

# CS 188: Artificial Intelligence

## Hidden Markov Models



Instructors: Dan Klein and Pieter Abbeel --- University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at <http://ai.berkeley.edu>.]

# CS 188: Artificial Intelligence

## Hidden Markov Models

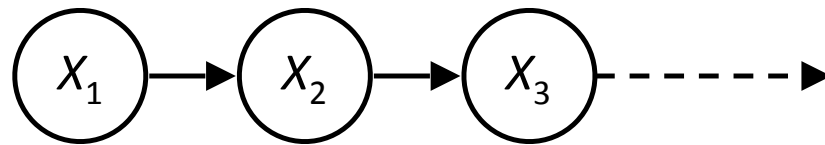


Instructors: Dan Klein and Pieter Abbeel --- University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at <http://ai.berkeley.edu>.]

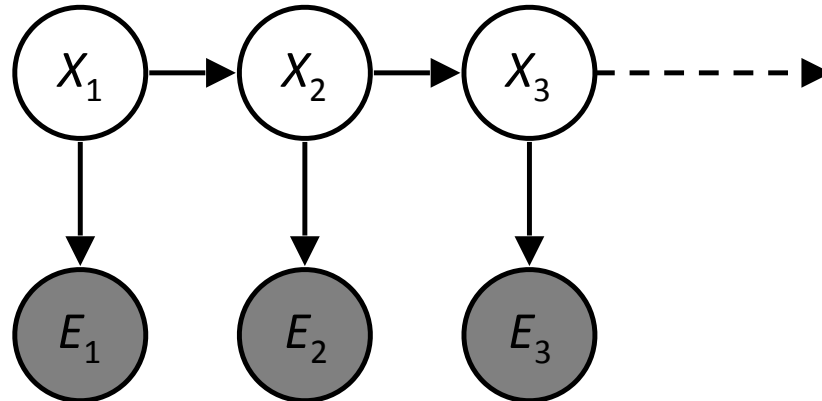
# Markov Chains

---



# Hidden Markov Models

---



# Demo: Ghostbusters

---

# Probability Recap

---

- Conditional probability

$$P(x|y) = \frac{P(x, y)}{P(y)}$$

- Product rule

$$P(x, y) = P(x|y)P(y)$$

- Chain rule

$$\begin{aligned} P(X_1, X_2, \dots, X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \dots \\ &= \prod_{i=1}^n P(X_i|X_1, \dots, X_{i-1}) \end{aligned}$$

- X, Y independent if and only if:  $\forall x, y : P(x, y) = P(x)P(y)$

- X and Y are conditionally independent given Z if and only if:

$$\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$$

$$X \perp\!\!\!\perp Y | Z$$

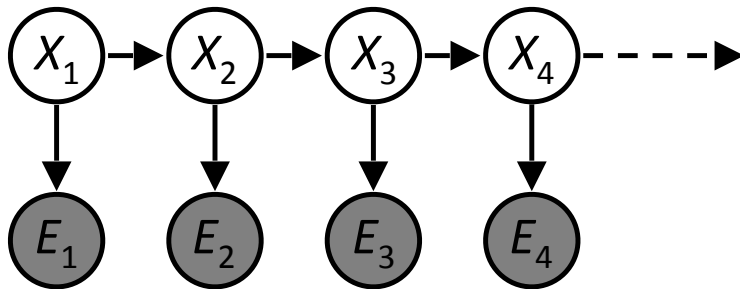
# Hidden Markov Models

---



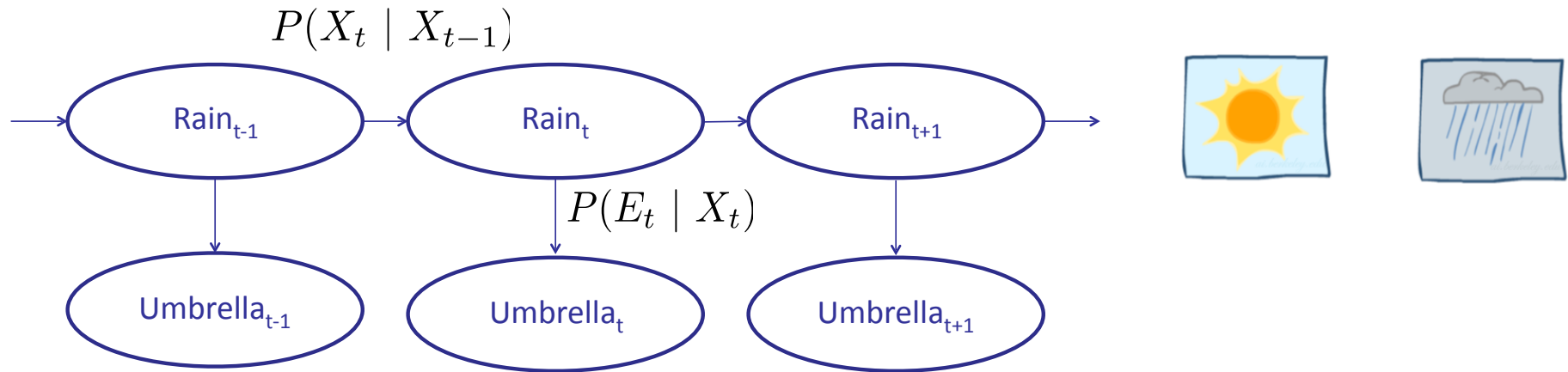
# Hidden Markov Models

- Markov chains not so useful for most agents
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states  $X$
  - You observe outputs (effects) at each time step





# Example: Weather HMM



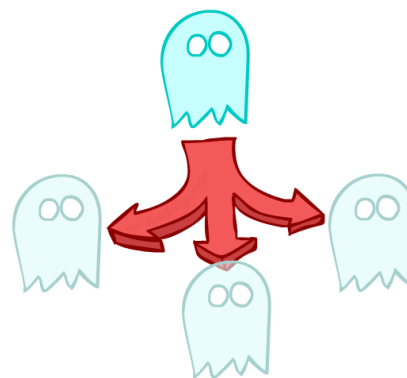
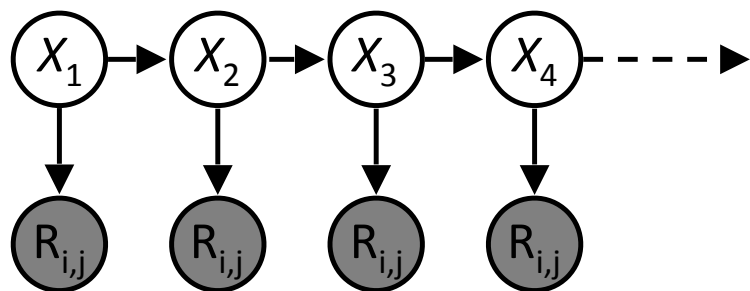
## ■ An HMM is defined by:

- Initial distribution:  $P(X_1)$
- Transitions:  $P(X_t | X_{t-1})$
- Emissions:  $P(E_t | X_t)$

$R_t$	$R_{t+1}$	$P(R_{t+1}   R_t)$	$R_t$	$U_t$	$P(U_t   R_t)$
+r	+r	0.7	+r	+u	0.9
+r	-r	0.3	+r	-u	0.1
-r	+r	0.3	-r	+u	0.2
-r	-r	0.7	-r	-u	0.8

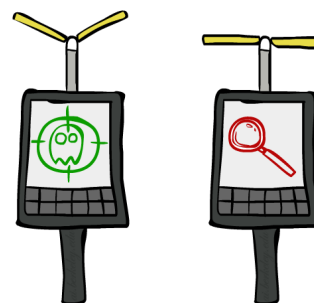
# Example: Ghostbusters HMM

- $P(X_1)$  = uniform
- $P(X|X')$  = usually move clockwise, but sometimes move in a random direction or stay in place
- $P(R_{ij}|X)$  = same sensor model as before: red means close, green means far away.



1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$P(X_1)$

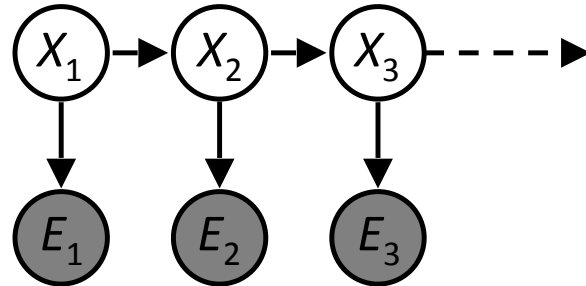


1/6	1/6	1/2
0	1/6	0
0	0	0

$P(X|X'=<1,2>)$

[Demo: Ghostbusters – Circular Dynamics – HMM (L14D2)]

# Joint Distribution of an HMM



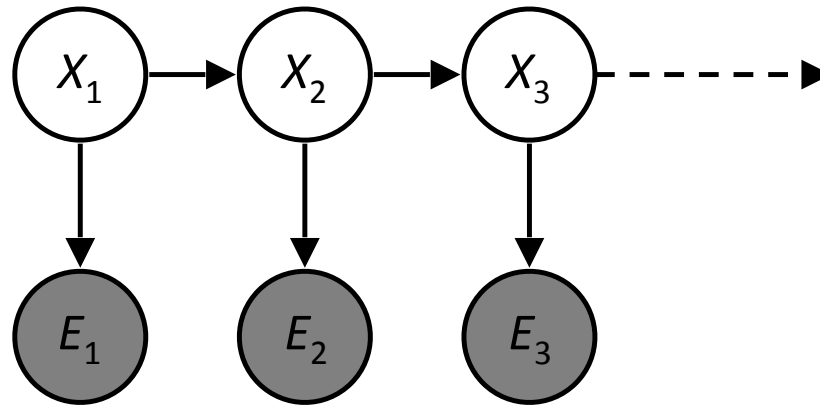
- Joint distribution:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- More generally:

$$P(X_1, E_1, \dots, X_T, E_T) = P(X_1)P(E_1|X_1) \prod_{t=2}^T P(X_t|X_{t-1})P(E_t|X_t)$$

# Implied Conditional Independencies



- Many implied conditional independencies, e.g.,

$$E_1 \perp\!\!\!\perp X_2, E_2, X_3, E_3 \mid X_1$$

# Real HMM Examples

---

- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options
- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

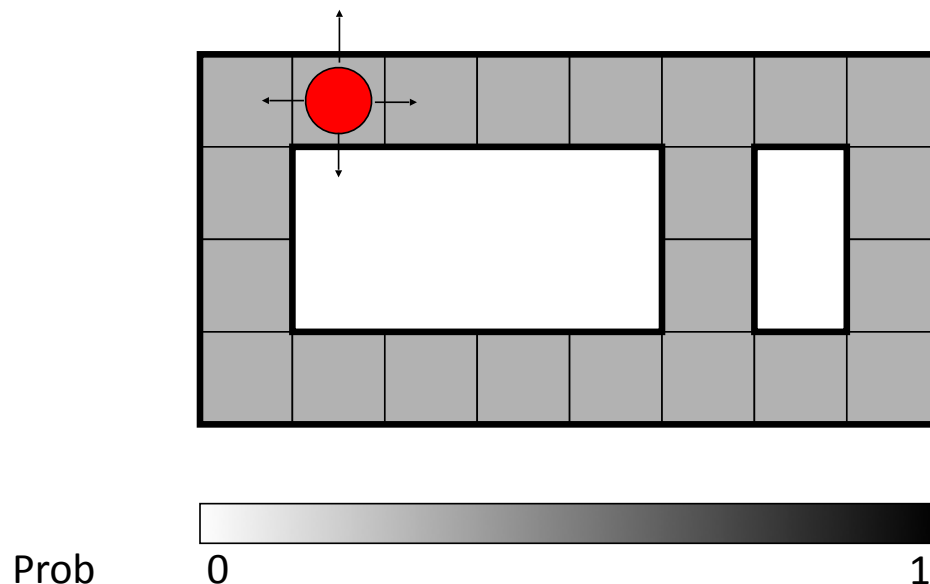
# Filtering / Monitoring

---

- Filtering, or monitoring, is the task of tracking the distribution  $B_t(X) = P_t(X_t \mid e_1, \dots, e_t)$  (the belief state) over time
- We start with  $B_1(X)$  in an initial setting, usually uniform
- As time passes, or we get observations, we update  $B(X)$
- The Kalman filter was invented in the 60's and first implemented as a method of trajectory estimation for the Apollo program

# Example: Robot Localization

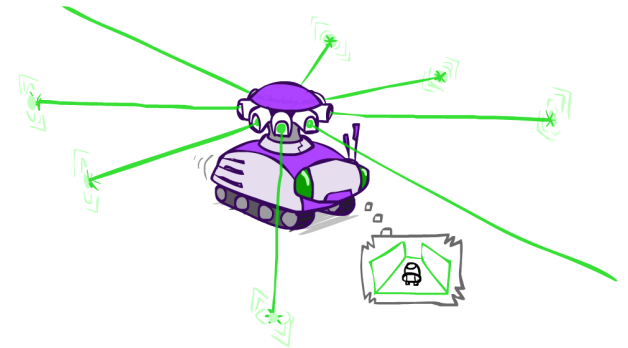
*Example from  
Michael Pfeiffer*



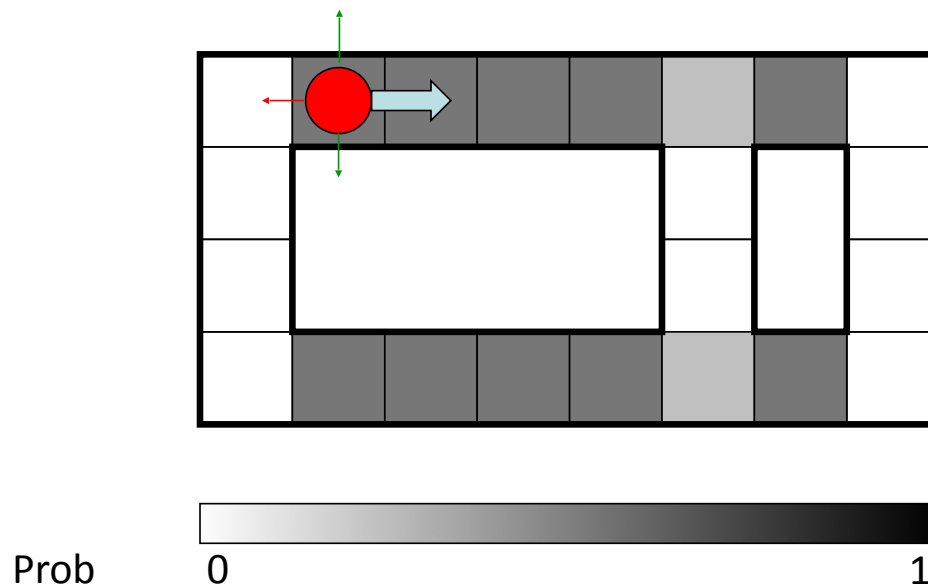
$t=0$

Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.

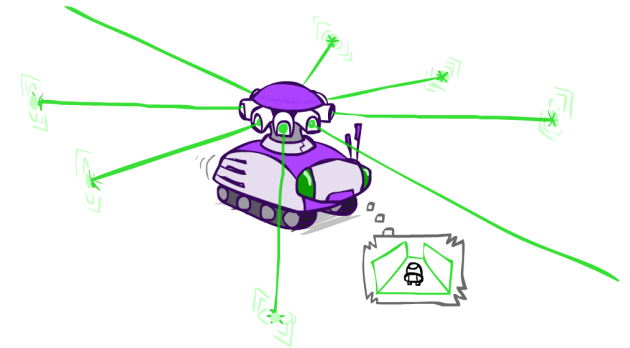


# Example: Robot Localization



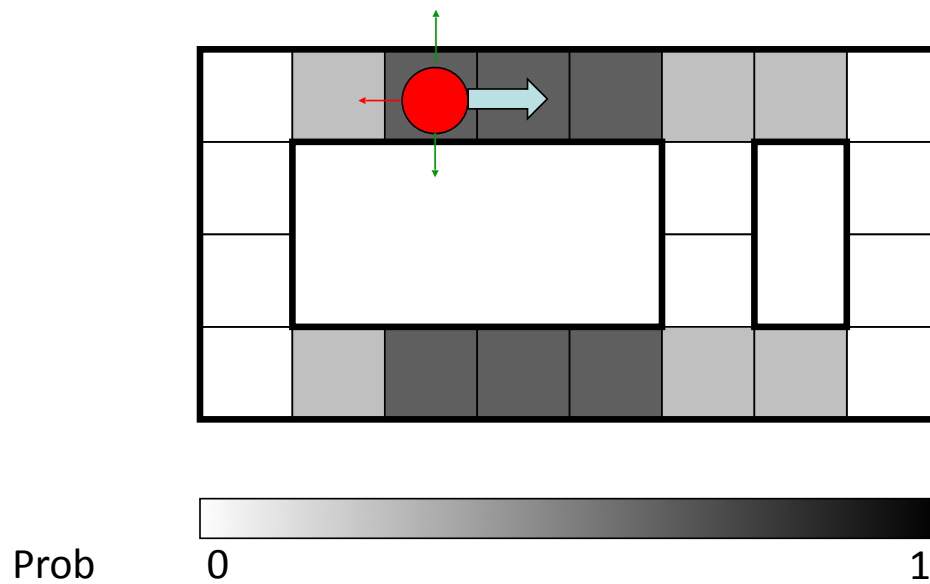
$t=1$

Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake

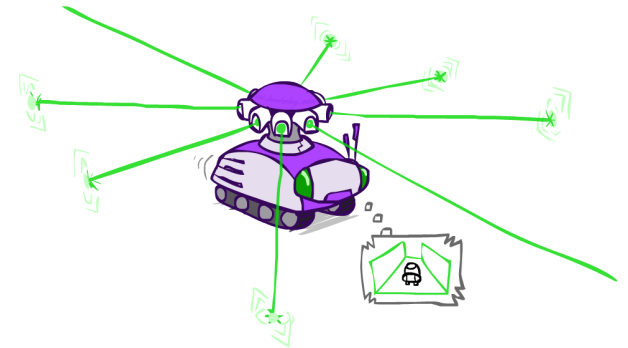




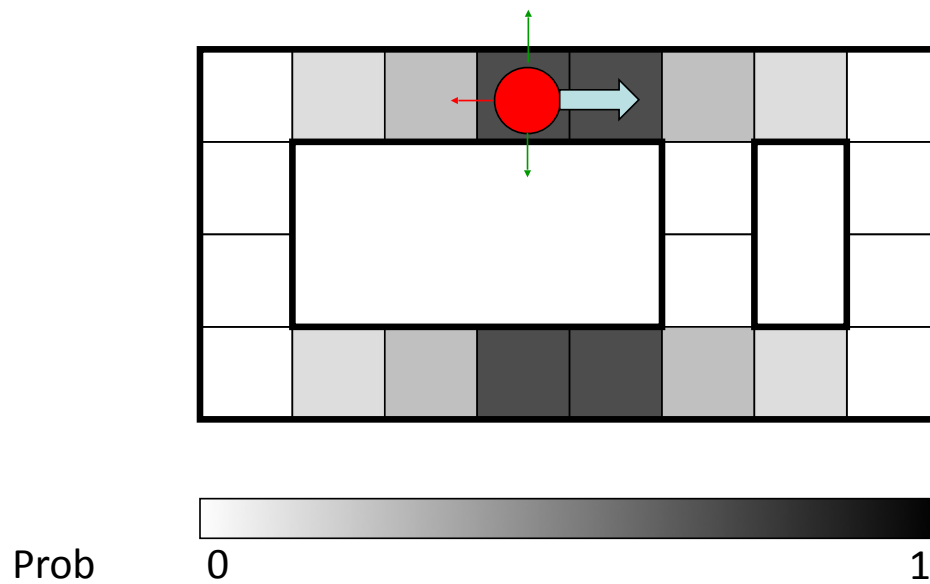
# Example: Robot Localization



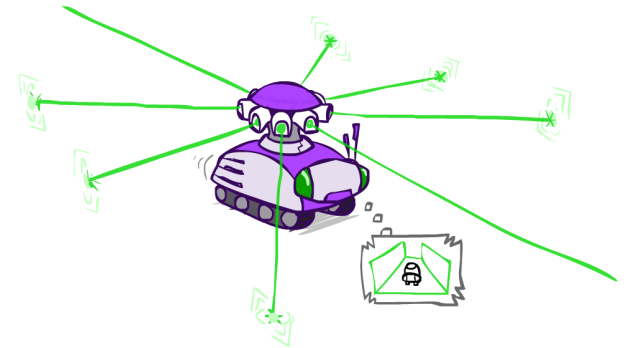
t=2



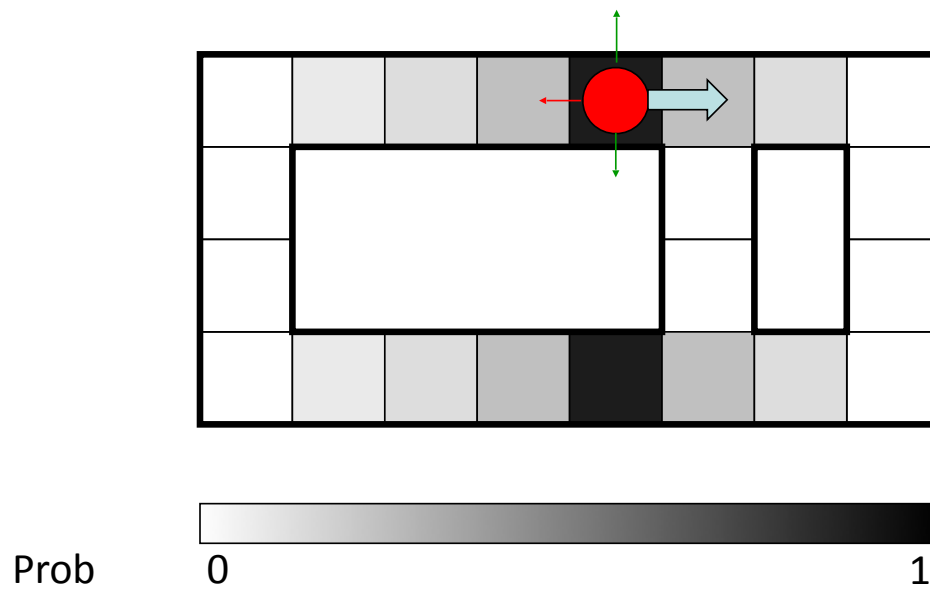
# Example: Robot Localization



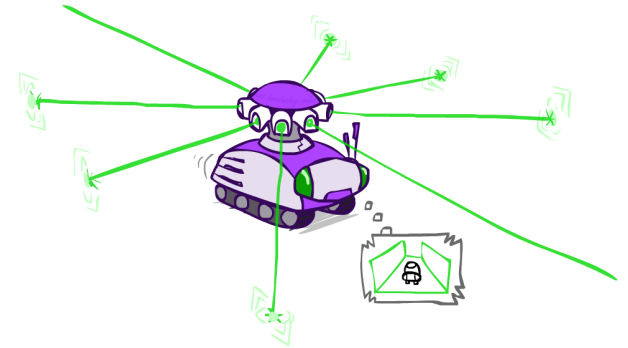
t=3



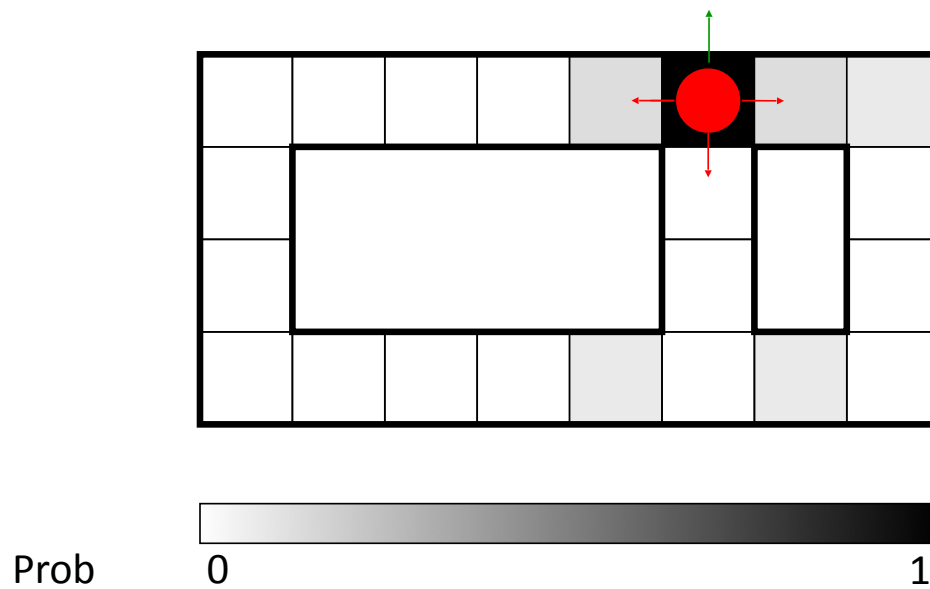
# Example: Robot Localization



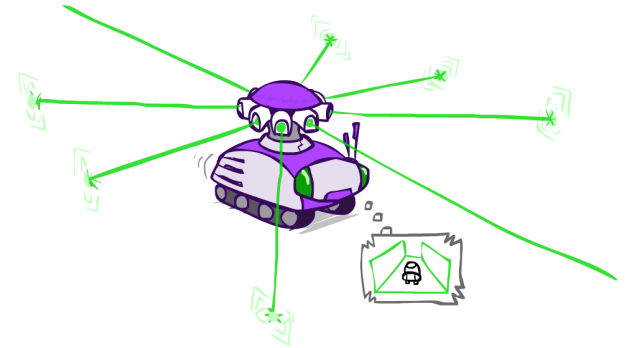
$t=4$



# Example: Robot Localization

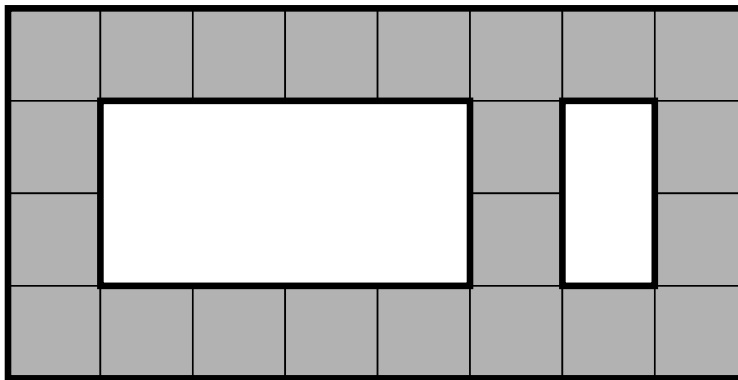


$t=5$



# Example: Robot Localization

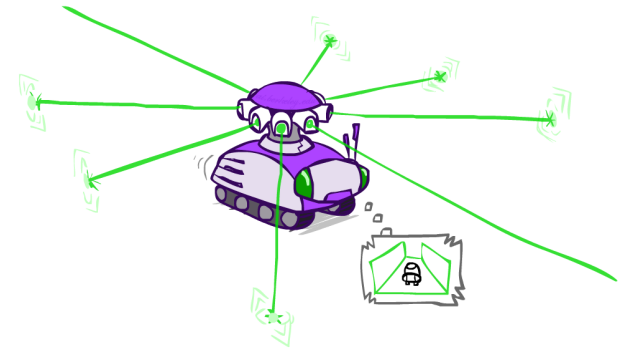
*Example from  
Michael Pfeiffer*



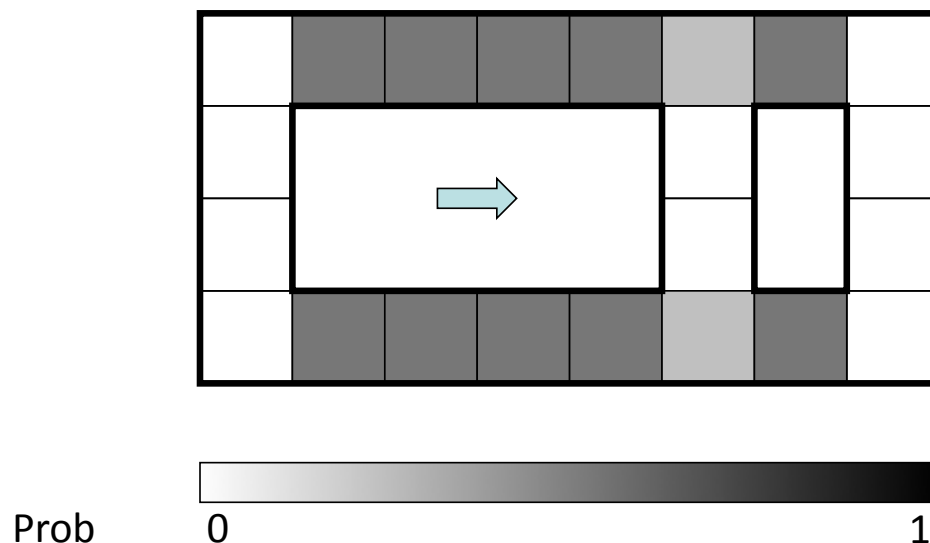
$t=0$

Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.

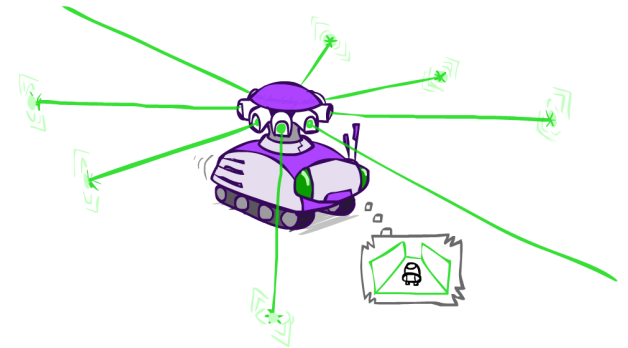


# Example: Robot Localization

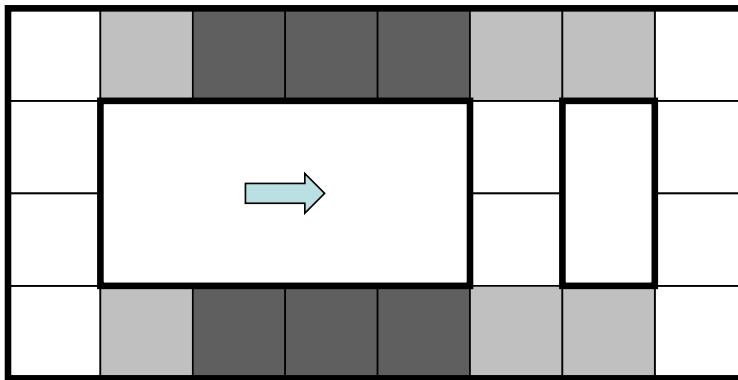


$t=1$

Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake



# Example: Robot Localization

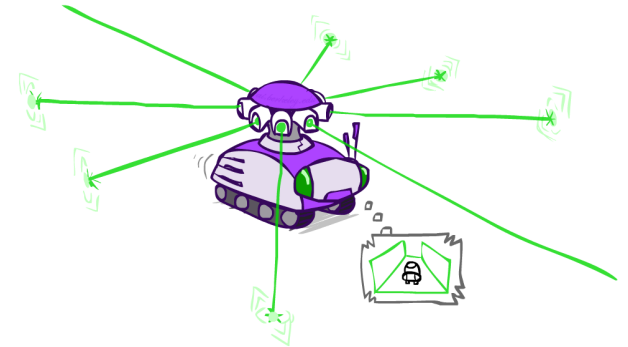


Prob

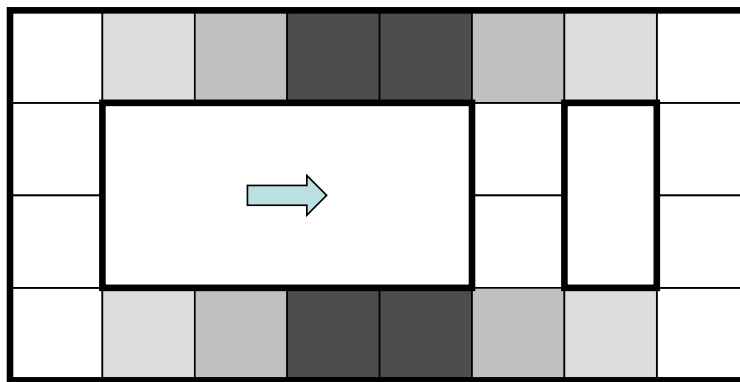
0

1

$t=2$



# Example: Robot Localization

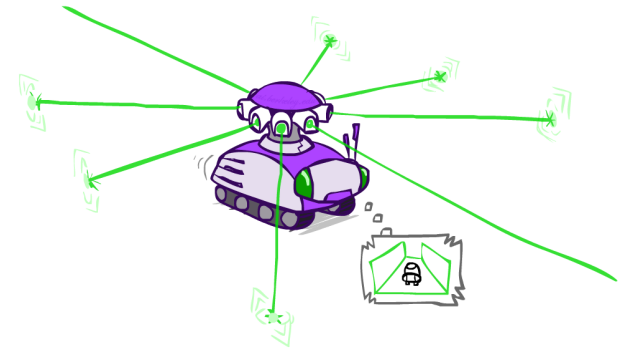


Prob

0

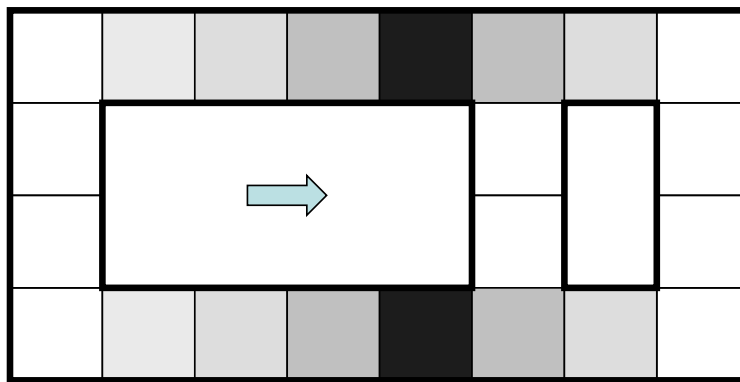
1

$t=3$

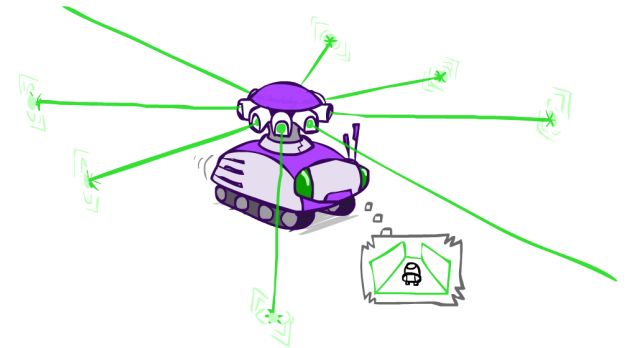




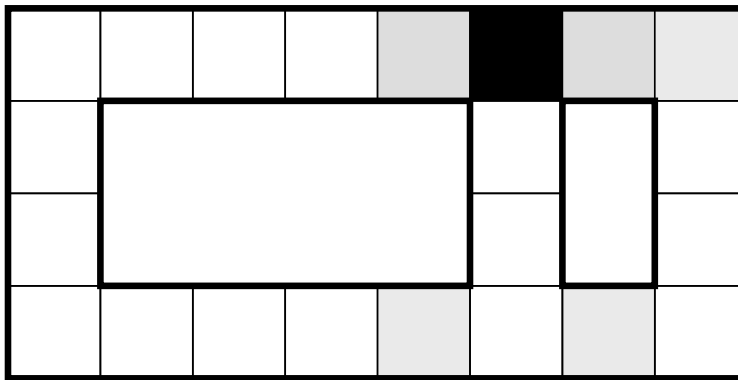
# Example: Robot Localization



$t=4$



# Example: Robot Localization

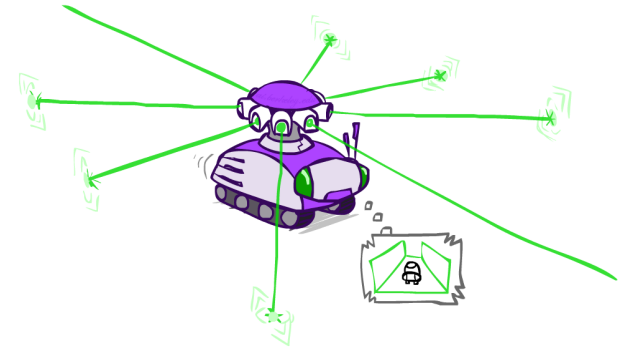


Prob

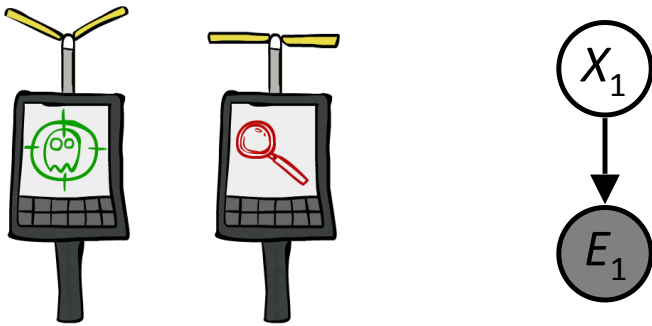
0

1

t=5

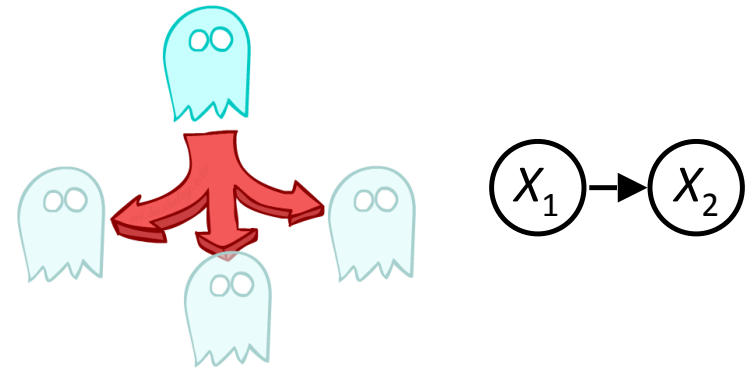


# Inference: Base Cases



$$P(X_1|e_1)$$

$$\begin{aligned} P(x_1|e_1) &= P(x_1, e_1)/P(e_1) \\ &\propto_{X_1} P(x_1, e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$



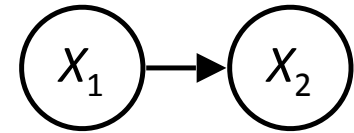
$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1, x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

# Passage of Time

- Assume we have current belief  $P(X \mid \text{evidence to date})$

$$B(X_t) = P(X_t | e_{1:t})$$



- Then, after one time step passes:

$$\begin{aligned} P(X_{t+1} | e_{1:t}) &= \sum_{x_t} P(X_{t+1}, x_t | e_{1:t}) \\ &= \sum_{x_t} P(X_{t+1} | x_t, e_{1:t}) P(x_t | e_{1:t}) \\ &= \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t}) \end{aligned}$$

- Or compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X' | x_t) B(x_t)$$

- Basic idea: beliefs get “pushed” through the transitions
  - With the “B” notation, we have to be careful about what time step  $t$  the belief is about, and what evidence it includes

# Example: Passage of Time

- As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

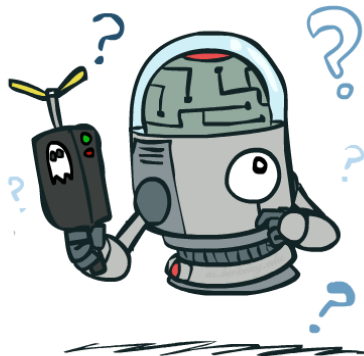
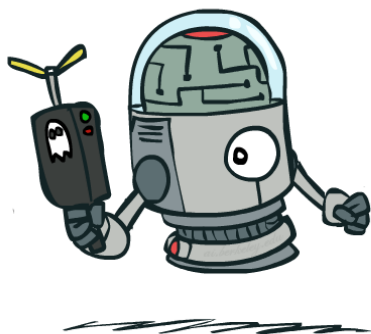
T = 1

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

T = 2

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

T = 5



# Observation

- Assume we have current belief  $P(X \mid \text{previous evidence})$ :

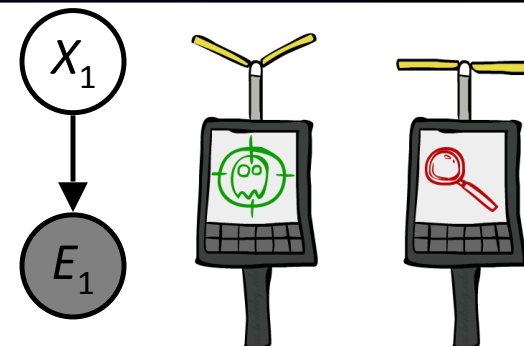
$$B'(X_{t+1}) = P(X_{t+1} | e_{1:t})$$

- Then, after evidence comes in:

$$\begin{aligned} P(X_{t+1} | e_{1:t+1}) &= P(X_{t+1}, e_{t+1} | e_{1:t}) / P(e_{t+1} | e_{1:t}) \\ &\propto_{X_{t+1}} P(X_{t+1}, e_{t+1} | e_{1:t}) \\ &= P(e_{t+1} | e_{1:t}, X_{t+1}) P(X_{t+1} | e_{1:t}) \\ &= P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t}) \end{aligned}$$

- Or, compactly:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1} | X_{t+1}) B'(X_{t+1})$$



- Basic idea: beliefs “reweighted” by likelihood of evidence
- Unlike passage of time, we have to renormalize

# Example: Observation

- As we get observations, beliefs get reweighted, uncertainty “decreases”

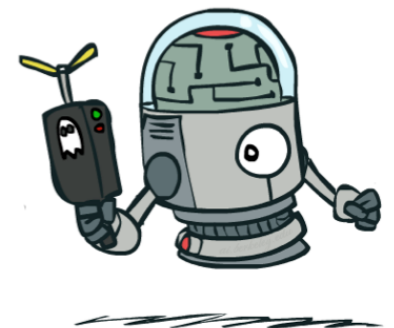
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Before observation

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

$$B(X) \propto P(e|X)B'(X)$$



# Example: Weather HMM

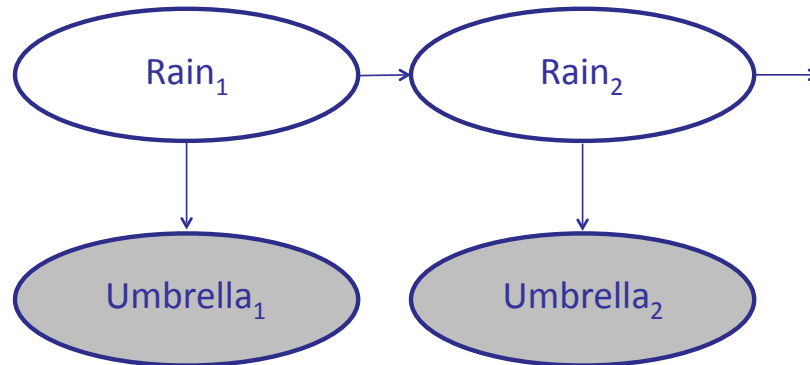


$$\begin{aligned} B'(+r) &= 0.5 \\ B'(-r) &= 0.5 \end{aligned}$$

$$\begin{aligned} B(+r) &= 0.818 \\ B(-r) &= 0.182 \end{aligned}$$

$$\begin{aligned} B'(+r) &= 0.627 \\ B'(-r) &= 0.373 \end{aligned}$$

$$\begin{aligned} B(+r) &= 0.883 \\ B(-r) &= 0.117 \end{aligned}$$





# The Forward Algorithm

- We are given evidence at each time and want to know

$$B_t(X) = P(X_t | e_{1:t})$$

- We can derive the following updates

$$\begin{aligned} P(x_t | e_{1:t}) &\propto_X P(x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\ &= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t | x_{t-1}) P(e_t | x_t) \\ &= P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}, e_{1:t-1}) \end{aligned}$$

We can normalize as we go if we want to have  $P(x|e)$  at each time step, or just once at the end...

# Online Belief Updates

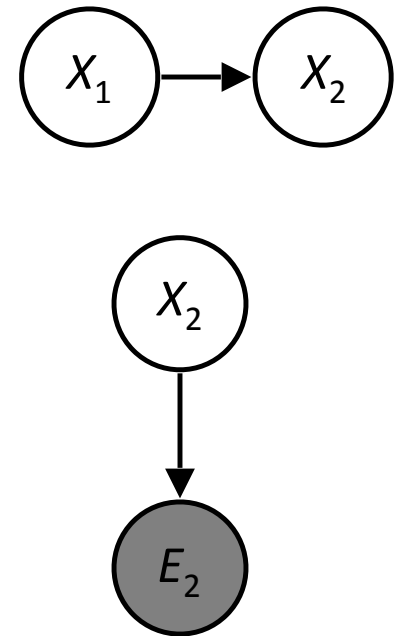
- Every time step, we start with current  $P(X \mid \text{evidence})$
- We update for time:

$$P(x_t | e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1} | e_{1:t-1}) \cdot P(x_t | x_{t-1})$$

- We update for evidence:

$$P(x_t | e_{1:t}) \propto_X P(x_t | e_{1:t-1}) \cdot P(e_t | x_t)$$

- The forward algorithm does both at once (and doesn't normalize)



# Pacman – Sonar (P5)

---

[Demo: Pacman – Sonar – No Beliefs(L14D1)]

# Next Time: Particle Filtering and Applications of HMMs

---