Artificial Intelligence

马尔可夫链

Hidden Markov Models

[These slides were created by Dan Klein, Pieter Abbeel, and Anca. http://ai.berkeley.edu.]

Reasoning over Time or Space

随着时间移动

- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring
- Need to introduce time (or space) into our models

Markov Models

- Value of X at a given time is called the state

(后面只依赖前面)

$$X_1 \to X_2 \to X_3 \to X_4 \dashrightarrow \cdots$$

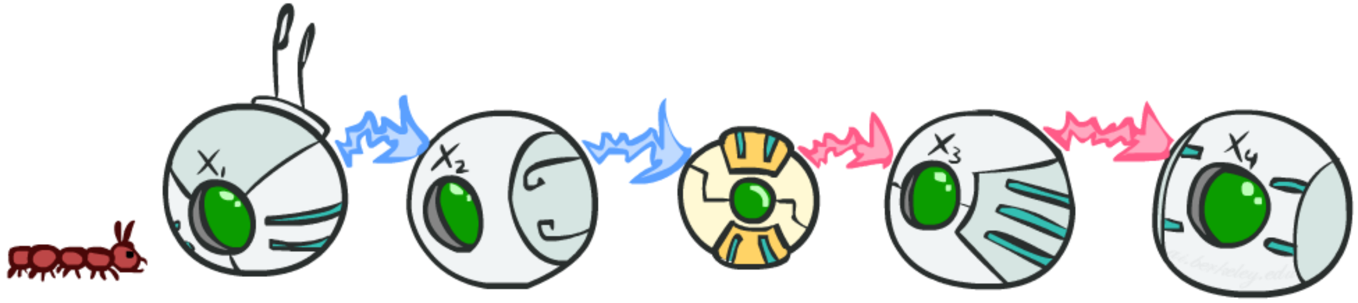
$$P(X_t) = ?$$

$$P(X_1) \qquad P(X_t|X_{t-1})$$

下一状态随时间变化 转化率固定

- Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial state probabilities)
- Stationarity assumption: transition probabilities the same at all times
- Same as MDP transition model, but no choice of action
- A (growable) BN: We can always use generic BN reasoning on it if we truncate the chain at a fixed length

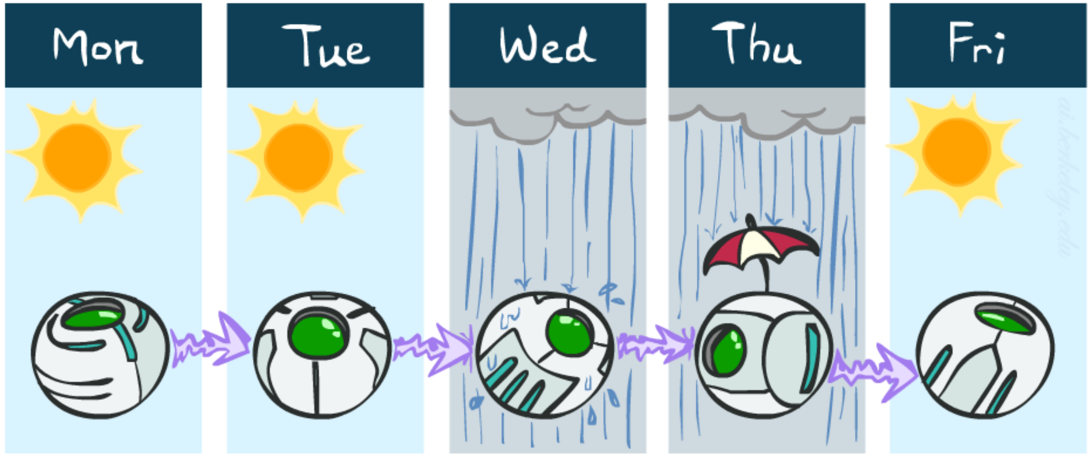
Markov Assumption: Conditional Independence



- Basic conditional independence:
  - Past and future independent given the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property

Example Markov Chain: Weather

- States: X = {rain, sun}

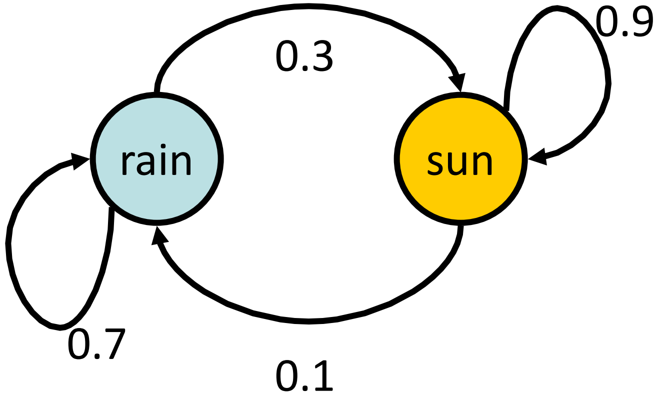


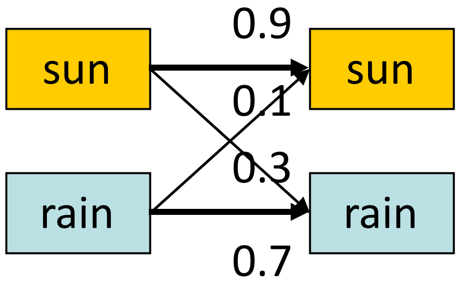
- Initial distribution: 1.0 sun

- CPT $P(X_t \mid X_{t-1})$:

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

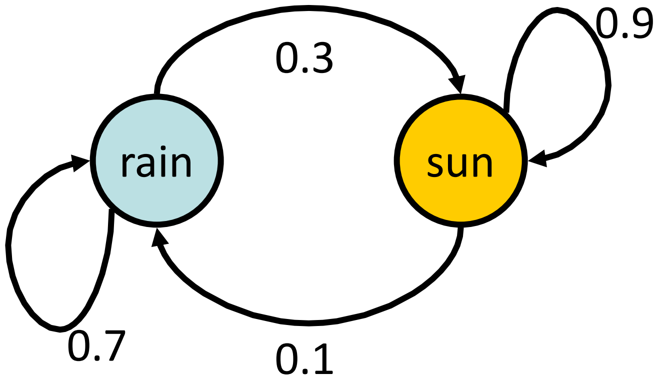
Two new ways of representing the same CPT





Example Markov Chain: Weather

- Initial distribution: 1.0 sun



- What is the probability distribution after one step?

$$P(X_2 = sun) = \sum_{x_1} P(x_1, X_2 = sun) = \sum_{x_1} P(X_2 = sun|x_1)P(x_1)$$

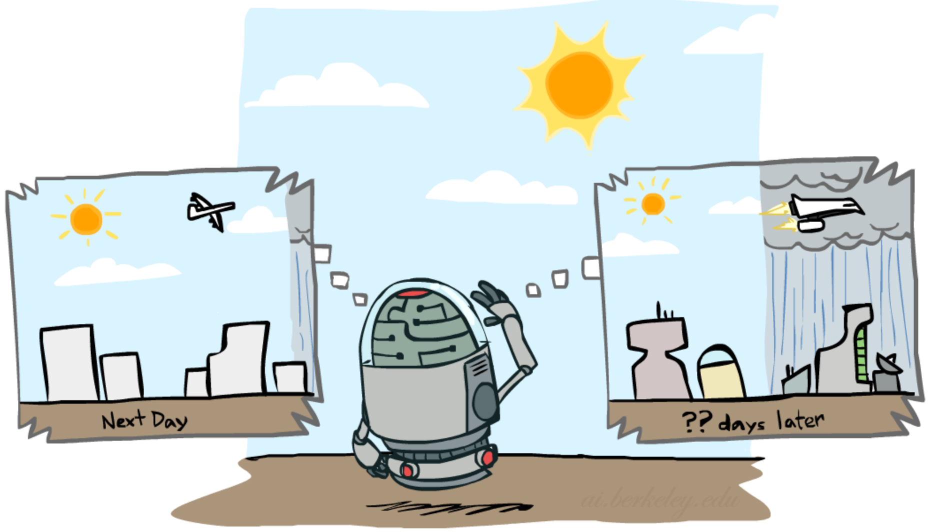
$$P(X_2 = \text{sun}) = \quad P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.3 \cdot 0.0 = 0.9$$

Mini-Forward Algorithm

- Question: What's P(X) on some day t?

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$



$$P(x_1) = \text{known}$$

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$

$$= \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1})$$

Forward simulation

Example Run of Mini-Forward Algorithm

- From initial observation of sun

$$\underset{P(X_1)}{\left\langle \begin{matrix}1.0\\0.0\end{matrix}\right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{matrix}0.9\\0.1\end{matrix}\right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{matrix}0.84\\0.16\end{matrix}\right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{matrix}0.804\\0.196\end{matrix}\right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix}0.75\\0.25\end{matrix}\right\rangle}$$

0.1, 0.3

- From initial observation of rain

$$\underset{P(X_1)}{\left\langle \begin{matrix}0.0\\1.0\end{matrix}\right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{matrix}0.3\\0.7\end{matrix}\right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{matrix}0.48\\0.52\end{matrix}\right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{matrix}0.588\\0.412\end{matrix}\right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix}0.75\\0.25\end{matrix}\right\rangle}$$

- From yet another initial distribution $P(X_1)$:

$$\underset{P(X_1)}{\left\langle \begin{matrix}p\\1-p\end{matrix}\right\rangle} \quad \cdots \quad \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix}0.75\\0.25\end{matrix}\right\rangle}$$

[Demo: L13D1,2,3]

Video of Demo Ghostbusters Basic Dynamics

Video of Demo Ghostbusters Circular Dynamics

Video of Demo Ghostbusters Whirlpool Dynamics

Stationary Distributions

- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution

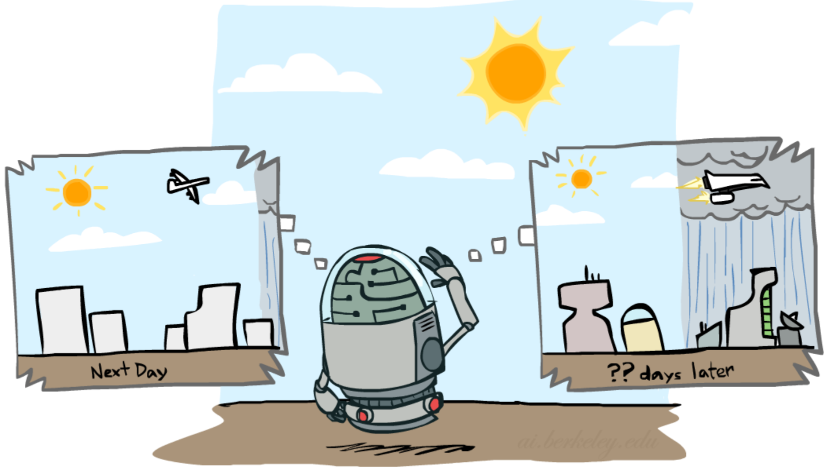
- Stationary distribution:
  - The distribution we end up with is called the stationary distribution P_∞ of the chain
  - It satisfies

$$P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x)P_\infty(x)$$

Example: Stationary Distributions

- Question: What's P(X) at time t = infinity?

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$



$$P_\infty(sun) = P(sun|sun)P_\infty(sun) + P(sun|rain)P_\infty(rain)$$
$$P_\infty(rain) = P(rain|sun)P_\infty(sun) + P(rain|rain)P_\infty(rain)$$

$$P_\infty(sun) = 0.9P_\infty(sun) + 0.3P_\infty(rain)$$
$$P_\infty(rain) = 0.1P_\infty(sun) + 0.7P_\infty(rain)$$

$$P_\infty(sun) = 3P_\infty(rain)$$
$$P_\infty(rain) = 1/3P_\infty(sun)$$

0.3

0.

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

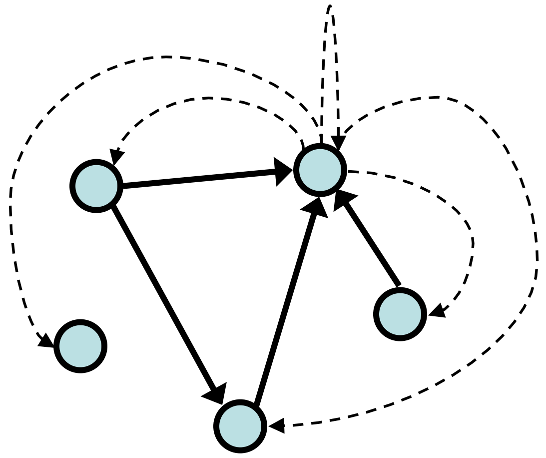
Also: $P_\infty(sun) + P_\infty(rain) = 1$

$\Rightarrow$

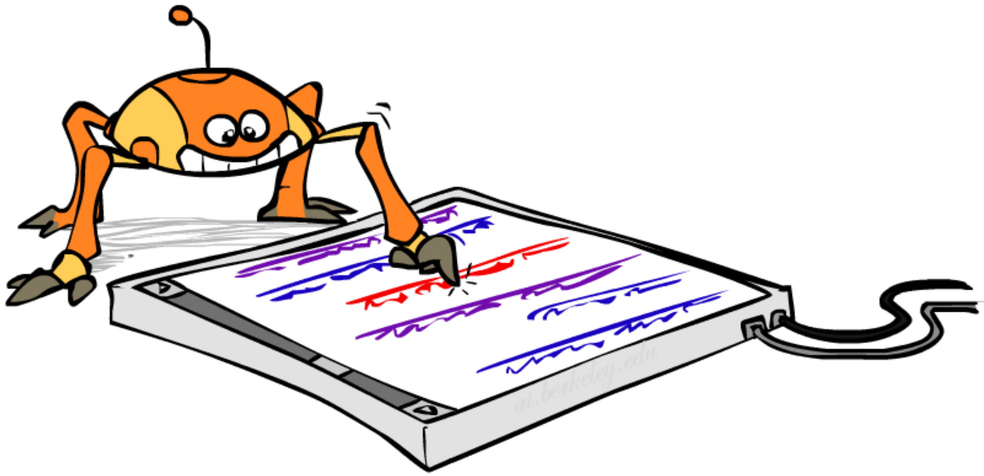
$$P_\infty(sun) = 3/4$$
$$P_\infty(rain) = 1/4$$

Application of Stationary Distribution: Web Link Analysis

- PageRank over a web graph
  - Each web page is a possible value of a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (dotted lines, not all shown)
    - With prob. 1-c, follow a random outlink (solid lines)



- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page
  - Somewhat robust to link spam.
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors (rank actually getting less important over time)



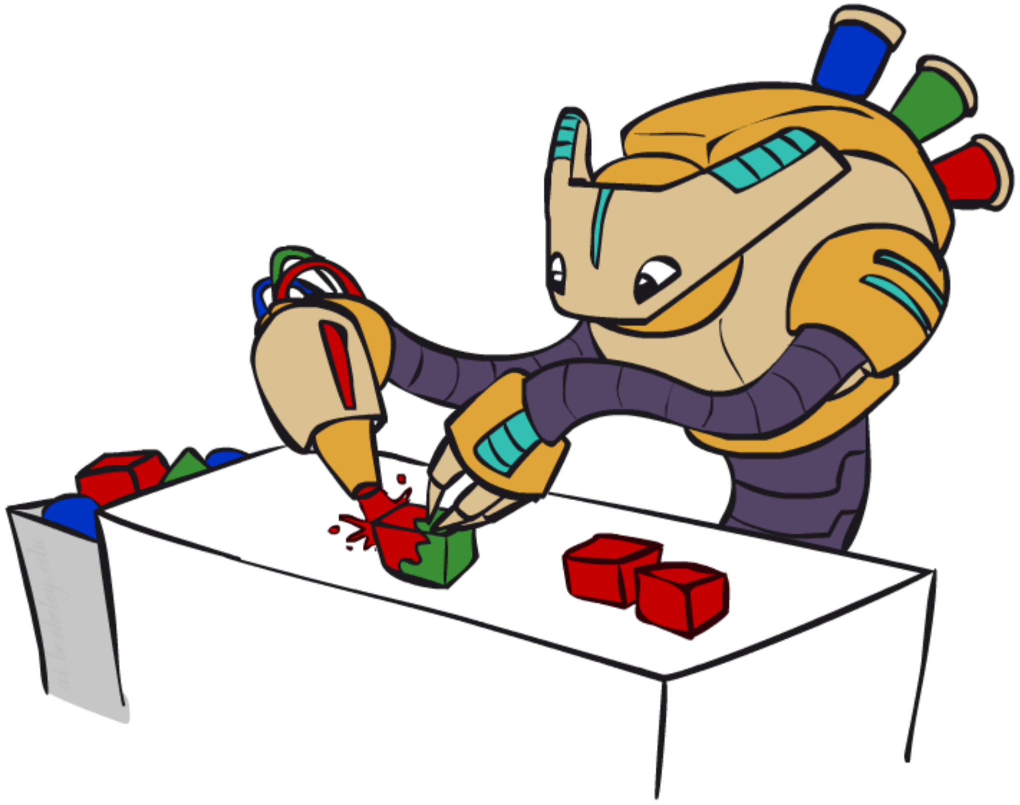
Application of Stationary Distributions: Gibbs Sampling*

- Each joint instantiation over all hidden and query variables is a state: $\{X_1, \ldots, X_n\} = H \cup Q$

- Transitions:
  - With probability 1/n resample variable X_j according to

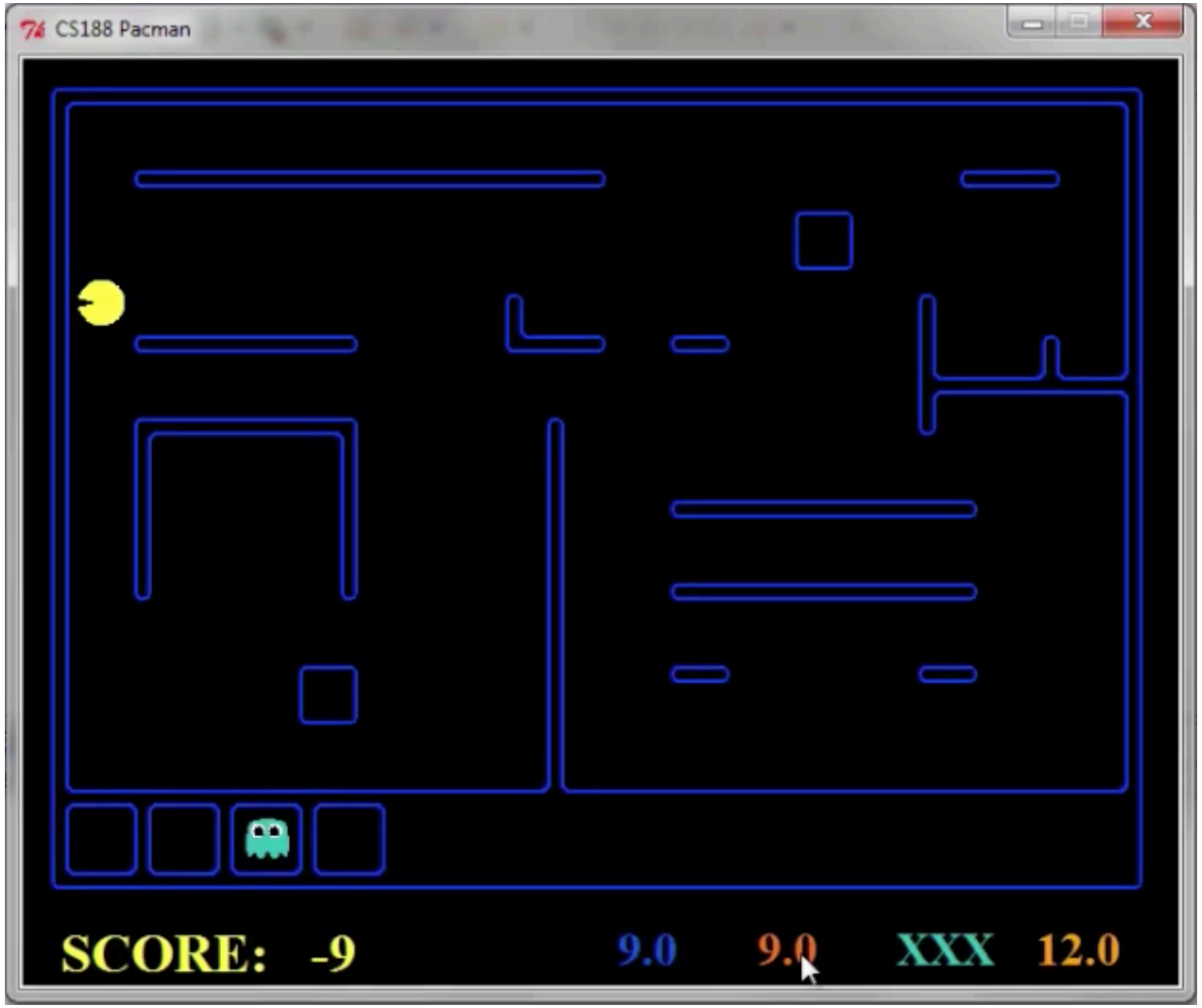
    $$P(X_j \mid x_1, x_2, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, e_1, \ldots, e_m)$$

- Stationary distribution:
  - Conditional distribution $P(X_1, X_2, \ldots, X_n \mid e_1, \ldots, e_m)$
  - Means that when running Gibbs sampling long enough we get a sample from the desired distribution
  - Requires some proof to show this is true!



Hidden Markov Models

Pacman – Sonar



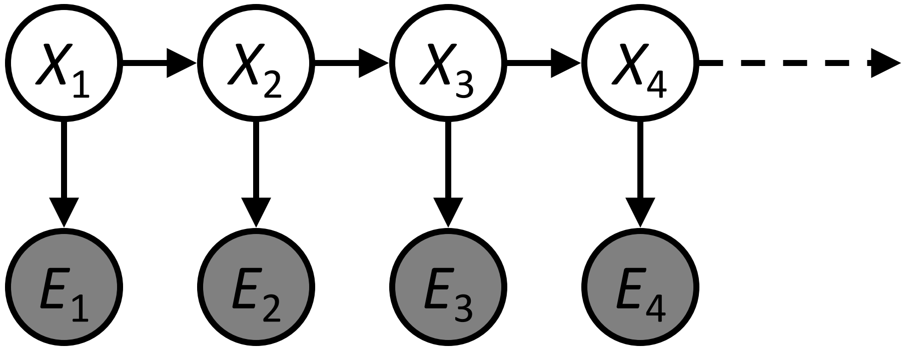
[Demo: Pacman – Sonar – No Beliefs(L14D1)]

Video of Demo Pacman – Sonar (no beliefs)

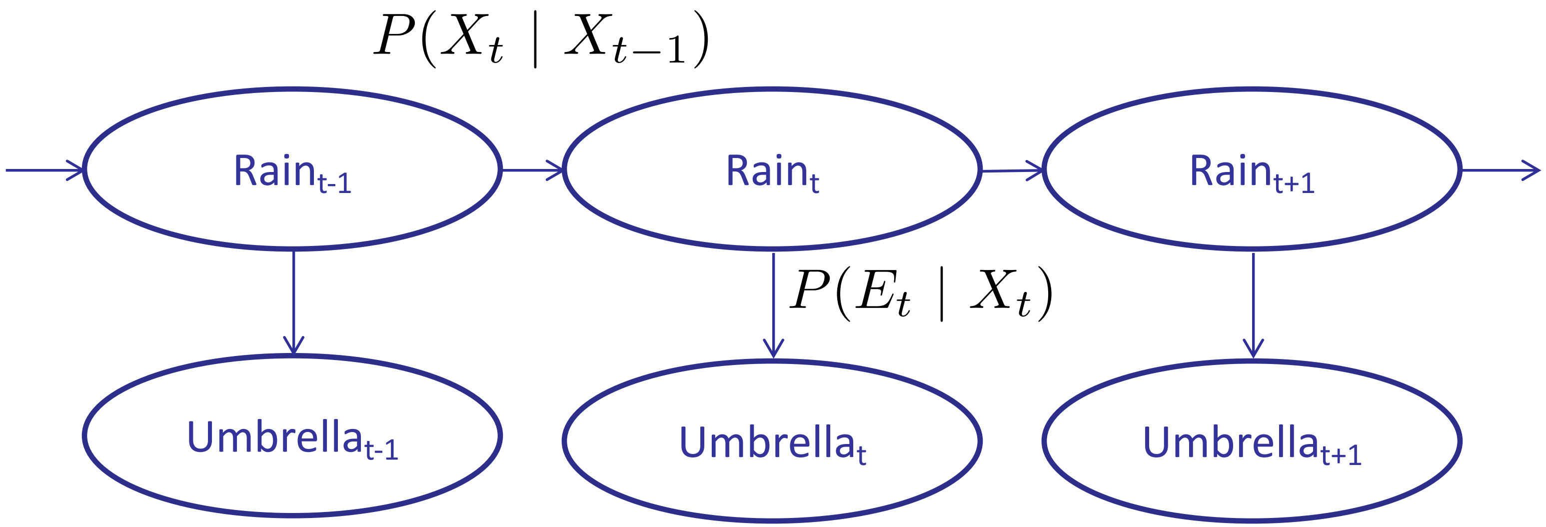
Hidden Markov Models

- 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: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

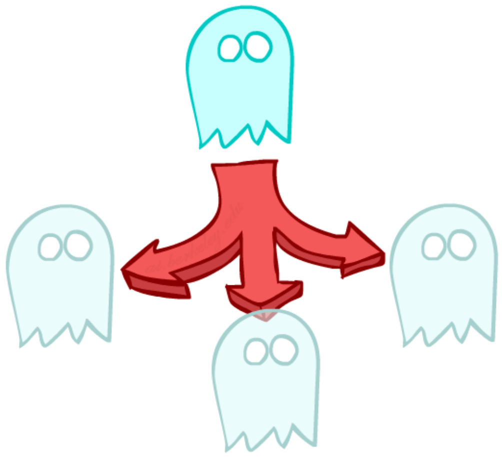
R_{t-1}	R_t	$P(R_t \mid R_{t-1})$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

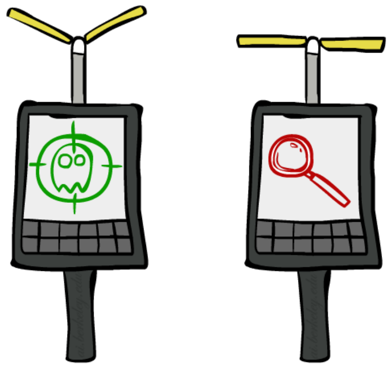
R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

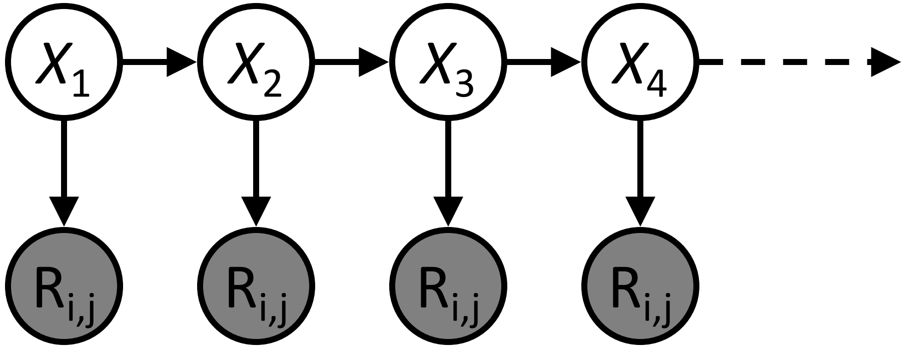
Video of Demo Ghostbusters – Circular Dynamics -- HMM

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

$P(X_1)$

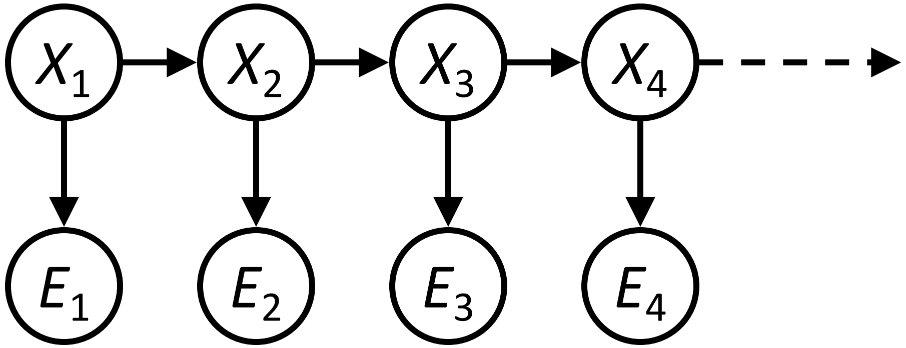
1/6	1/6	1/2
0	1/6	0
0	0	0

$P(X|X'=\langle 1,2\rangle)$

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

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present
  - Current observation independent of all else given current state



- Does this mean that evidence variables are guaranteed to be independent?
  - [No, they tend to correlated by the hidden state]

Real HMM Examples

- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

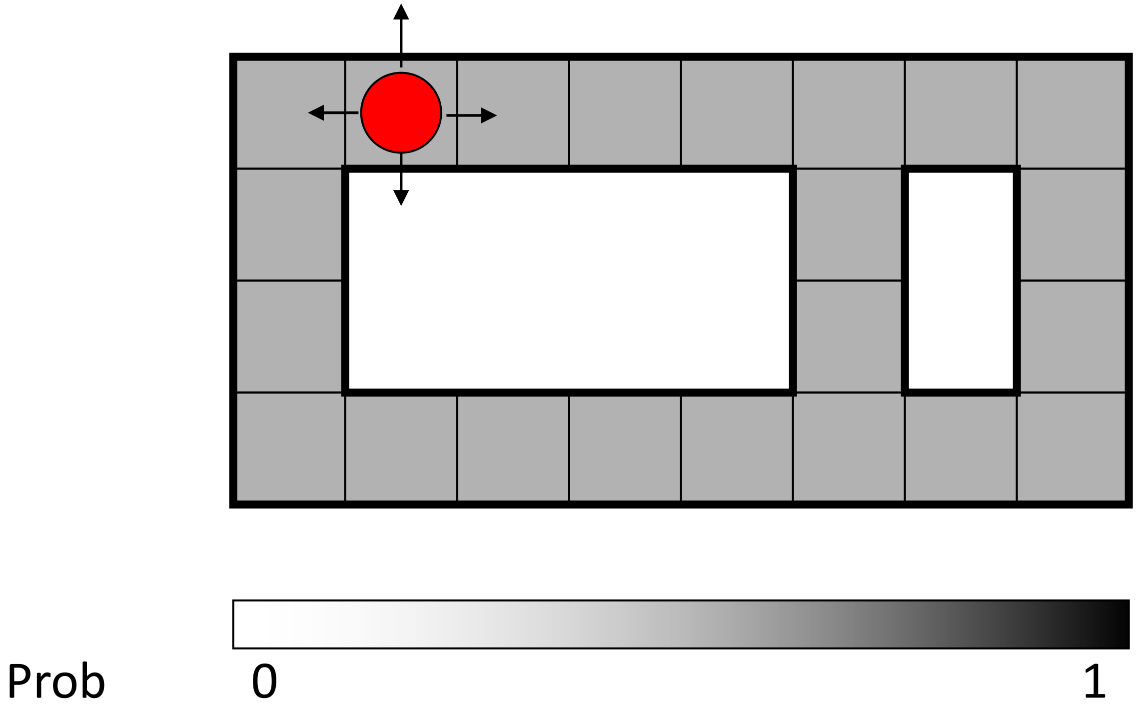
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options

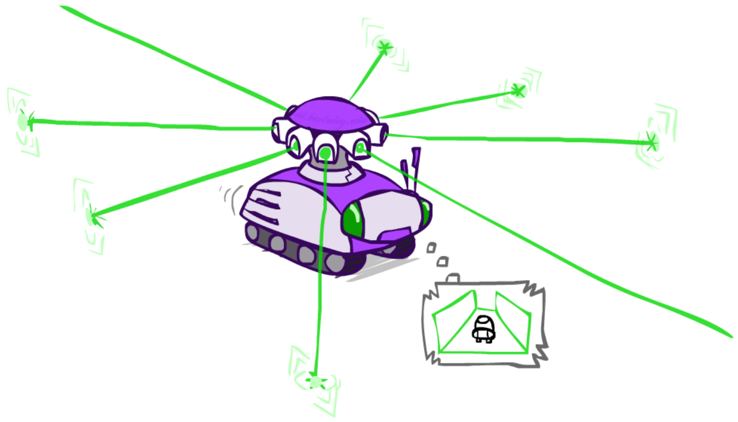
Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P_t(X_t \mid e_1, \ldots, e_t)$ (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



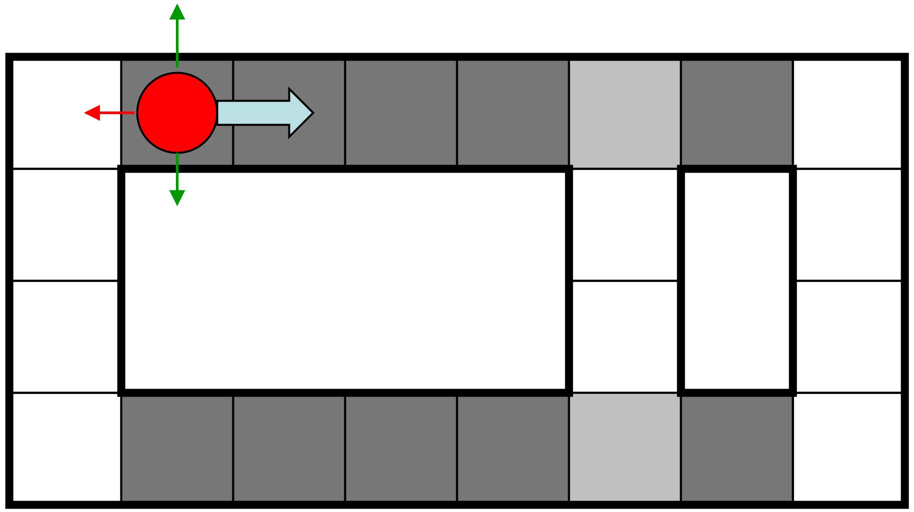


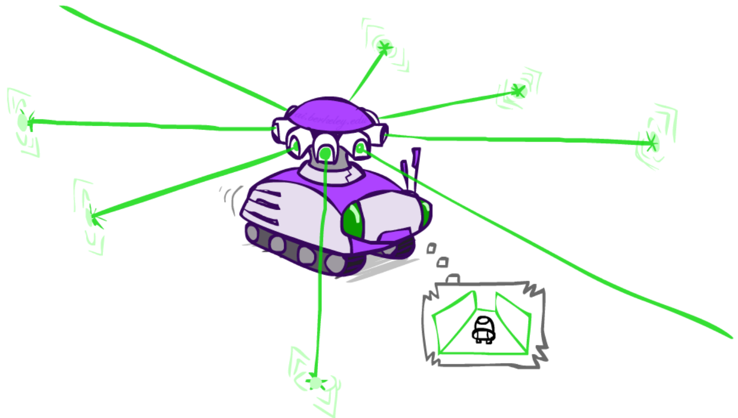
t=0

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



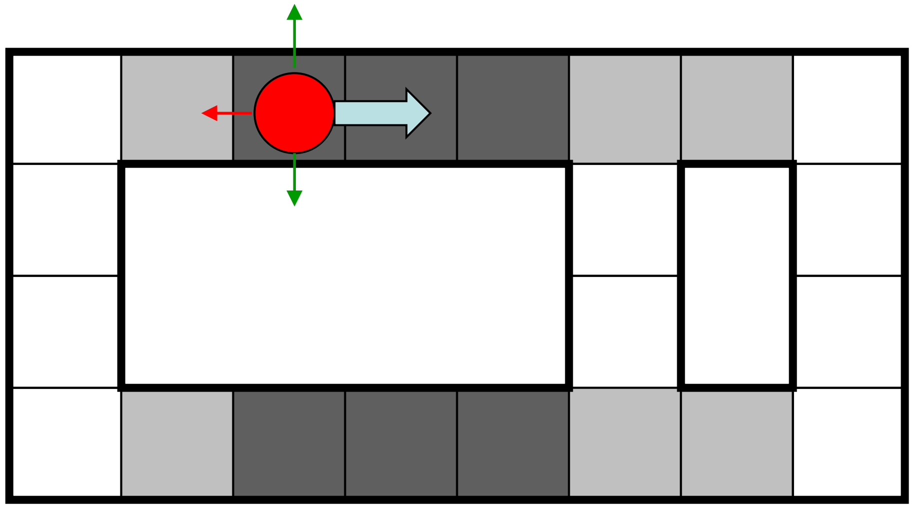


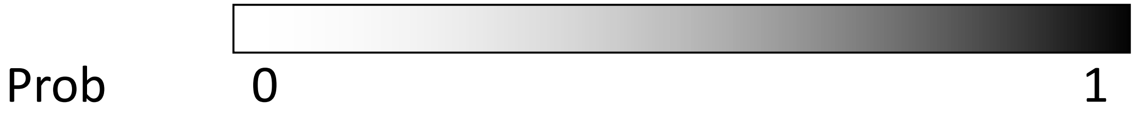
Prob 0 1

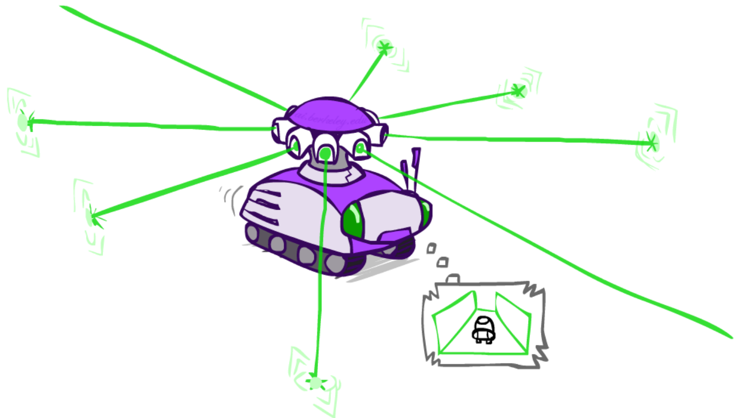
t=1

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

Example: Robot Localization

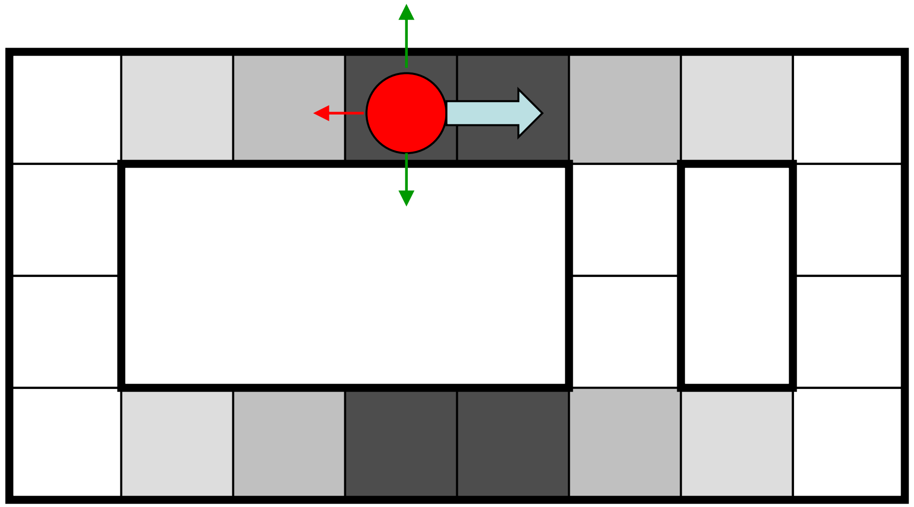


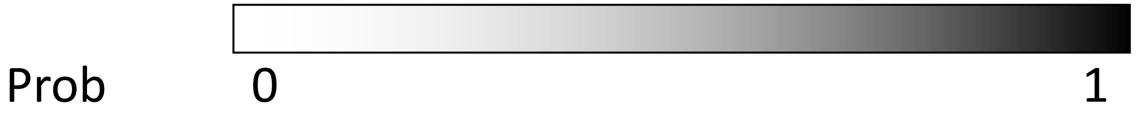


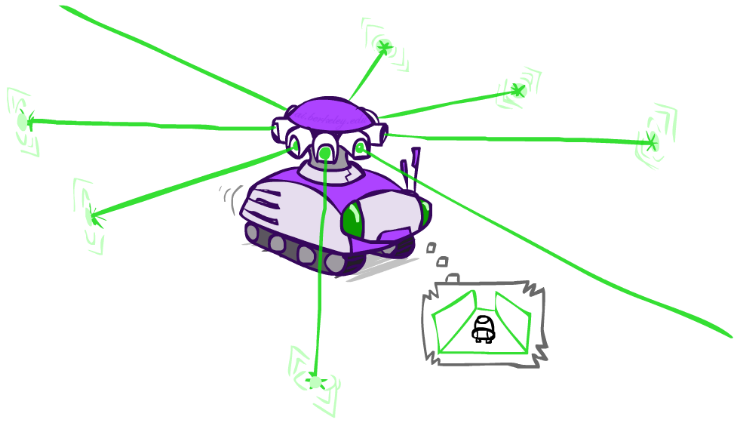


t=2

Example: Robot Localization

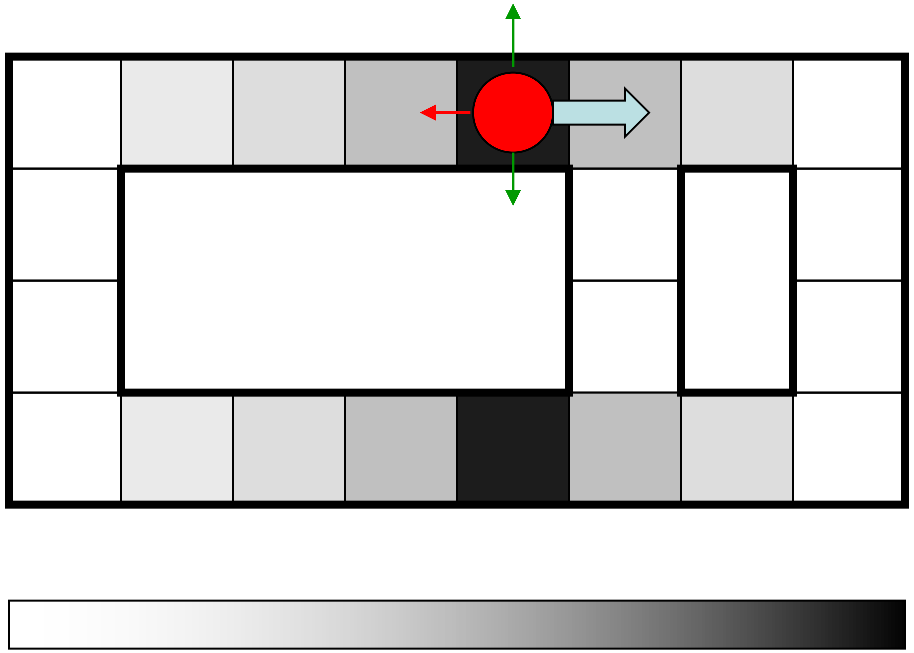




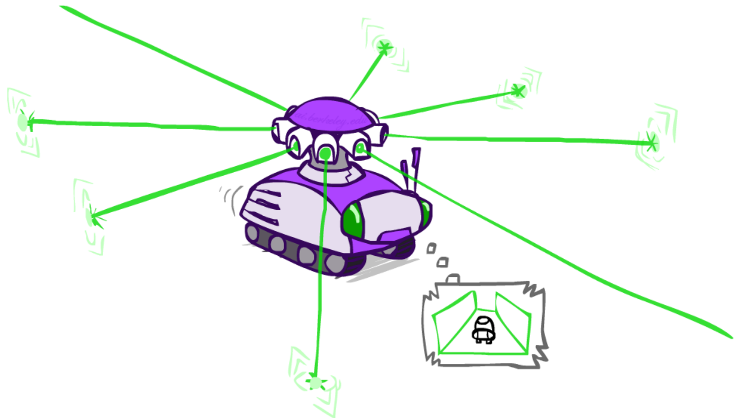


t=3

Example: Robot Localization

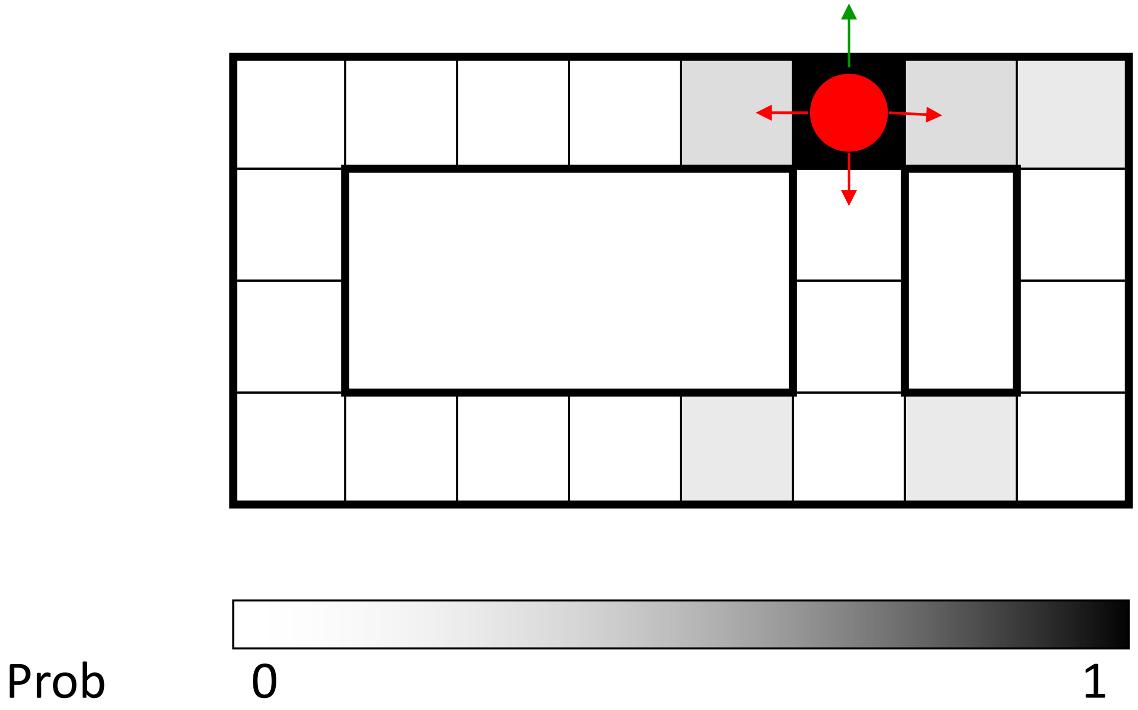


Prob 0 1

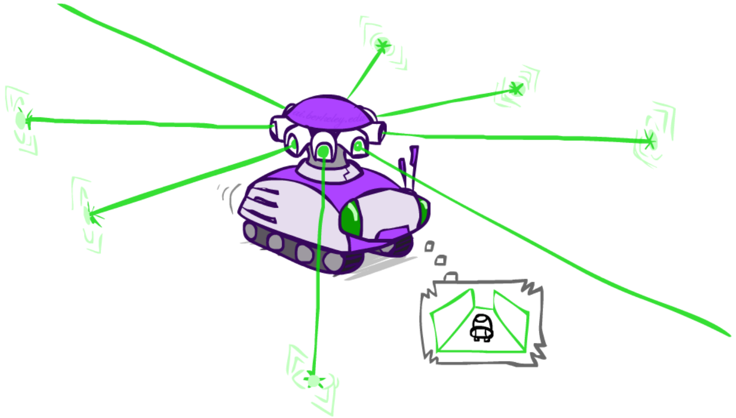


t=4

Example: Robot Localization



t=5



Inference: Find State Given Evidence

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

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

- Idea: start with $P(X_1)$ and derive B_t in terms of B_{t-1}
  - equivalently, derive B_{t+1} in terms of B_t

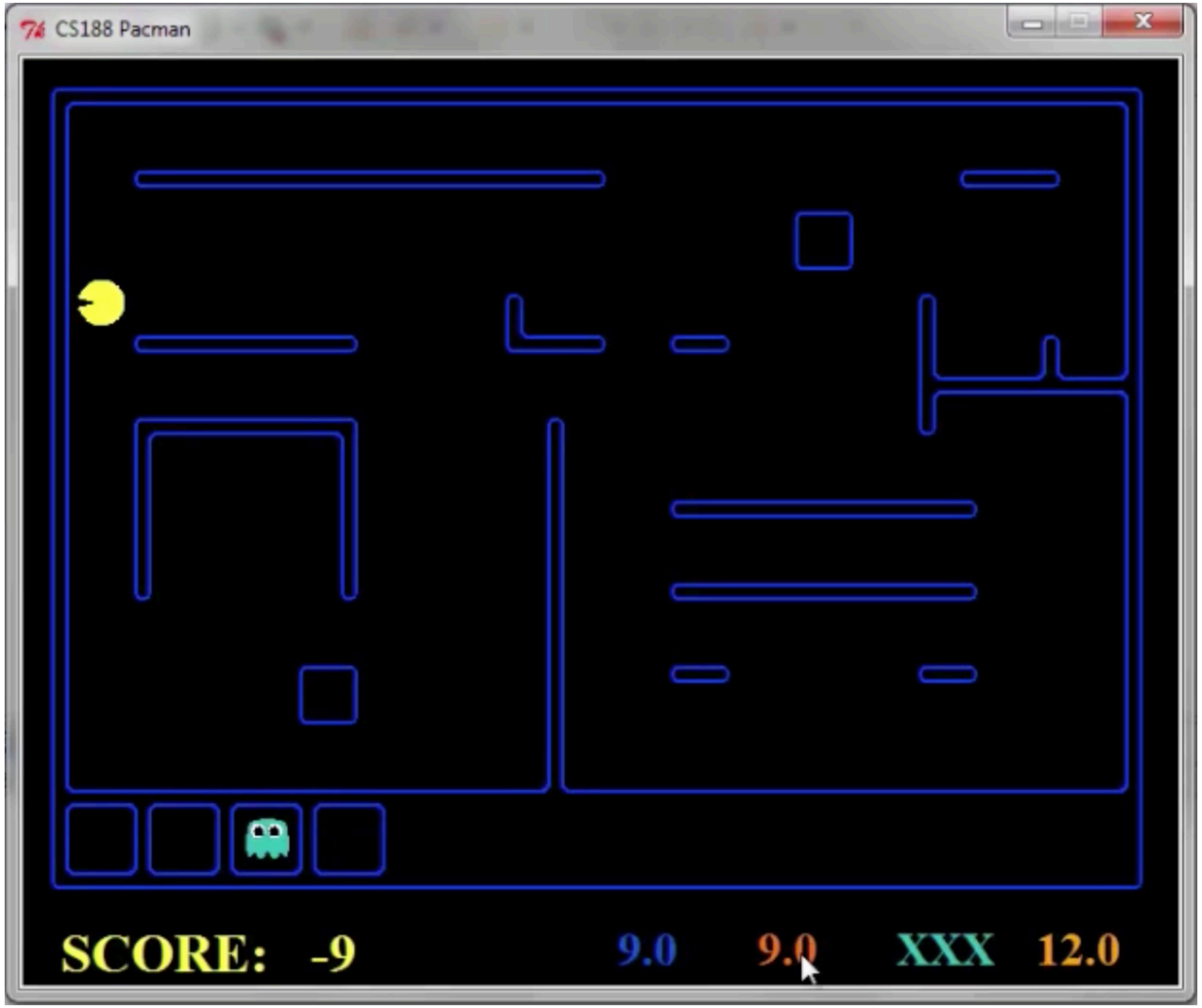
Artificial Intelligence

Filtering

[These slides were created by Dan Klein, Pieter Abbeel, and Anca. http://ai.berkeley.edu.]

Hidden Markov Models

Pacman – Sonar

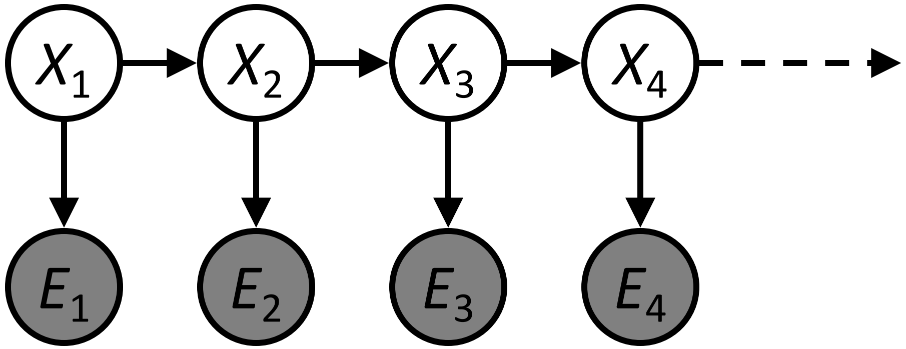


[Demo: Pacman – Sonar – No Beliefs(L14D1)]

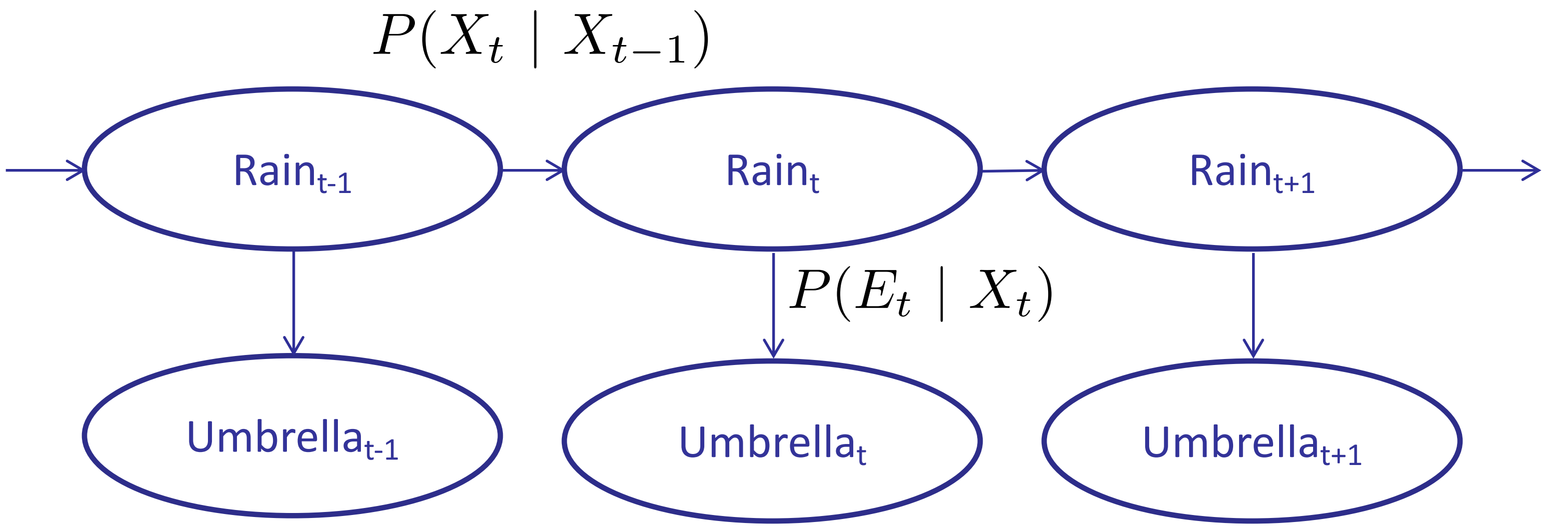
Video of Demo Pacman – Sonar (no beliefs)

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: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

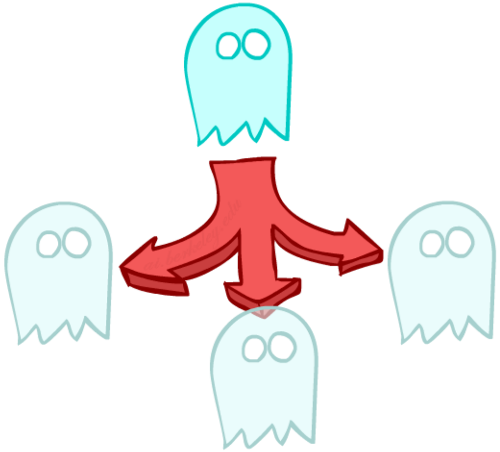
R_{t-1}	R_t	$P(R_t \mid R_{t-1})$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

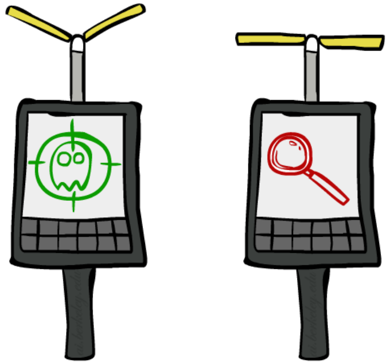
R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

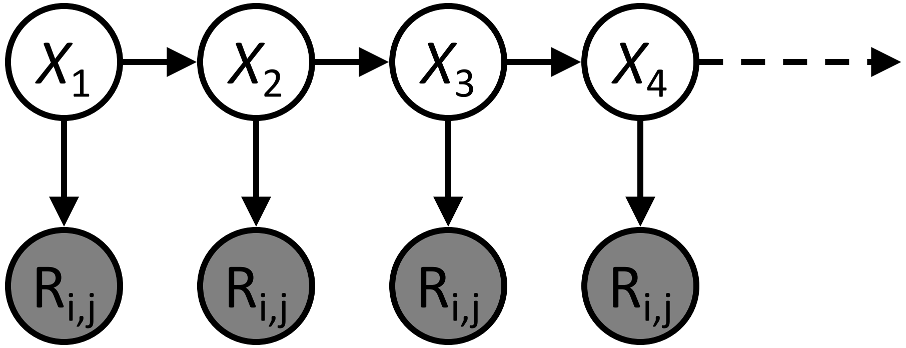
Video of Demo Ghostbusters – Circular Dynamics -- HMM

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

$P(X_1)$

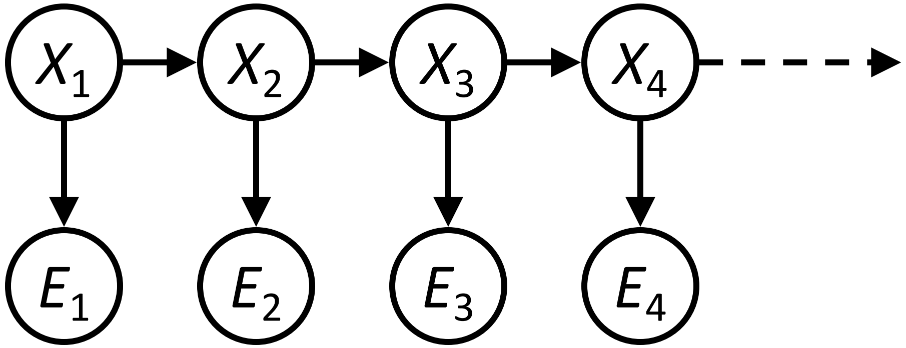
1/6	1/6	1/2
0	1/6	0
0	0	0

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

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

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present
  - Current observation independent of all else given current state



- Does this mean that evidence variables are guaranteed to be independent?
  - [No, they tend to correlated by the hidden state]

Real HMM Examples

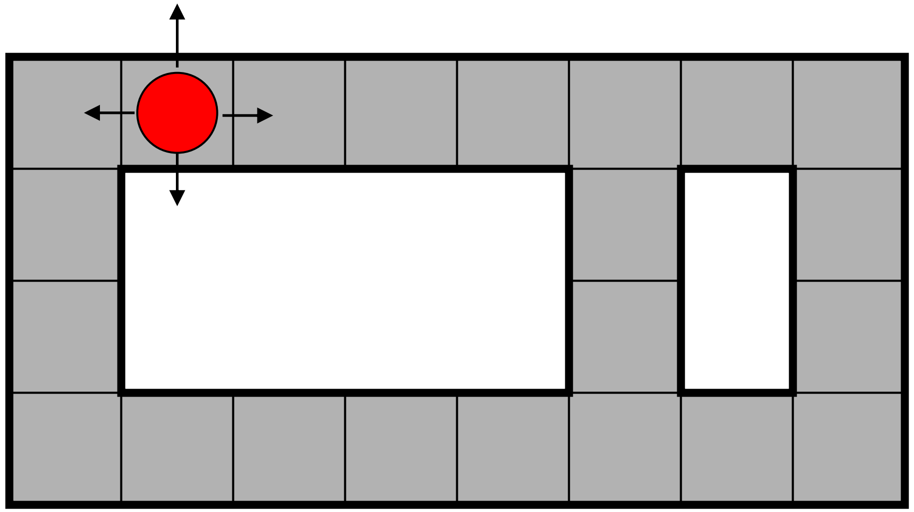
- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)
- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs: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

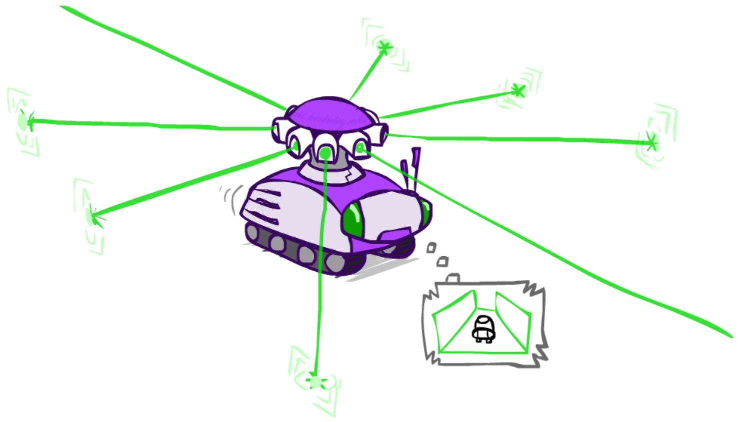
Filtering / Monitoring

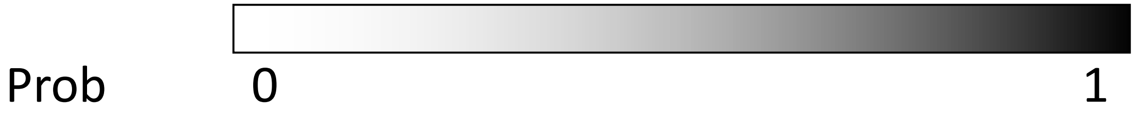
- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P_t(X_t \mid e_1, \ldots, e_t)$ (the belief state) over time
- We start with $B_1(X)$ in an initial setting, usually uniform
- As time passes, or we get observations, we update $B(X)$
- The Kalman filter was invented in the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

Example: Robot Localization

Example from Michael Pfeiffer





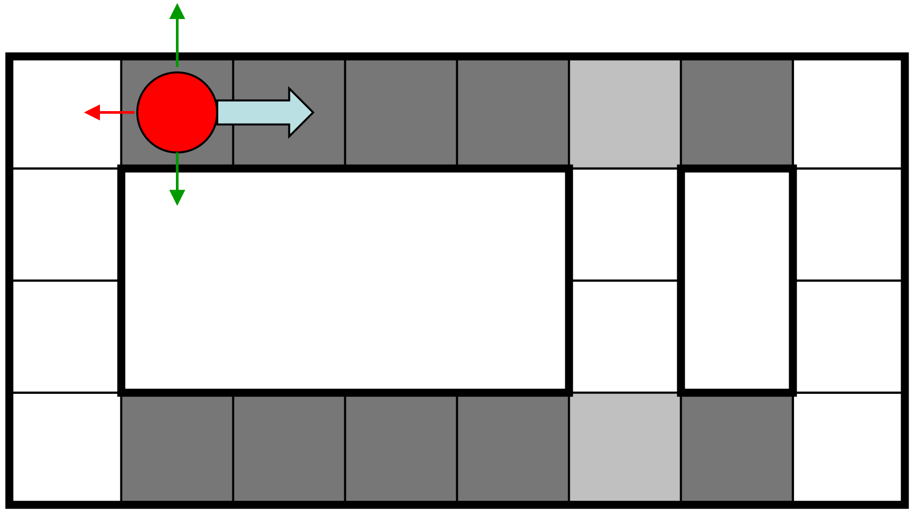


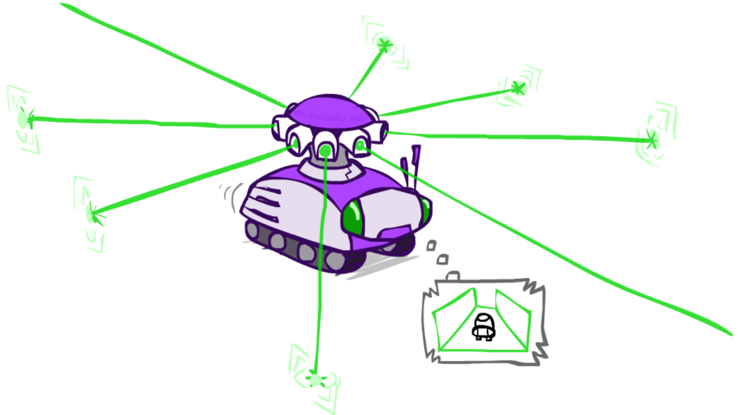
t=0

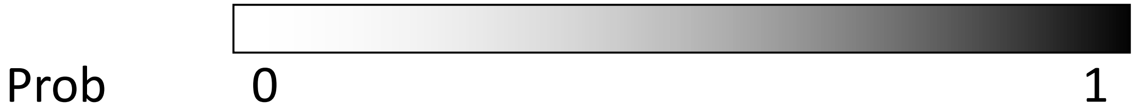
Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.

Example: Robot Localization



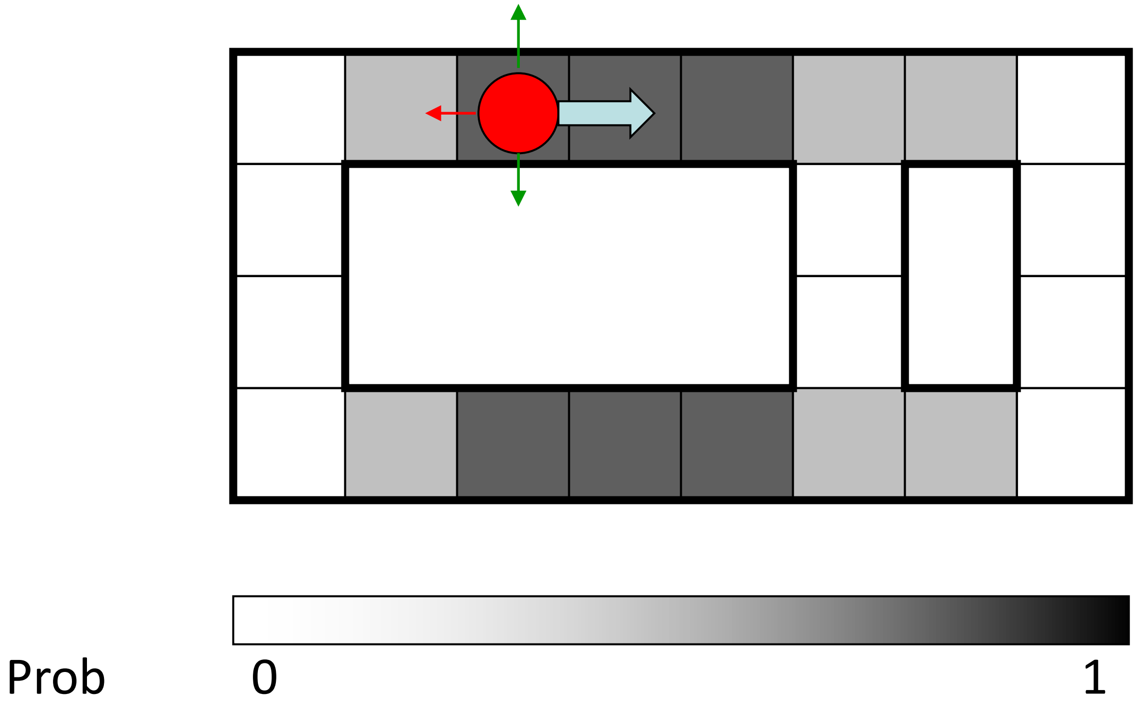


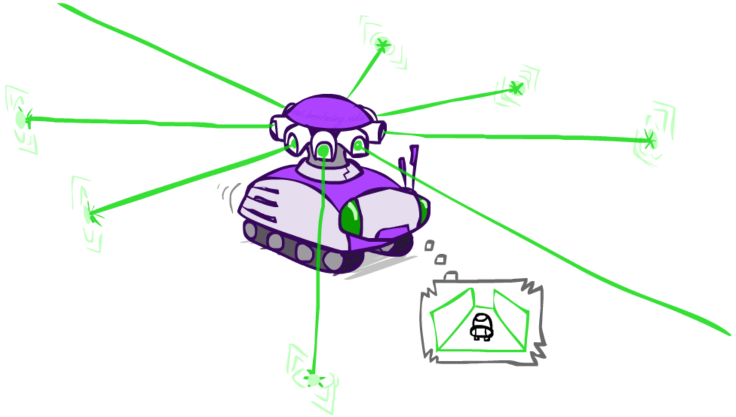


t=1

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

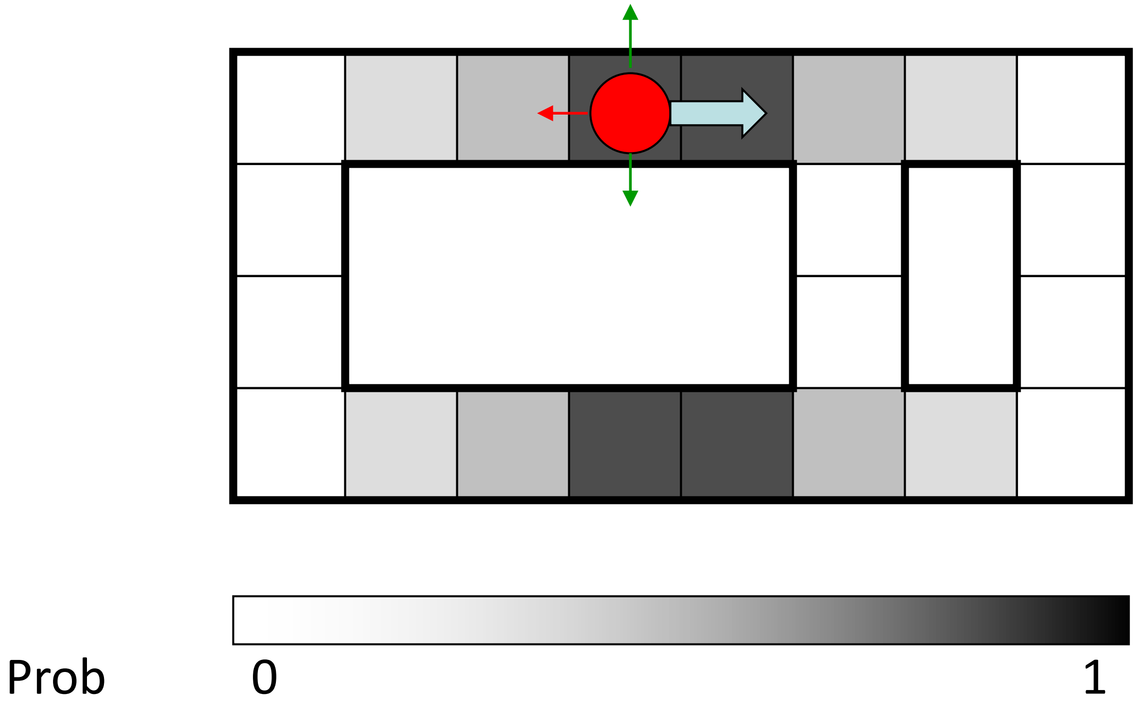
Example: Robot Localization

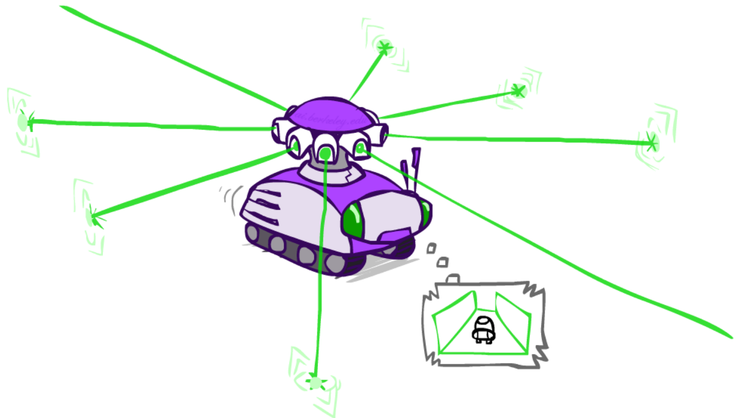




t=2

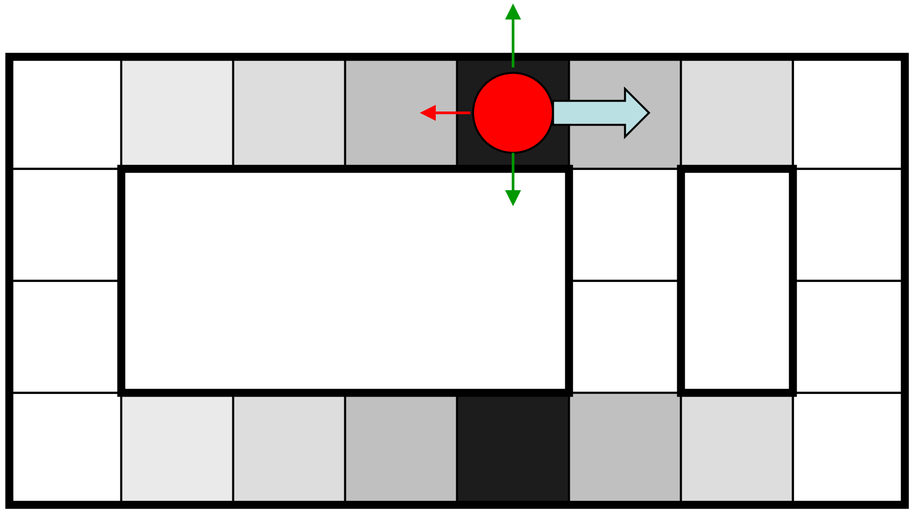
Example: Robot Localization

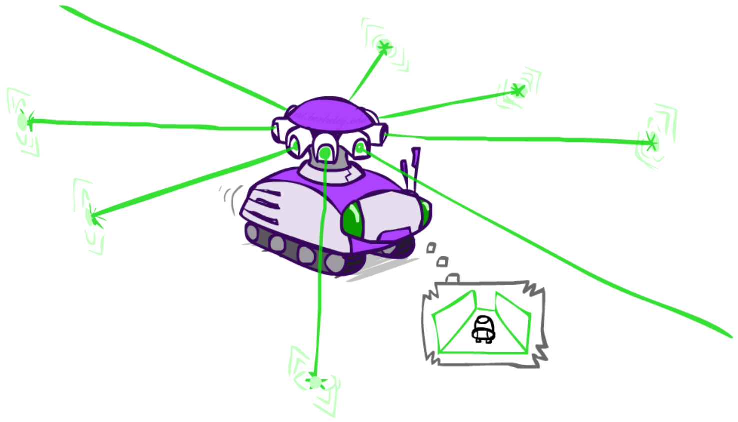




t=3

Example: Robot Localization

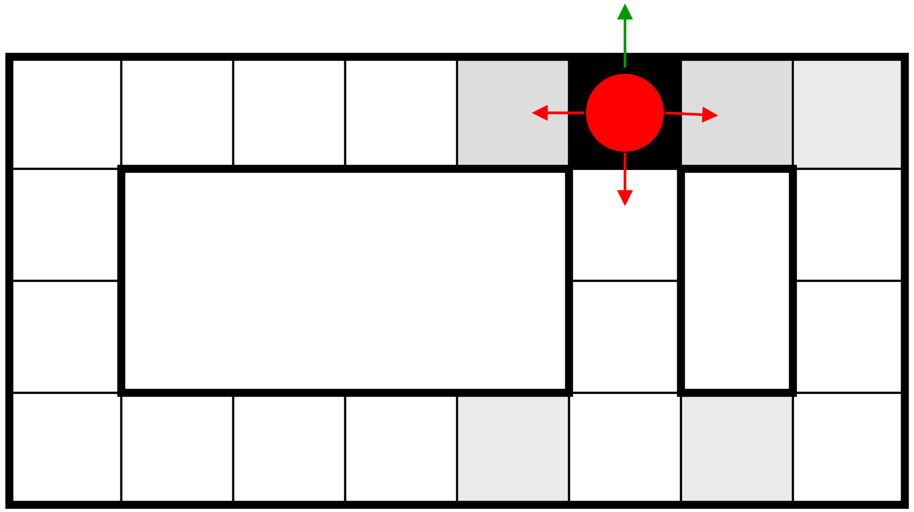


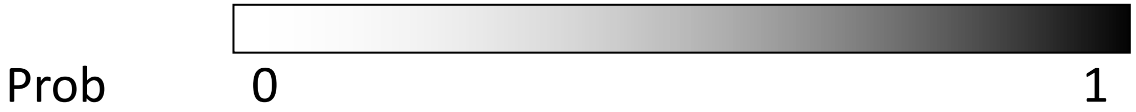


Prob

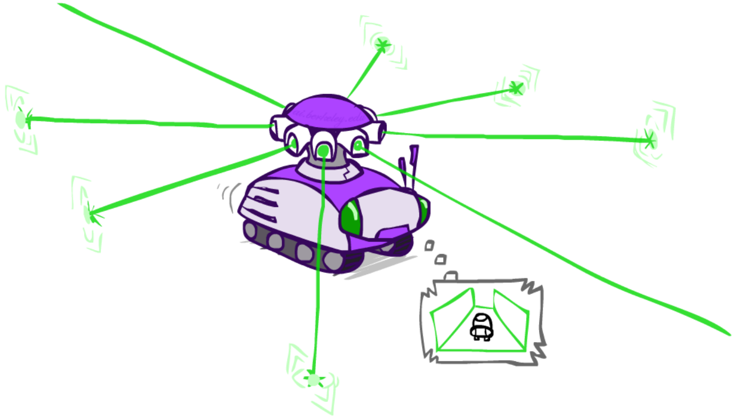
t=4

Example: Robot Localization





t=5



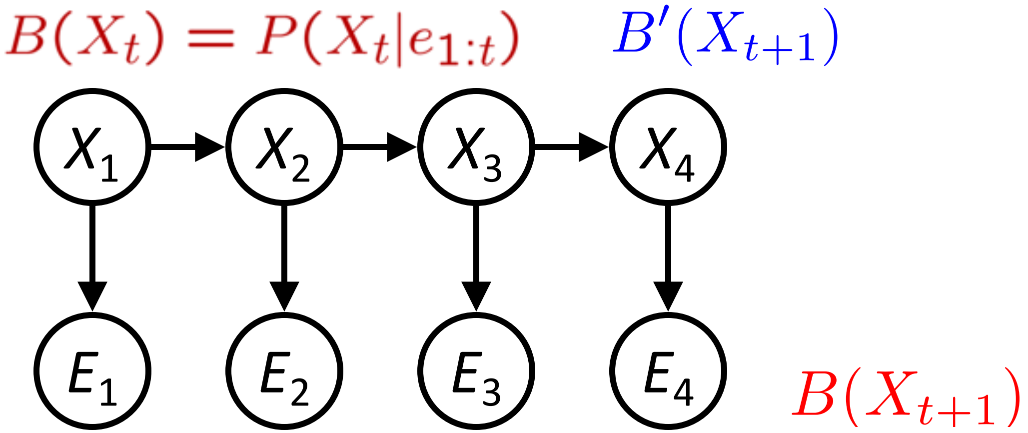
Inference: Find State Given Evidence

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

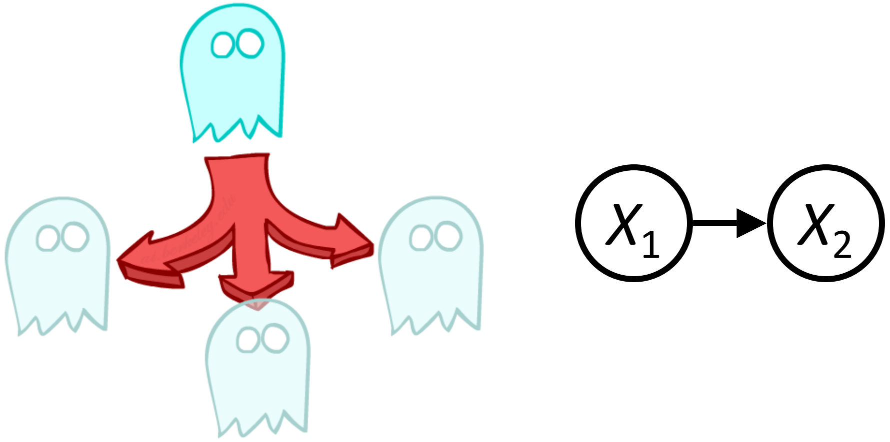
$$B_t(X) = P(X_t | e_{1:t})$$

- Idea: start with $P(X_1)$ and derive B_t in terms of B_{t-1}
  - equivalently, derive B_{t+1} in terms of B_t

Two Steps: Passage of Time + Observation



Inference: Base Cases



$$P(X_1|e_1)$$

$$P(X_1|e_1) = \frac{P(X_1, e_1)}{\sum_{x_1} P(x_1, e_1)}$$

$$P(X_1|e_1) = \frac{P(e_1|X_1)P(X_1)}{\sum_{x_1} P(e_1|x_1)P(x_1)}$$

$$P(X_2)$$

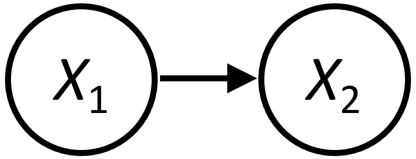
$$P(X_2) = \sum_{x_1} P(x_1, X_2)$$

$$P(X_2) = \sum_{x_1} P(X_2|x_1)P(x_1)$$

Passage of Time

- Assume we have current belief P(X | evidence to date)

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



- Then, after one time step passes:

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

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

$$= \sum_{x_t} P(X_{t+1}|x_t)P(x_t|e_{1:t})$$

- 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

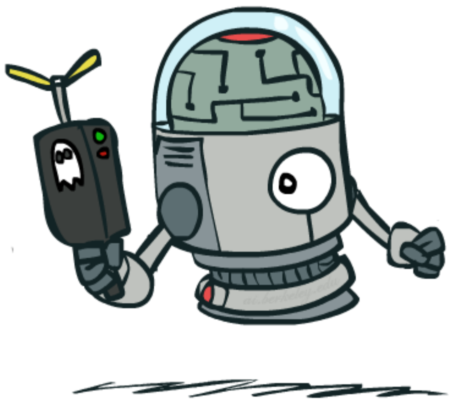
T = 1

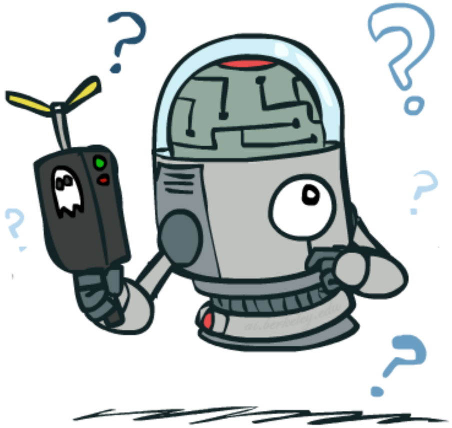
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

T = 2

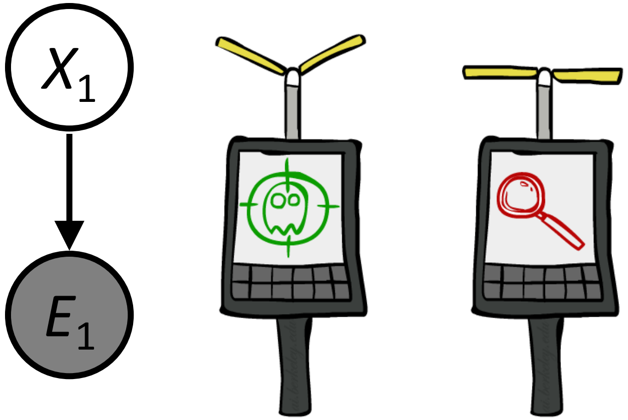
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

T = 5





Observation



- Assume we have current belief P(X | 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”

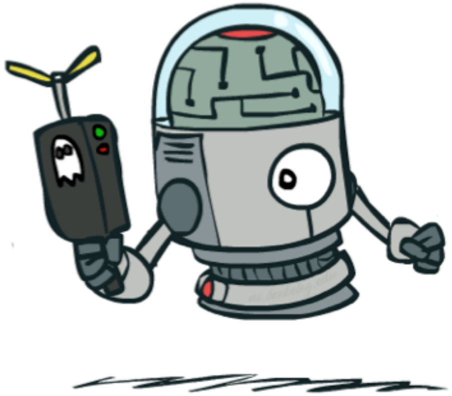
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Before observation

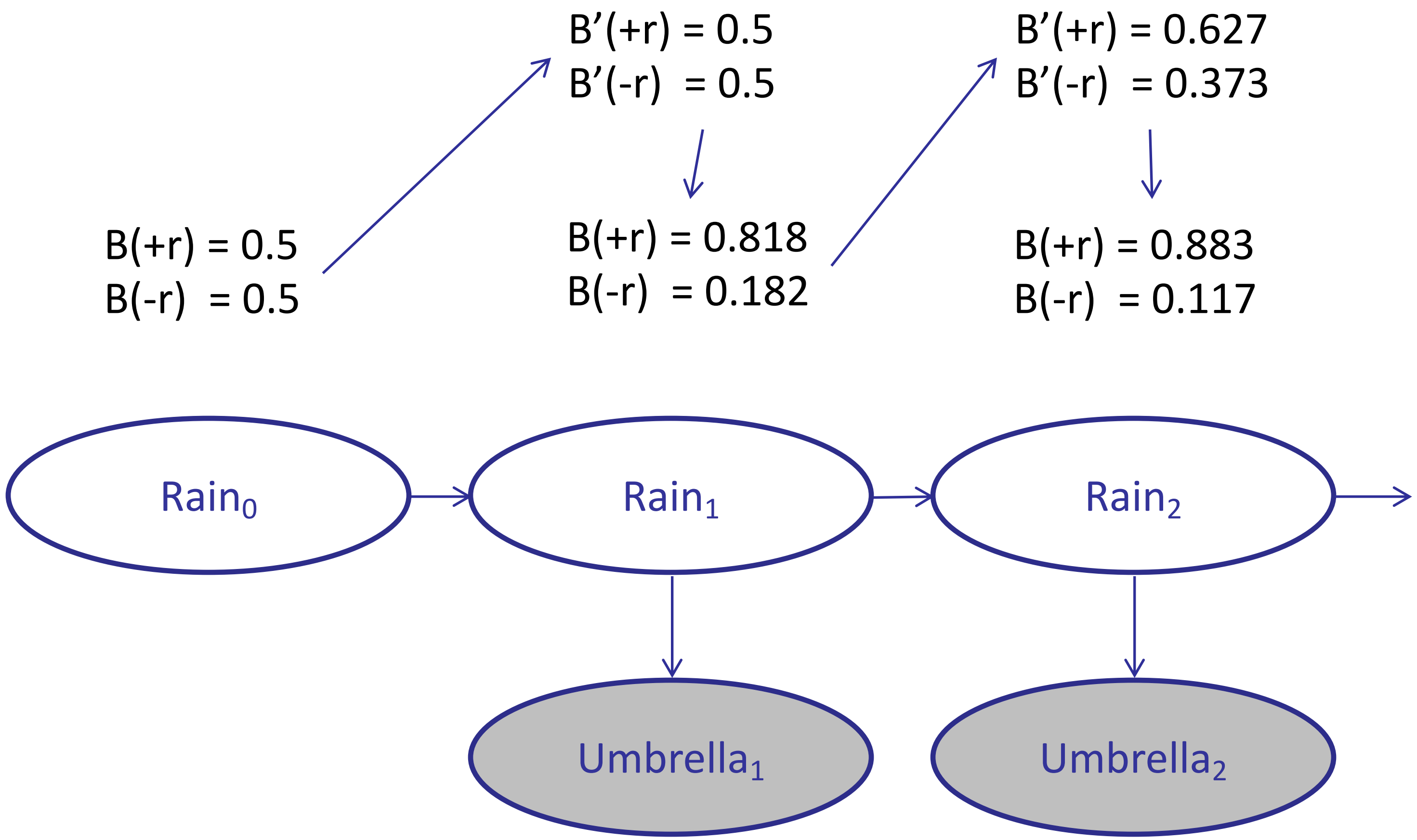
<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

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



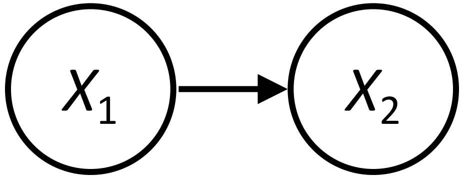
Example: Weather HMM



Online Belief Updates

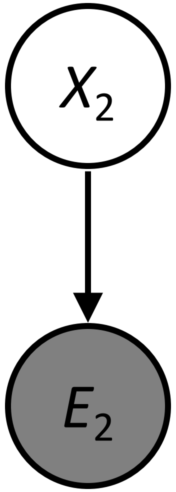
- Every time step, we start with current P(X | evidence)
- We update for time:

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



- 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)

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

$$P(x_t|e_{1:t}) \propto_{X_t} P(x_t, e_{1:t})$$

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

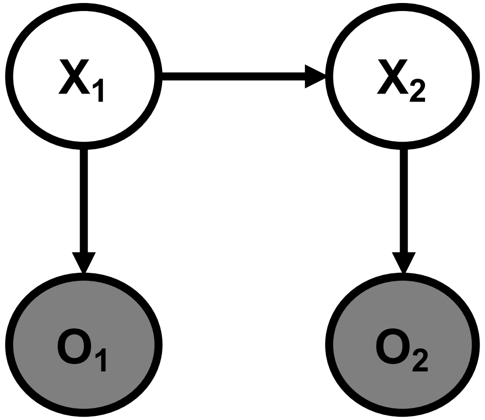
$$= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)$$

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

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

[Demo: Ghostbusters Exact Filtering (L15D2)]

习题:假设有以下隐式马尔可夫模型。



X_1	$P(X_1)$
0	0.3
1	0.7

X_t	X_{t+1}	$P(X_{t+1}\|X_{t+1})$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

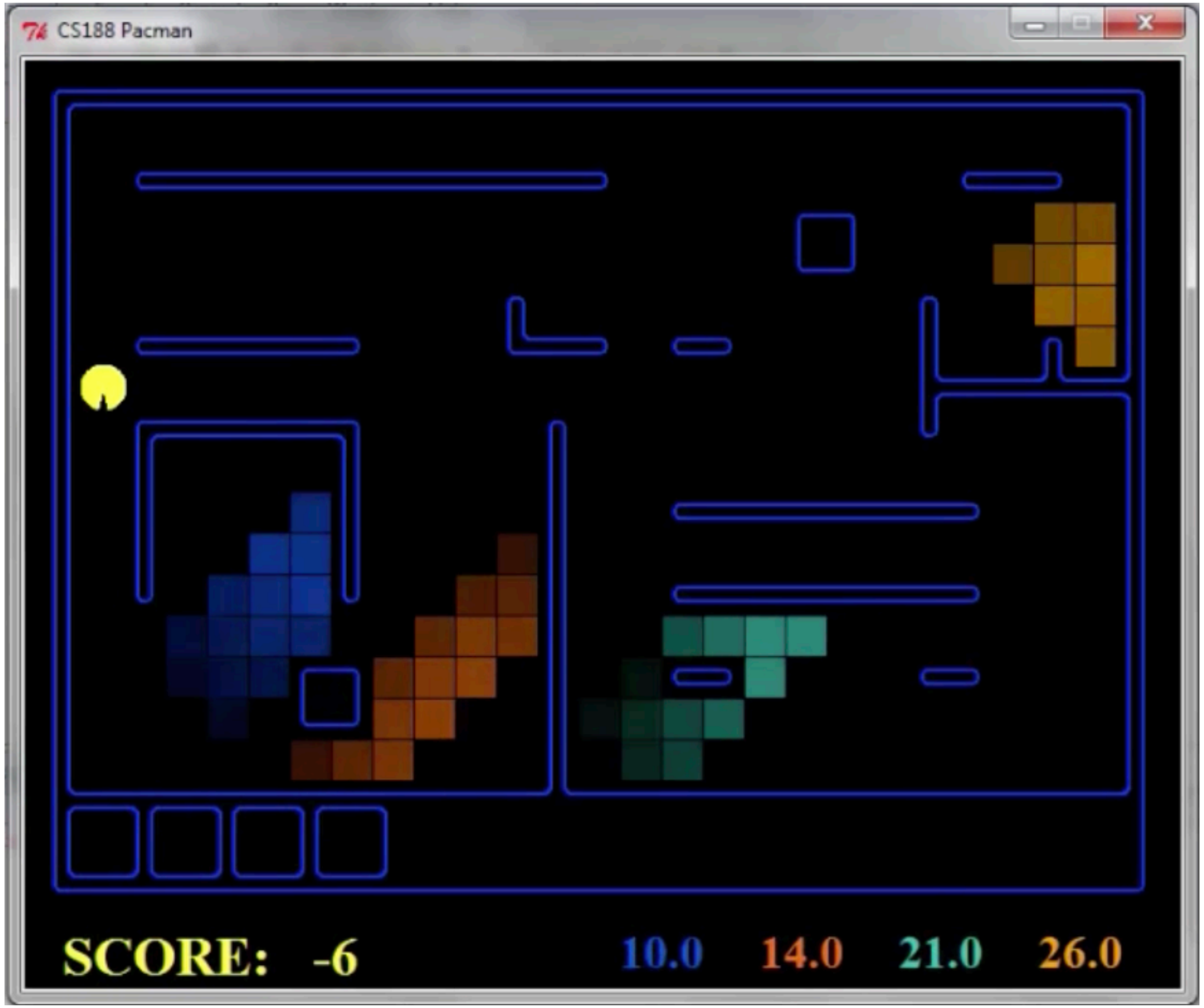
X_t	O_t	$P(O_t\|X_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

假设我们已知O_1=A和O_2=B,

1. **用前向算法求$P(X_2| O_1=A, O_2=B)$.**
2. **用粒子滤波求$P(X_2| O_1=A, O_2=B)$. 假设初始有2个粒子$P_1=0$, $P_2=1$, 产生的随机数如下:**

   { 0.22, 0.05, 0.33, 0.20, 0.84, 0.54, 0.79, 0.66, 0.14, 0.96}

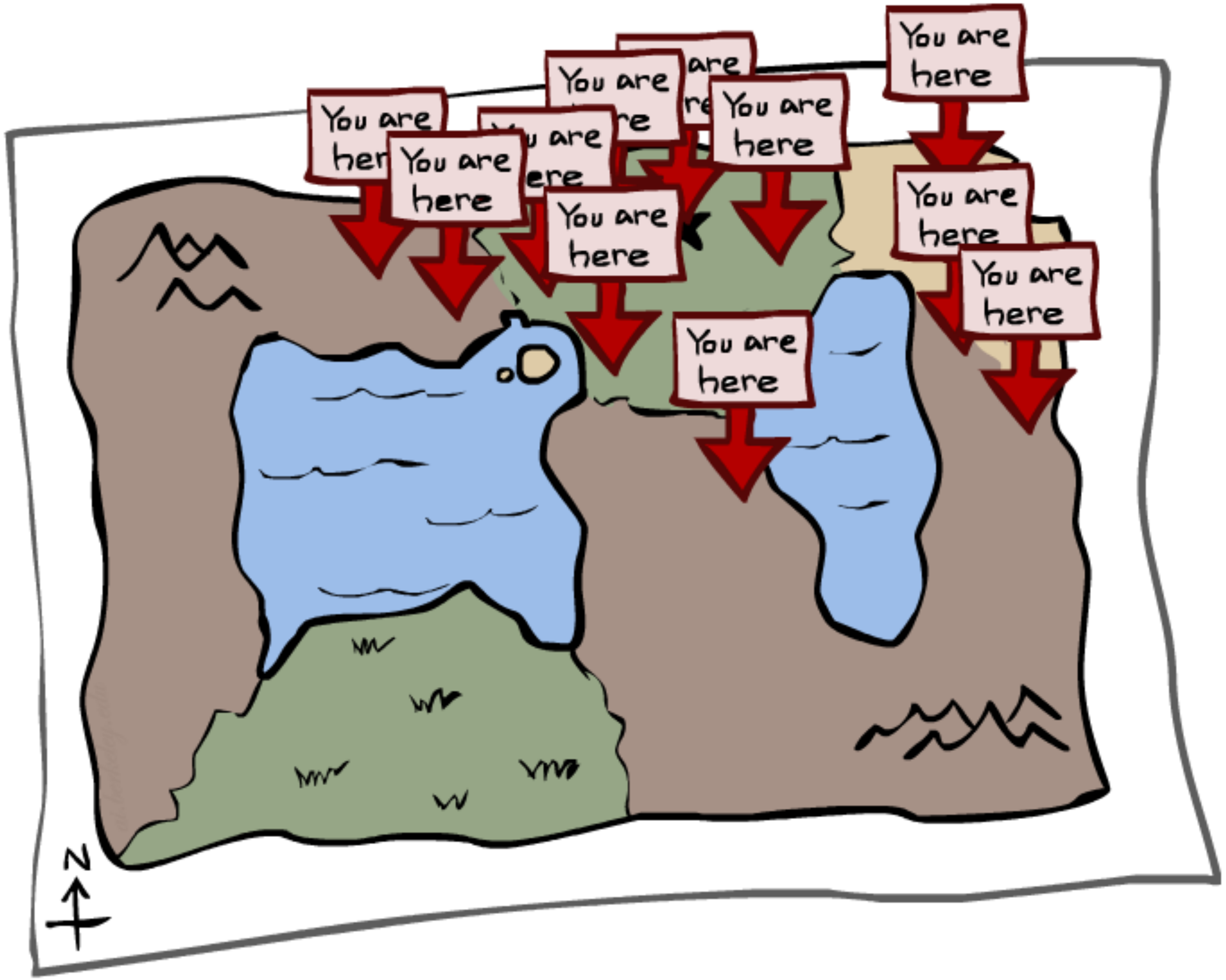
Pacman – Sonar



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

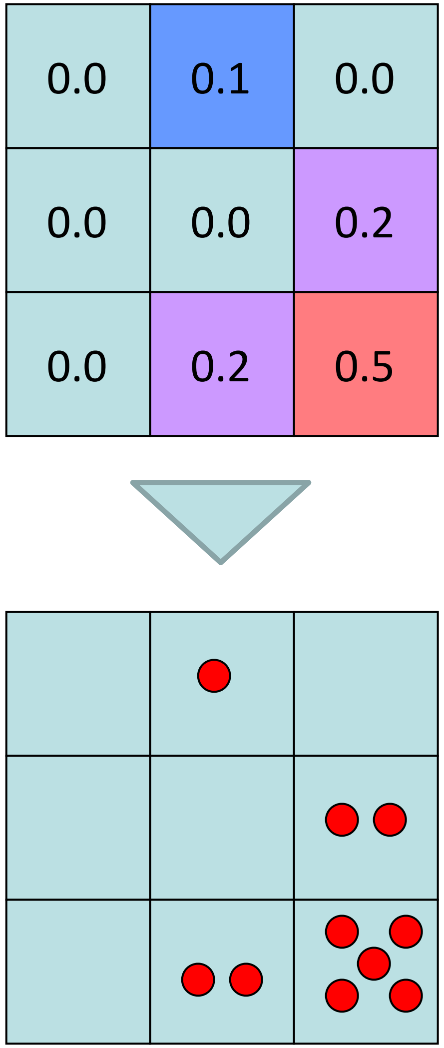
Video of Demo Pacman – Sonar (with beliefs)

Particle Filtering



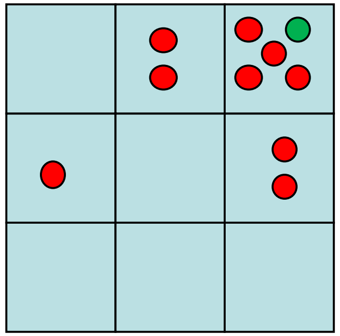
Particle Filtering

- Filtering: approximate solution
- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
- Solution: approximate inference
  - Track samples of X, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice
- Particle is just new name for sample



Representation: Particles

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|
  - Storing map from X to counts would defeat the point
- P(x) approximated by number of particles with value x
  - So, many x may have P(x) = 0!
  - More particles, more accuracy
- For now, all particles have a weight of 1



Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model

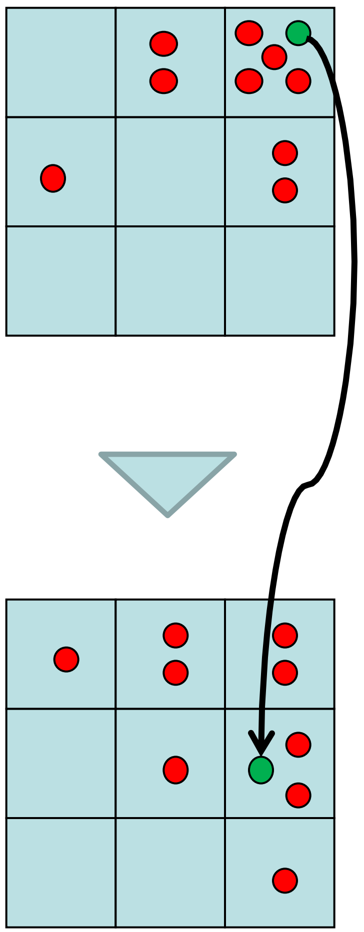
$$x' = \text{sample}(P(X'|x))$$

  - This is like prior sampling – samples' frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)

Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)



Particle Filtering: Observe

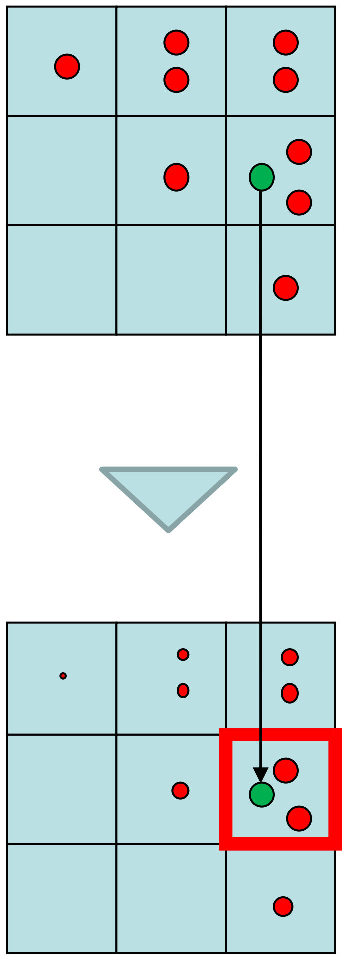
- Slightly trickier:
  - Don’t sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

$$w(x) = P(e|x)$$

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

  - As before, the probabilities don’t sum to one, since all have been downweighted (in fact they now sum to (N times) an approximation of P(e))

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)



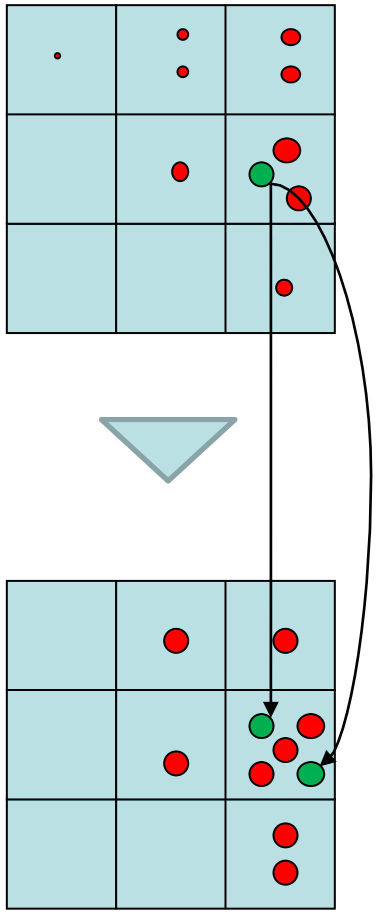
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

Particle Filtering: Resample

- Rather than tracking weighted samples, we resample
- N times, we choose from our weighted sample distribution (i.e. draw with replacement)
- This is equivalent to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one

Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

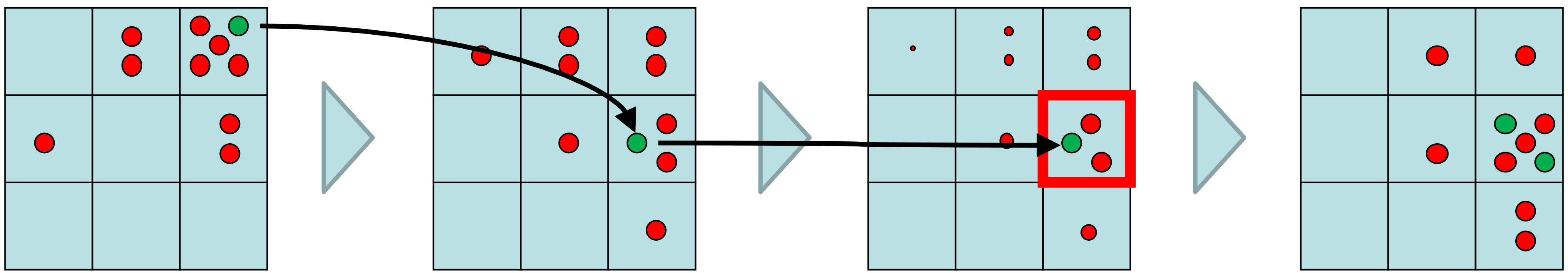
(New) Particles:
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)



Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution

Elapse Weight Resample



Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

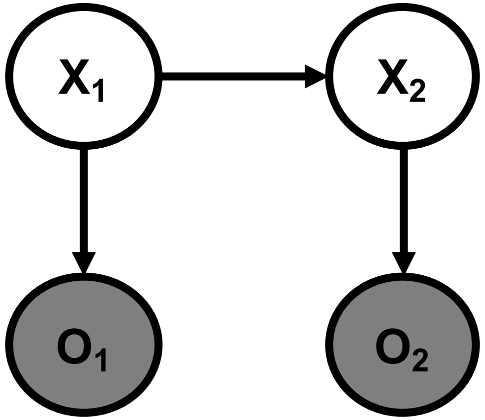
[Demos: ghostbusters particle filtering (L15D3,4,5)]

Video of Demo – Moderate Number of Particles

Video of Demo – One Particle

Video of Demo – Huge Number of Particles

习题:假设有以下隐式马尔可夫模型。



X_1	$P(X_1)$
0	0.3
1	0.7

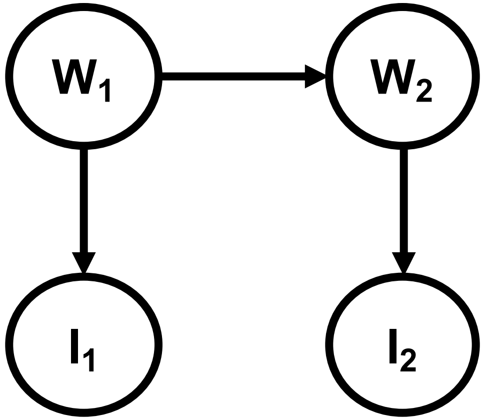
X_t	X_{t+1}	$P(X_{t+1}\|X_{t+1})$
0	0	0.4
0	1	0.6
1	0	0.8
1	1	0.2

X_t	O_t	$P(O_t\|X_t)$
0	A	0.9
0	B	0.1
1	A	0.5
1	B	0.5

假设我们已知O_1=A和O_2=B，

1. 用前向算法求$P(X_2| O_1=A, O_2=B)$.
2. 用粒子滤波求$P(X_2| O_1=A, O_2=B)$. 假设初始有2个粒子$P_1=0$，$P_2=1$，产生的随机数如下：
   { 0.22，0.05，0.33，0.20，0.84，0.54，0.79，0.66，0.14，0.96}

习题:假设有以下隐式马尔可夫模型。



W_1	$P(W_1)$
S	0.6
R	0.4

W_1	W_2	$P(W_2 \mid W_1)$
S	S	0.7
S	R	0.3
R	S	0.5
R	R	0.5

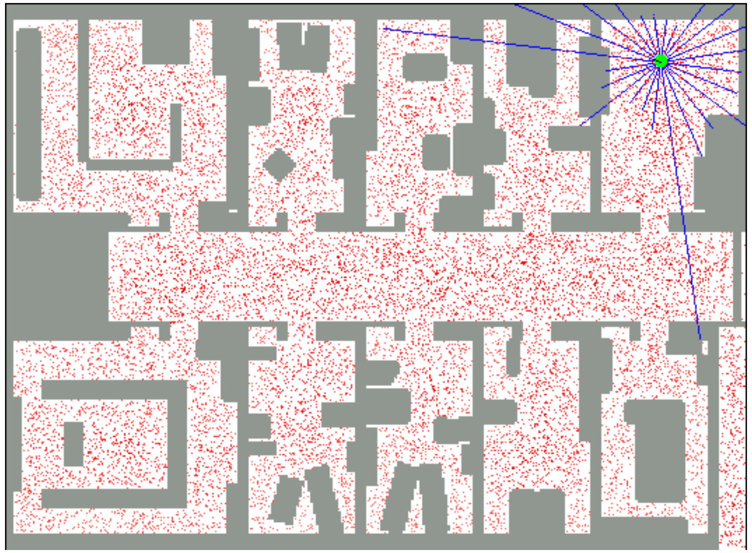
W	I	$P(I \mid W)$
S	T	0.9
S	F	0.1
R	T	0.2
R	F	0.8

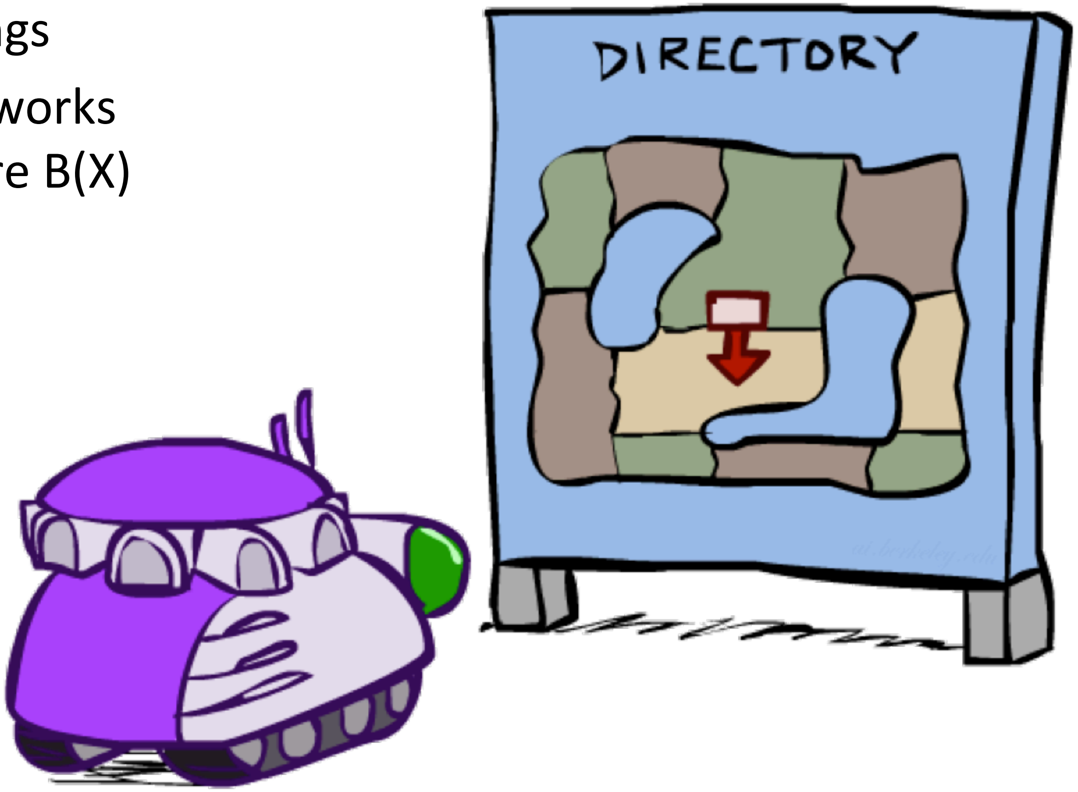
假设我们已知O_1=A和O_2=B,

1. **用前向算法求$P(X_2 \mid O_1=A, O_2=B)$.**
2. **用粒子滤波求$P(X_2 \mid O_1=A, O_2=B)$. 假设初始有2个粒子$P_1=0$, $P_2=1$, 产生的随机数如下:**
   { 0.22, 0.05, 0.33, 0.20, 0.84, 0.54, 0.79, 0.66, 0.14, 0.96}

Robot Localization

- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique





Particle Filter Localization (Sonar)

[Dieter Fox, et al.]

[Video: global-sonar-uw-annotated.avi]