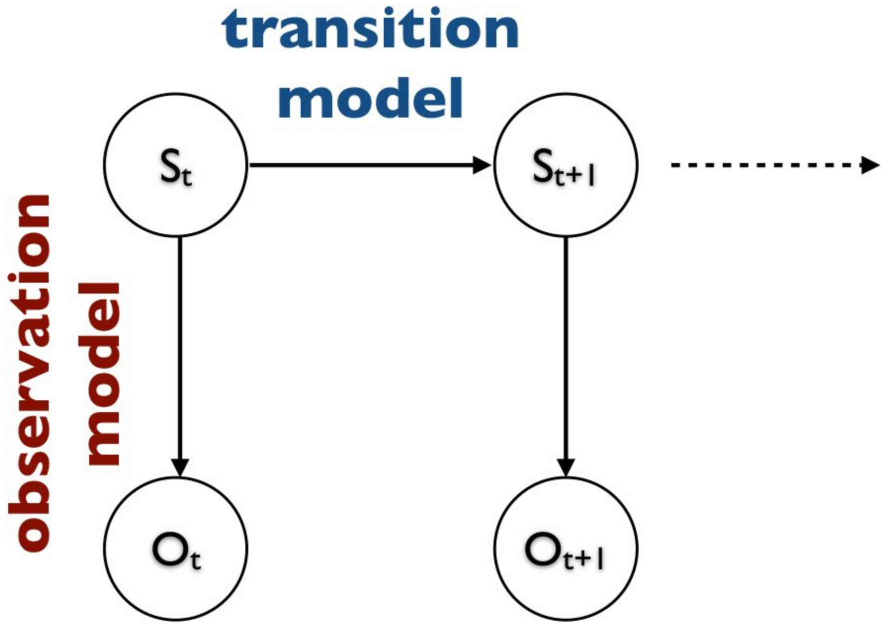
Hidden Markov Models



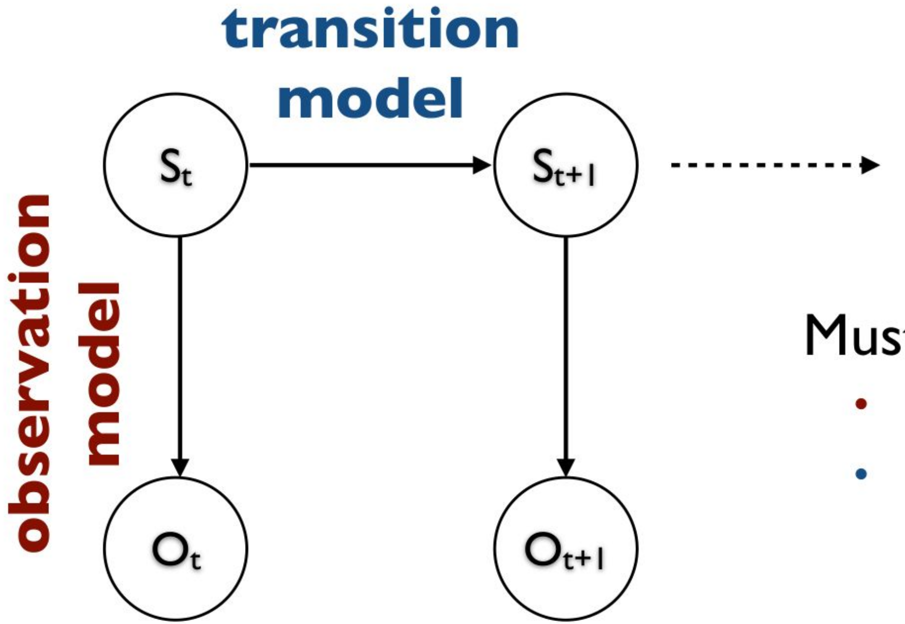
Hidden Markov Models



Hidden Markov Models



Hidden Markov Models



Must store:

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

HMMs

Monitoring/Filtering

- $P(S_t \mid O_0 \ldots O_t)$
- E.g., estimate patient disease state.

Prediction

- $P(S_t \mid O_0 \ldots O_k)$, $k < t$.
- Given first two phonemes, what word?

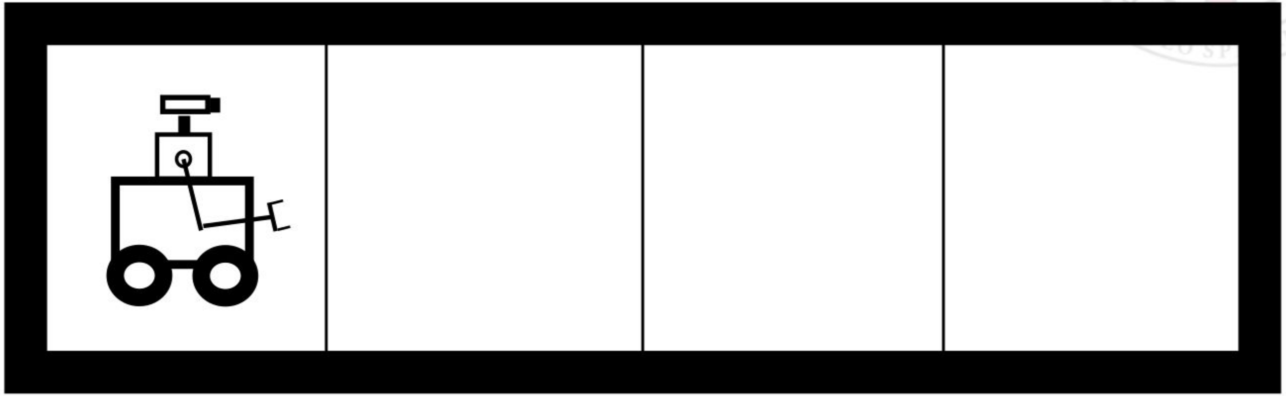
Smoothing

- $P(S_t \mid O_0 \ldots O_k)$, $k > t$
- What happened back there?

Most Likely Path

- $P(S_0 \ldots S_t \mid O_0 \ldots O_t)$
- How did I get here?

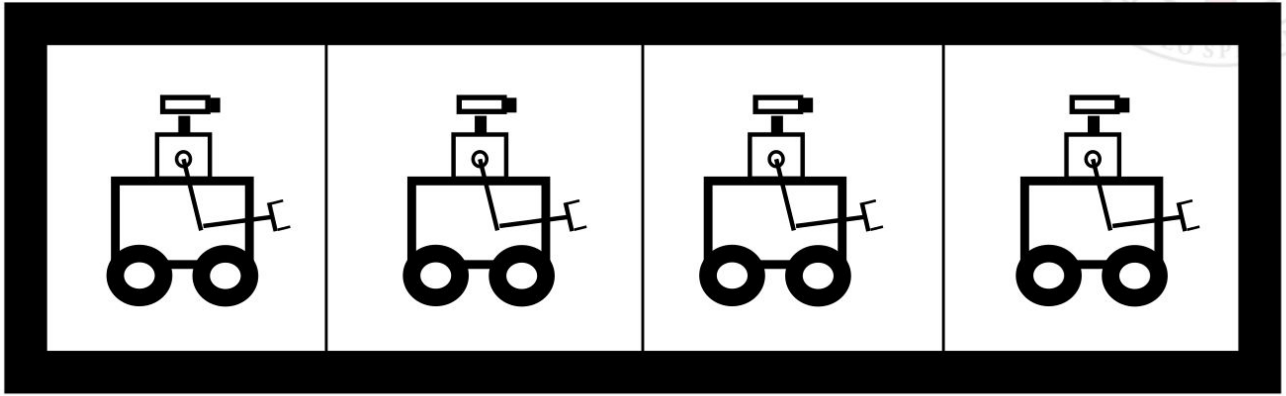
Example: Robot Localization



observations:
walls each side?

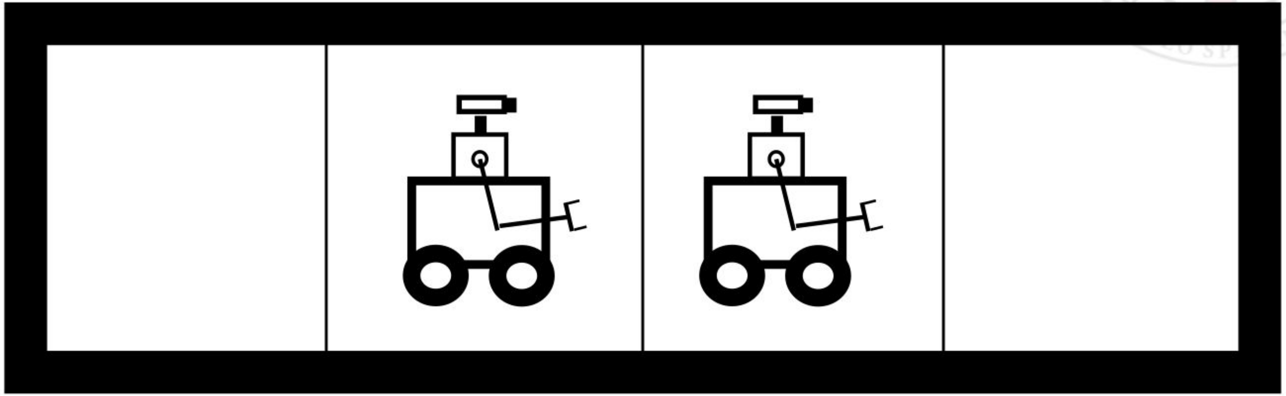
states:
position

Example: Robot Localization



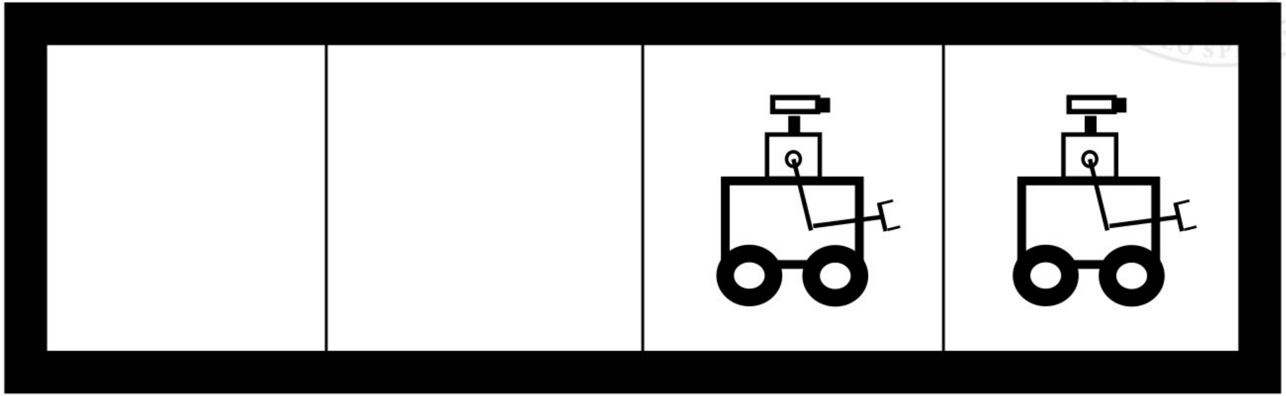
We start off not knowing where the robot is.

Example: Robot Localization



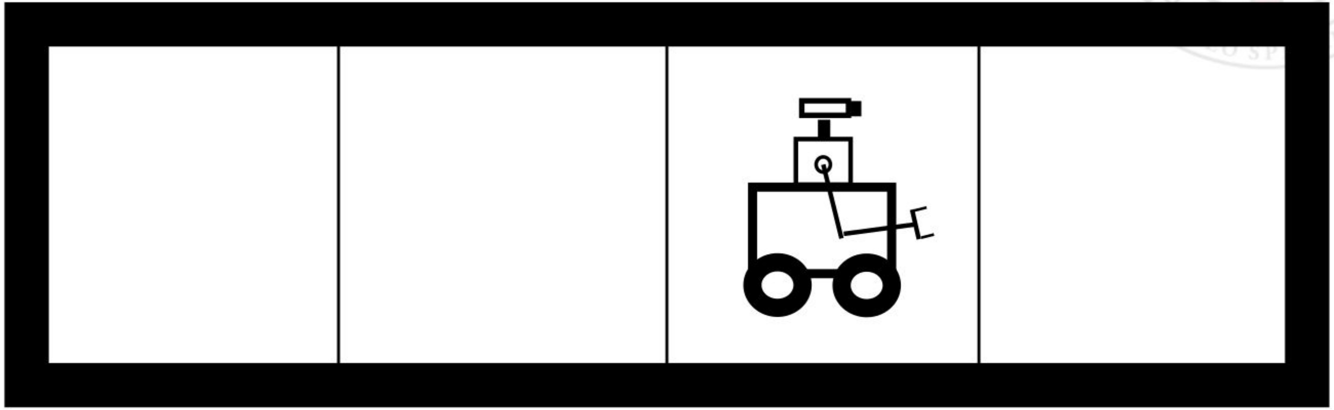
Robot sense: obstacles up and down.
Updates distribution.

Example: Robot Localization



Robot moves right: updates distribution.

Example: Robot Localization



Obstacles up and down, updates distribution.

What Happened

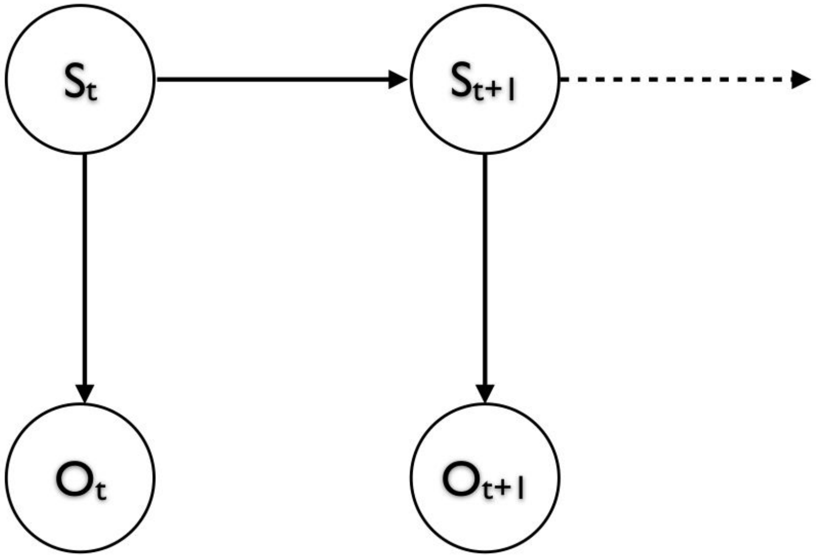
This is an instance of robot tracking - *filtering*.

Could also:

- Predict (where will the robot be in 3 steps?)
- Smooth (where was the robot?)
- Most likely path (what was the robot's path?)

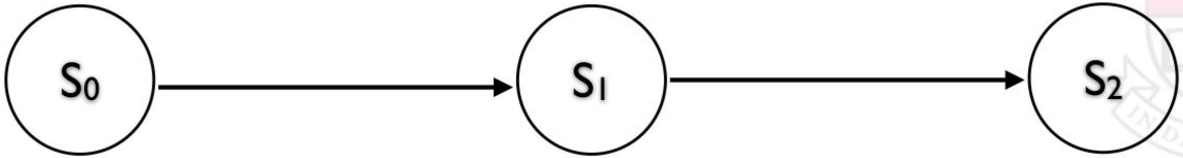
All of these are questions about the HMM's state at various times.

How?



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

Prediction

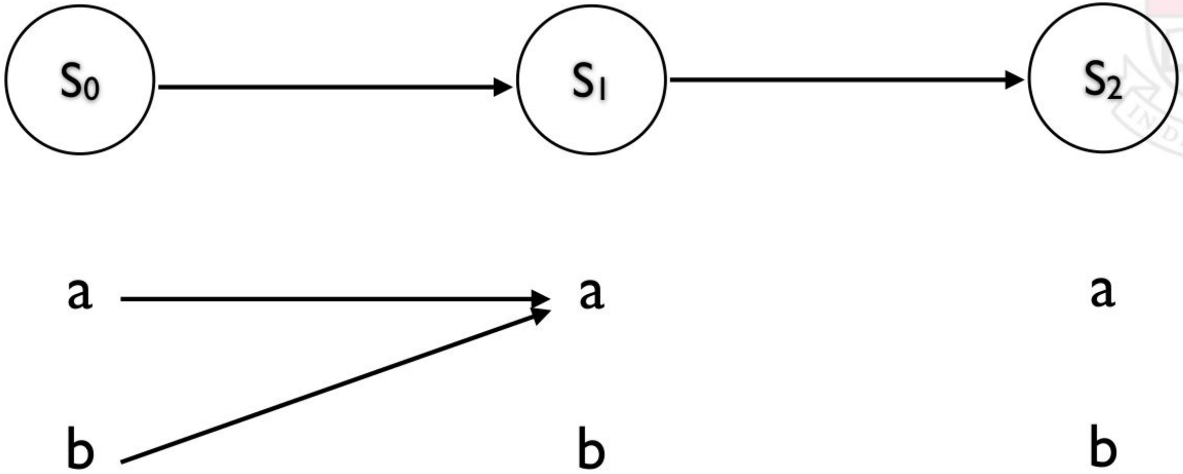


a a a

b b b

$P(S_0)$
(prior)

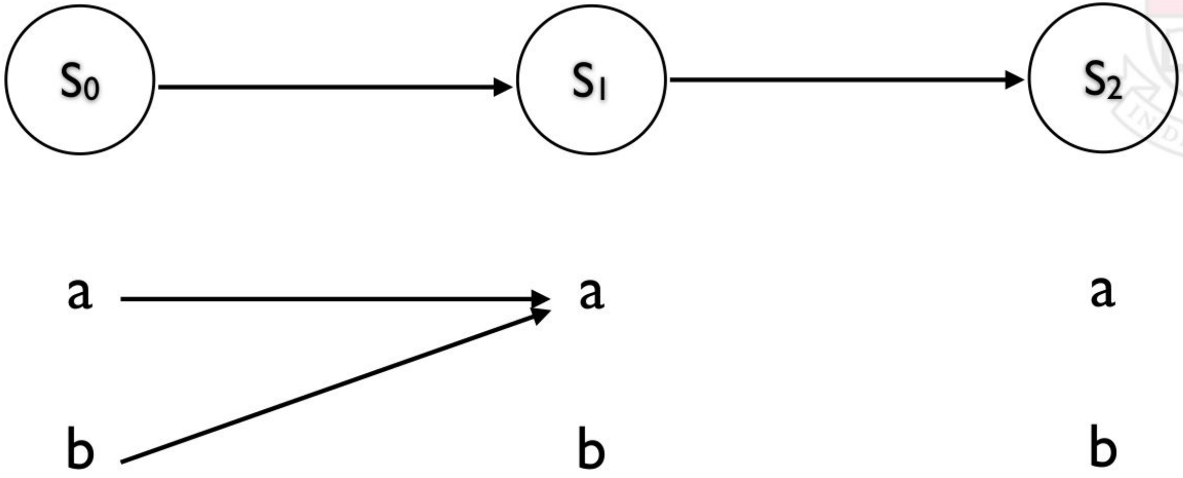
Prediction



$P(S_0)$

(prior)

Prediction

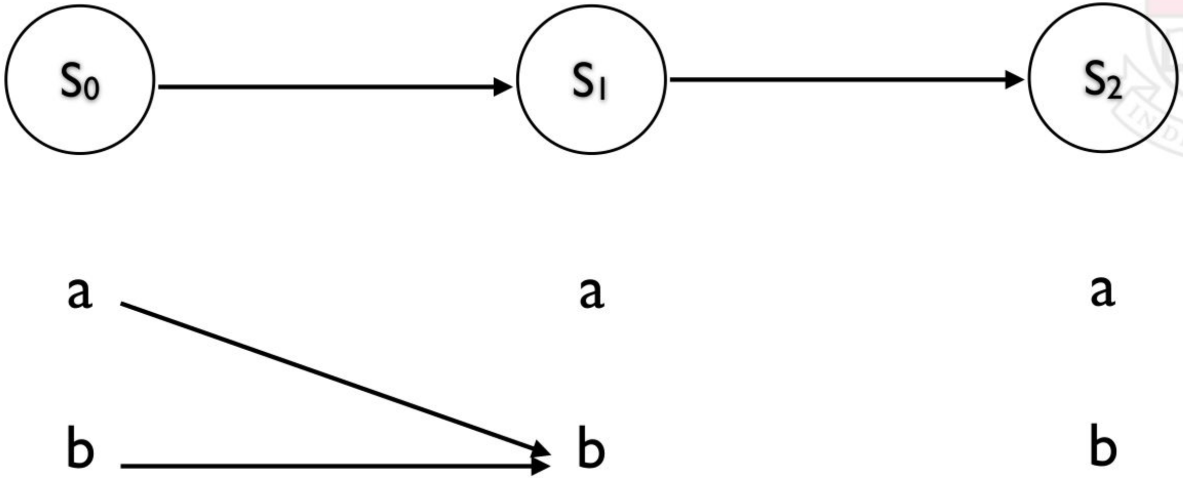


$P(S_0)$

(prior)

$$P(S_1 = a) = P(S_0 = a)P(a \mid a) + P(S_0 = b)P(a \mid b)$$

Prediction



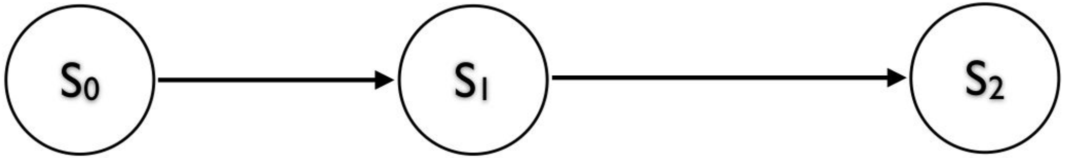
$P(S_0)$

(prior)

$$P(S_1 = a) = P(S_0 = a)P(a \mid a) + P(S_0 = b)P(a \mid b)$$

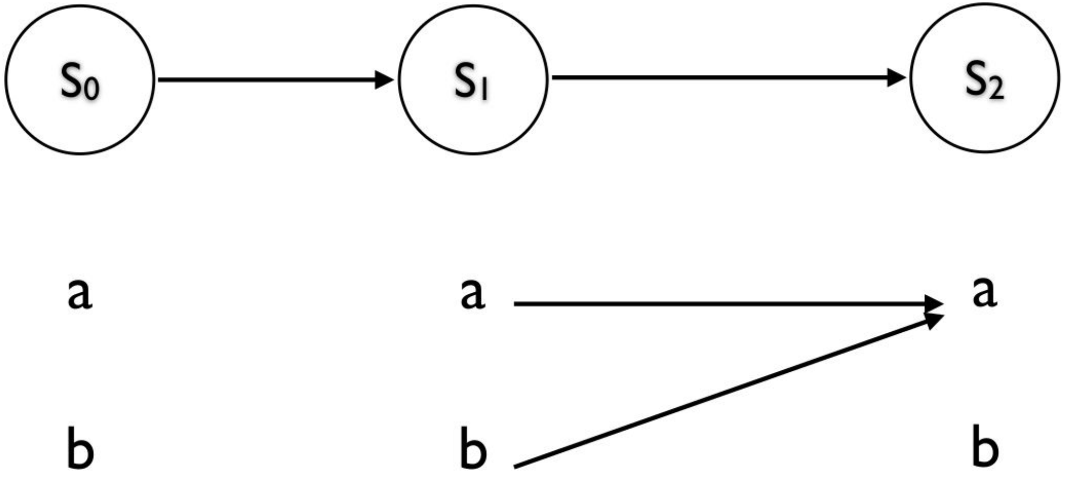
$$P(S_1 = b) = P(S_0 = a)P(b \mid a) + P(S_0 = b)P(b \mid b)$$

Prediction



a	a	a
b	b	b
$P(S_0)$ (prior)	$P(S_1)$	

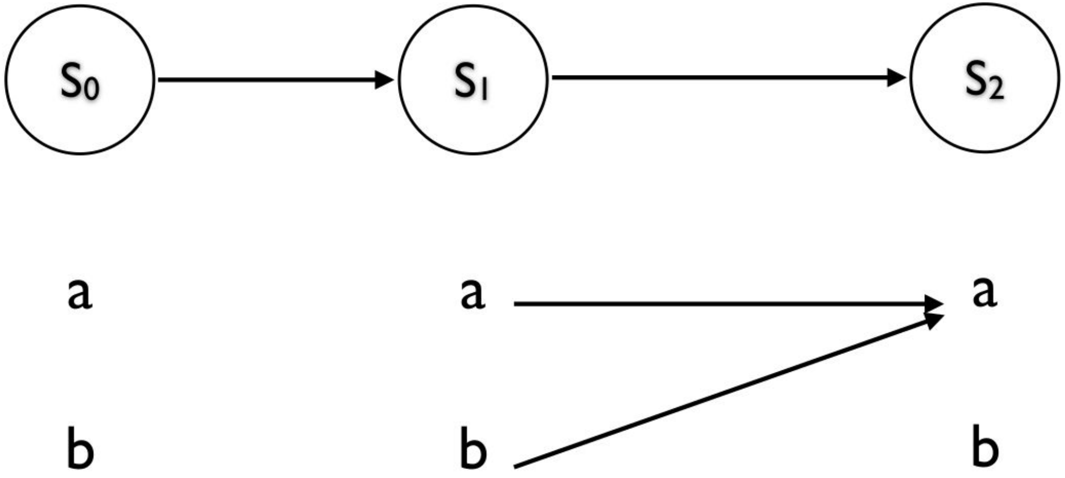
Prediction



$P(S_0)$ $P(S_1)$

(prior)

Prediction



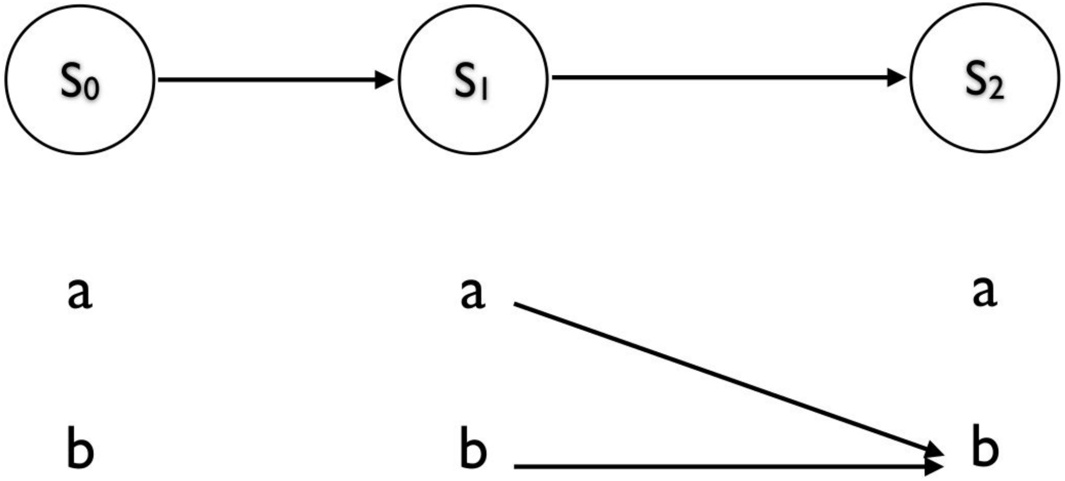
$P(S_0)$

(prior)

$P(S_1)$

$$P(S_2 = a) = P(S_1 = a)P(a \mid a) + P(S_1 = b)P(a \mid b)$$

Prediction



$P(S_0)$

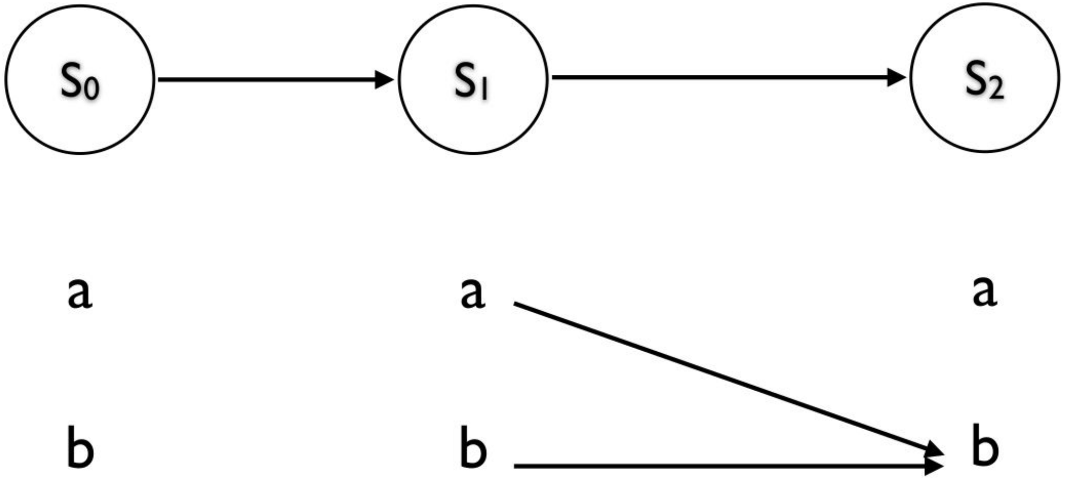
(prior)

$P(S_1)$

$$P(S_2 = a) = P(S_1 = a)P(a \mid a) + P(S_1 = b)P(a \mid b)$$

$$P(S_2 = b) = P(S_1 = a)P(b \mid a) + P(S_1 = b)P(b \mid b)$$

Prediction



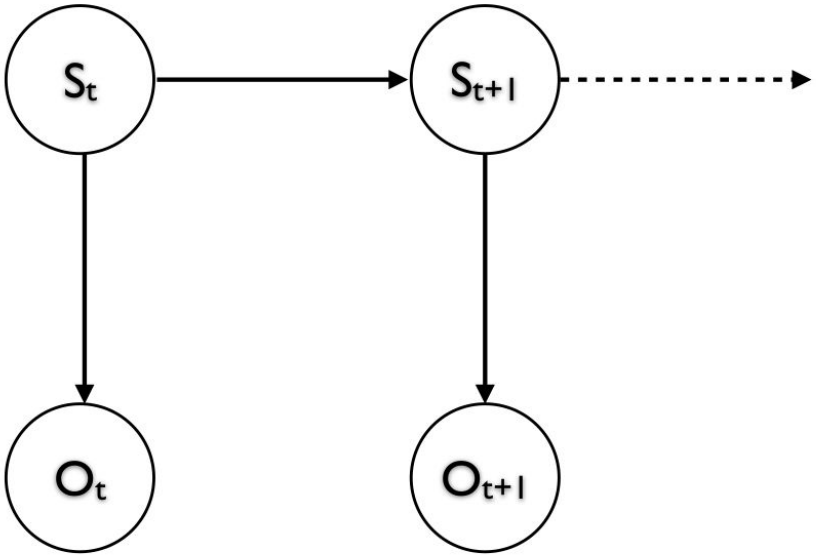
$P(S_0)$
(prior)

$P(S_1)$

$$P(S_2 = a) = P(S_1 = a)P(a \mid a) + P(S_1 = b)P(a \mid b)$$

$$P(S_2 = b) = P(S_1 = a)P(b \mid a) + P(S_1 = b)P(b \mid b)$$

Filtering



$$\max_{S_t} P(S_t \mid O_0 \ldots O_t).$$

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

Filtering

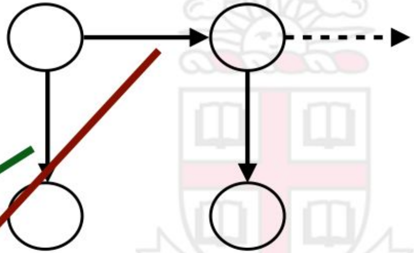
Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

Filtering



Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

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

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

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

Filtering

Where to start?

$P(S_t \mid O_0 \ldots O_t)$? Let's use $P(S_t, O_0 \ldots O_t)$.

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

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

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

Forward Algorithm

Let $F(k, 0) = P(S_0 = s_k)P(O_0 \mid S_0 = s_k)$.

For t = 1, …, 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 $P(S_T = s_k, O_0 \ldots O_T)$

(normalize to get $P(S_T \mid O_0 \ldots O_T)$)

Smoothing

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

Bayes Rule:

- $P(S_t \mid O_0 \ldots O_k) \propto P(O_t \ldots O_k \mid S_t)\, P(S_t \mid O_0 \ldots O_t)$

Smoothing

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

Bayes Rule:

- $P(S_t \mid O_0 \ldots O_k) \propto P(O_t \ldots O_k \mid S_t) P(S_t \mid O_0 \ldots O_t)$

forward algorithm

Smoothing

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

Bayes Rule:

- $P(S_t \mid O_0 \ldots O_k) \propto P(O_t \ldots O_k \mid S_t) P(S_t \mid O_0 \ldots O_t)$

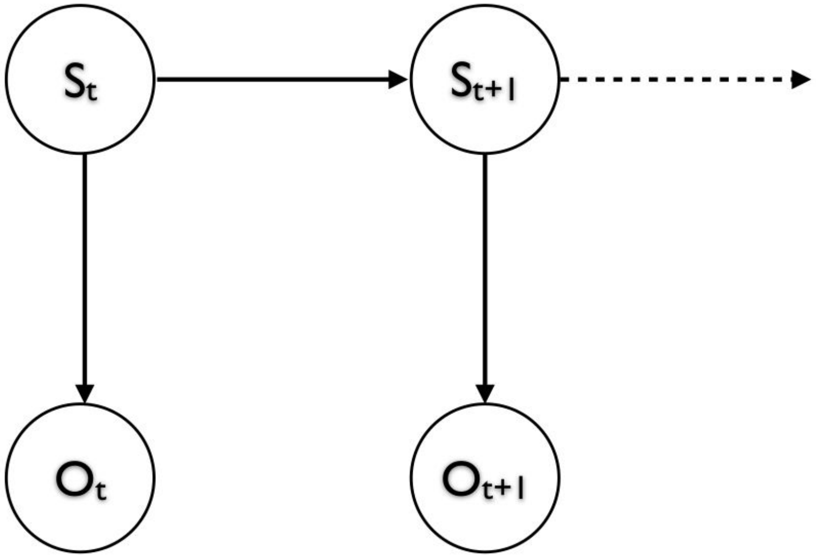
forward algorithm

Compute using backward pass:

$P(O_i \ldots O_k \mid S_i)$ computed using similar recursion.

Forward-backward algorithm.

Most Likely Path



$$\max_{S_0 \ldots S_t} P(S_0 \ldots S_t \mid O_0 \ldots O_t)$$

Viterbi

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, replace *sum* with *max*.

Viterbi Algorithm

Most likely path $S_0 \ldots S_T$:

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

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

For each state s_k:

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

$L_{0,k} = 0$

For $i = 1 \ldots T$,

For each k:

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

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

Viterbi Algorithm

Most likely path $S_0 \ldots S_T$:

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

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

For each state s_k:

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

$L_{0,k} = 0$

For $i = 1 \ldots T$,

For each k:

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

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

observation model

Viterbi Algorithm

Most likely path $S_0 \ldots S_T$:

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

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

For each state s_k:

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

$L_{0,k} = 0$

For $i = 1 \ldots T$,

For each k:

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

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

observation model

transition model

Viterbi Algorithm

Most likely path $S_0 \ldots S_T$:

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

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

For each state s_k:

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

$L_{0,k} = 0$

For $i = 1 \ldots T$,

For each k:

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

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

observation model

transition model

probability of path to x

Viterbi Algorithm

Most likely path $S_0 \ldots S_T$:

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

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

For each state s_k:

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

$L_{0,k} = 0$

For $i = 1 \ldots T$,

For each k:

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

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

observation model

transition model

probability of path to x

most likely ancestor