CSE 473: Artificial Intelligence

Uncertainty and
Hidden Markov Models

Luke Zettlemoyer

Many slides over the course adapted from either Dan Klein,
Stuart Russell or Andrew Moore

Outline

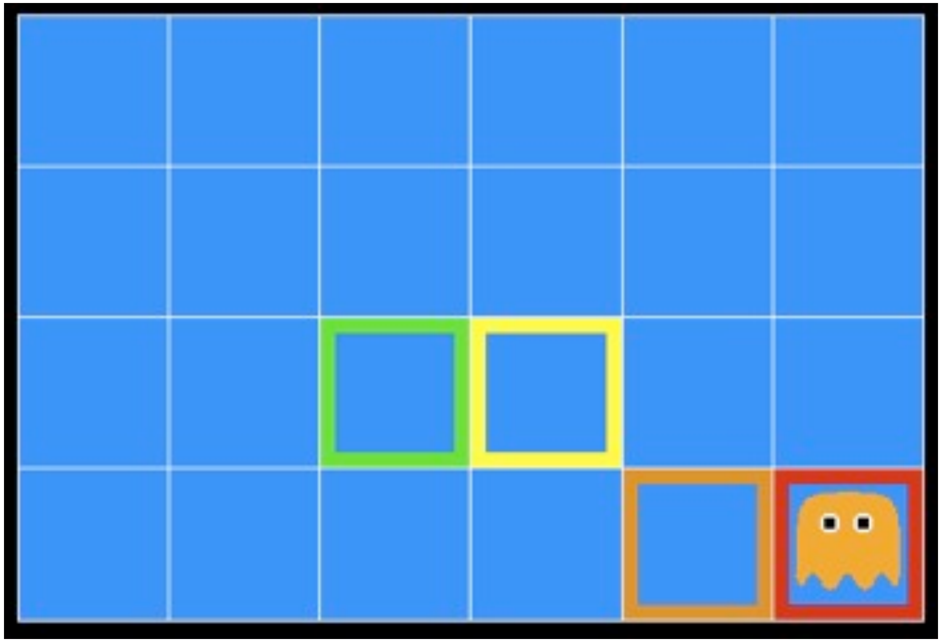
- Probability review
  - Random Variables and Events
  - Joint / Marginal / Conditional Distributions
  - Product Rule, Chain Rule, Bayes' Rule
  - Probabilistic Inference
- Probabilistic sequence models (and inference)
  - Markov Chains
  - Hidden Markov Models
  - Particle Filters

Probability Review

- Probability
  - Random Variables
  - Joint and Marginal Distributions
  - Conditional Distribution
  - Product Rule, Chain Rule, Bayes' Rule
  - Inference
- You'll need all this stuff A LOT for the next few weeks, so make sure you go over it now!

Inference in Ghostbusters

- A ghost is in the grid somewhere
- Sensor readings tell how close a square is to the ghost
  - On the ghost: red
  - 1 or 2 away: orange
  - 3 or 4 away: yellow
  - 5+ away: green



- Sensors are noisy, but we know P(Color | Distance)

P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
0.05	0.15	0.5	0.3

Random Variables

- A *random variable* is some aspect of the world about which we (may) have uncertainty
  - R = Is it raining?
  - D = How long will it take to drive to work?
  - L = Where am I?

- We denote random variables with capital letters

- Random variables have domains
  - R in {true, false}
  - D in $[0, \infty)$
  - L in possible locations, maybe {(0,0), (0,1), …}

Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$ specifies a real number for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$

$$P(x_1, x_2, \ldots x_n)$$

  - Size of distribution if n variables with domain sizes d?
  - Must obey: $P(x_1, x_2, \ldots x_n) \geq 0$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

$$P(T, W)$$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- A *probabilistic model* is a joint distribution over variables of interest
- For all but the smallest distributions, impractical to write out

Events

- An *outcome* is a joint assignment for all the variables

$$(x_1, x_2, \ldots x_n)$$

- An *event* is a set E of outcomes

$$P(E) = \sum_{(x_1 \ldots x_n) \in E} P(x_1 \ldots x_n)$$

- From a joint distribution, we can calculate the probability of any event
  - Probability that it's hot AND sunny?
  - Probability that it's hot?
  - Probability that it's hot OR sunny?

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Marginal Distributions

- *Marginal distributions* are sub-tables which eliminate variables
- Marginalization (summing out): Combine collapsed rows by adding

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_{w} P(t, w)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$$P(w) = \sum_{t} P(t, w)$$

$P(W)$

W	P
sun	0.6
rain	0.4

Conditional Distributions

- Conditional distributions are probability distributions over some variables given fixed values of others

Conditional Distributions

$P(W|T)$

$P(W|T = hot)$

W	P
sun	0.8
rain	0.2

$P(W|T = cold)$

W	P
sun	0.4
rain	0.6

Joint Distribution

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)}$$

Normalization Trick

- A trick to get a whole conditional distribution at once:
  - Select the joint probabilities matching the evidence
  - Normalize the selection (make it sum to one)

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

Select →

$P(T, r)$

T	R	P
hot	rain	0.1
cold	rain	0.3

Normalize →

$P(T|r)$

T	P
hot	0.25
cold	0.75

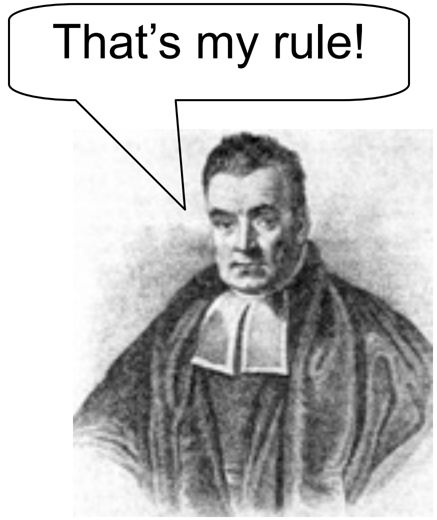
- Why does this work? Sum of selection is P(evidence)! (P(r), here)

$$P(x_1|x_2) = \frac{P(x_1, x_2)}{P(x_2)} = \frac{P(x_1, x_2)}{\sum_{x_1} P(x_1, x_2)}$$

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

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



- Dividing, we get:

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

- Why is this at all helpful?
  - Lets us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many systems we'll see later (e.g. ASR, MT)

- In the running for most important AI equation!

The Product Rule

- Sometimes have conditional distributions but want the joint

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

- Example:

$P(W)$

R	P
sun	0.8
rain	0.2

$P(D|W)$

D	W	P
wet	sun	0.1
dry	sun	0.9
wet	rain	0.7
dry	rain	0.3

$\Leftrightarrow$

$P(D,W)$

D	W	P
wet	sun	0.08
dry	sun	0.72
wet	rain	0.14
dry	rain	0.06

The Chain Rule

- More generally, can always write any joint distribution as an incremental product of conditional distributions

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})$$

- Why is this always true?

Probabilistic Inference

- *Probabilistic inference*: compute a desired probability from other known probabilities (e.g. conditional from joint)

- We generally compute conditional probabilities
  - P(on time | no reported accidents) = 0.90
  - These represent the agent's *beliefs* given the evidence

- Probabilities change with new evidence:
  - P(on time | no accidents, 5 a.m.) = 0.95
  - P(on time | no accidents, 5 a.m., raining) = 0.80
  - Observing new evidence causes *beliefs to be updated*

Inference by Enumeration

- P(sun)?
- P(sun | winter)?
- P(sun | winter, warm)?

S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \dots E_k = e_1 \dots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \dots H_r$

  $\Big\}$ $X_1, X_2, \dots X_n$ *All variables*

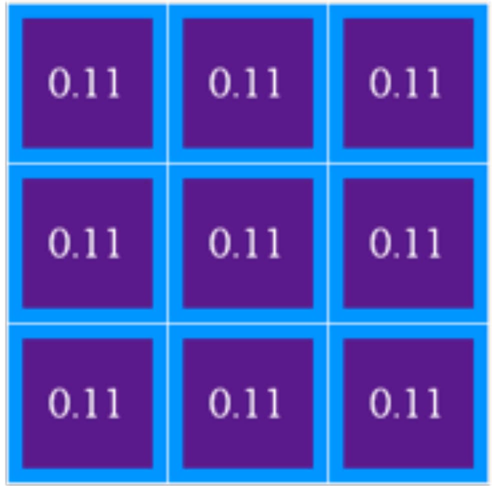
- We want: $P(Q|e_1 \dots e_k)$
- First, select the entries consistent with the evidence
- Second, sum out H to get joint of Query and evidence:

$$P(Q, e_1 \dots e_k) = \sum_{h_1 \dots h_r} \underbrace{P(Q, h_1 \dots h_r, e_1 \dots e_k)}_{X_1, X_2, \dots X_n}$$

- Finally, normalize the remaining entries to conditionalize
- Obvious problems:
  - Worst-case time complexity $O(d^n)$
  - Space complexity $O(d^n)$ to store the joint distribution

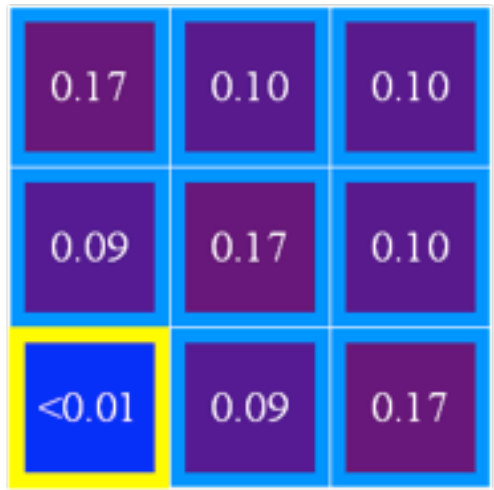
Ghostbusters, Revisited

- Let’s say we have two distributions:
  - Prior distribution over ghost location: P(G)
    - Let’s say this is uniform
  - Sensor reading model: P(R | G)
    - Given: we know what our sensors do
    - R = reading color measured at (1,1)
    - E.g. P(R = yellow | G=(1,1)) = 0.1



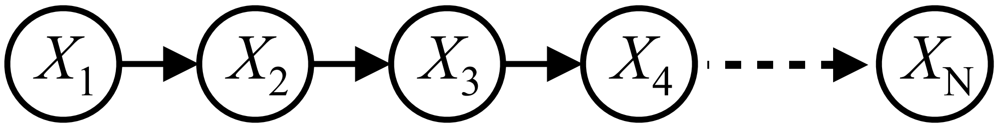
- We can calculate the posterior distribution P(G|r) over ghost locations given a reading using Bayes’ rule:

$$P(g|r) \propto P(r|g)P(g)$$



Markov Models (Markov Chains)

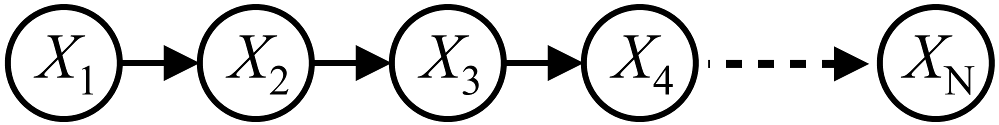
- A Markov model is:
  - a MDP with no actions (and no rewards)
  - a chain-structured Bayesian Network (BN)



- A Markov model includes:
  - Random variables X_t for all time steps t (the state)
  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

$$P(X_1) \quad \text{and} \quad P(X_t|X_{t-1})$$

Markov Models (Markov Chains)

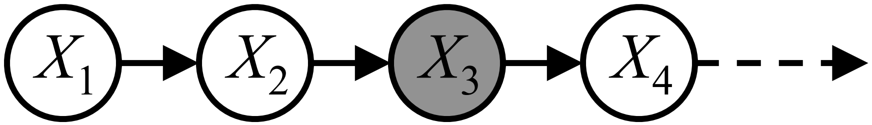


- A Markov model defines
  - a joint probability distribution:

$$P(X_1, \ldots, X_n) = P(X_1) \prod_{t=2}^{N} P(X_t | X_{t-1})$$

- One common inference problem:
  - Compute marginals $P(X_t)$ for all time steps t

Conditional Independence

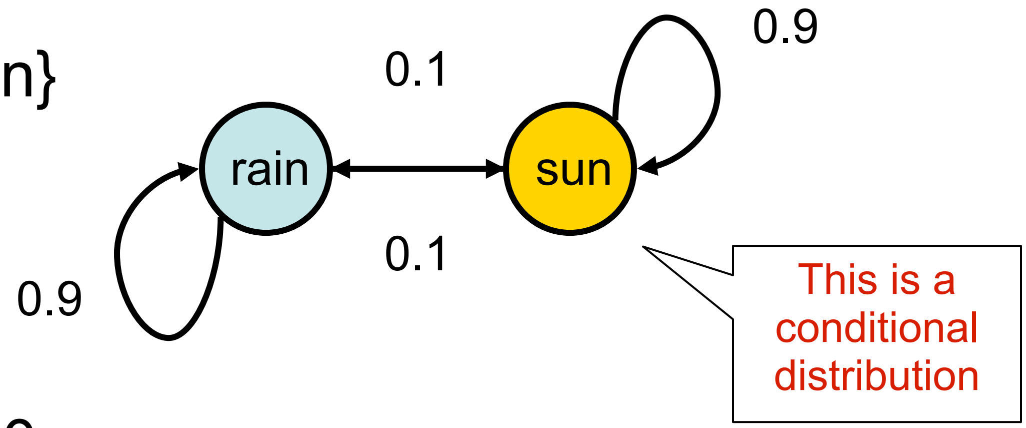


- Basic conditional independence:
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property

- Note that the chain is just a (growing) BN
  - As we will see later, we can use generic BN reasoning on it if we truncate the chain at a fixed length

Example: Markov Chain

- Weather:
  - States: X = {rain, sun}
  - Transitions:



  - Initial distribution: 1.0 sun
  - What's the probability distribution after one step?

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

$$0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9$$

Markov Chain Inference

- Question: probability of being in state x at time t?
- Slow answer:
  - Enumerate all sequences of length t which end in s
  - Add up their probabilities

$$P(X_t = sun) = \sum_{x_1 \ldots x_{t-1}} P(x_1, \ldots x_{t-1}, sun)$$

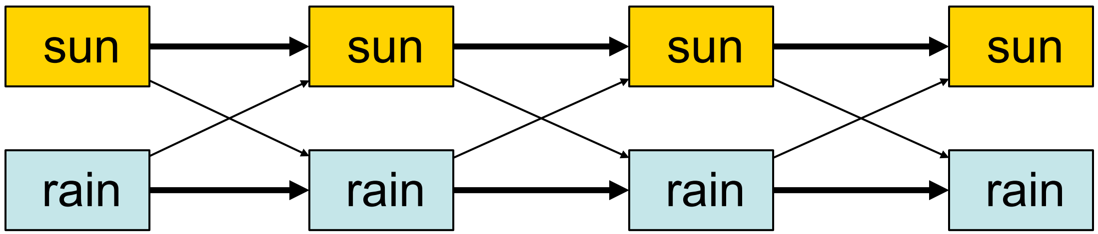
$$P(X_1 = sun)P(X_2 = sun|X_1 = sun)P(X_3 = sun|X_2 = sun)P(X_4 = sun|X_3 = sun)$$

$$P(X_1 = sun)P(X_2 = rain|X_1 = sun)P(X_3 = sun|X_2 = rain)P(X_4 = sun|X_3 = sun)$$

$$\vdots$$

Mini-Forward Algorithm

- Question: What's P(X) on some day t?
  - We don't need to enumerate every sequence!



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

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

Forward simulation

Proof that $P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$

$$
\begin{aligned}
P(x_t) &= \sum_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_t) \\
&= \sum_{x_1,\ldots,x_{t-1}} P(x_1)\prod_{i=1}^{t} P(x_i|x_{i-1}) \\
&= \sum_{x_{t-1}} P(x_t|x_{t-1}) \sum_{x_1,\ldots,x_{t-2}} P(x_1)\prod_{i=1}^{t-1} P(x_i|x_{i-1}) \\
&= \sum_{x_{t-1}} P(x_t|x_{t-1}) \sum_{x_1,\ldots,x_{t-2}} P(x_1,\ldots,x_{t-1}) \\
&= \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})
\end{aligned}
$$

Example

- From initial observation of sun

$$\left\langle \begin{matrix} 1.0 \\ 0.0 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.9 \\ 0.1 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.82 \\ 0.18 \end{matrix} \right\rangle \quad \Rightarrow \quad \left\langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \right\rangle$$

$$P(X_1) \quad P(X_2) \quad P(X_3) \quad\quad P(X_\infty)$$

- From initial observation of rain

$$\left\langle \begin{matrix} 0.0 \\ 1.0 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.1 \\ 0.9 \end{matrix} \right\rangle \quad \left\langle \begin{matrix} 0.18 \\ 0.82 \end{matrix} \right\rangle \quad \Rightarrow \quad \left\langle \begin{matrix} 0.5 \\ 0.5 \end{matrix} \right\rangle$$

$$P(X_1) \quad P(X_2) \quad P(X_3) \quad\quad P(X_\infty)$$

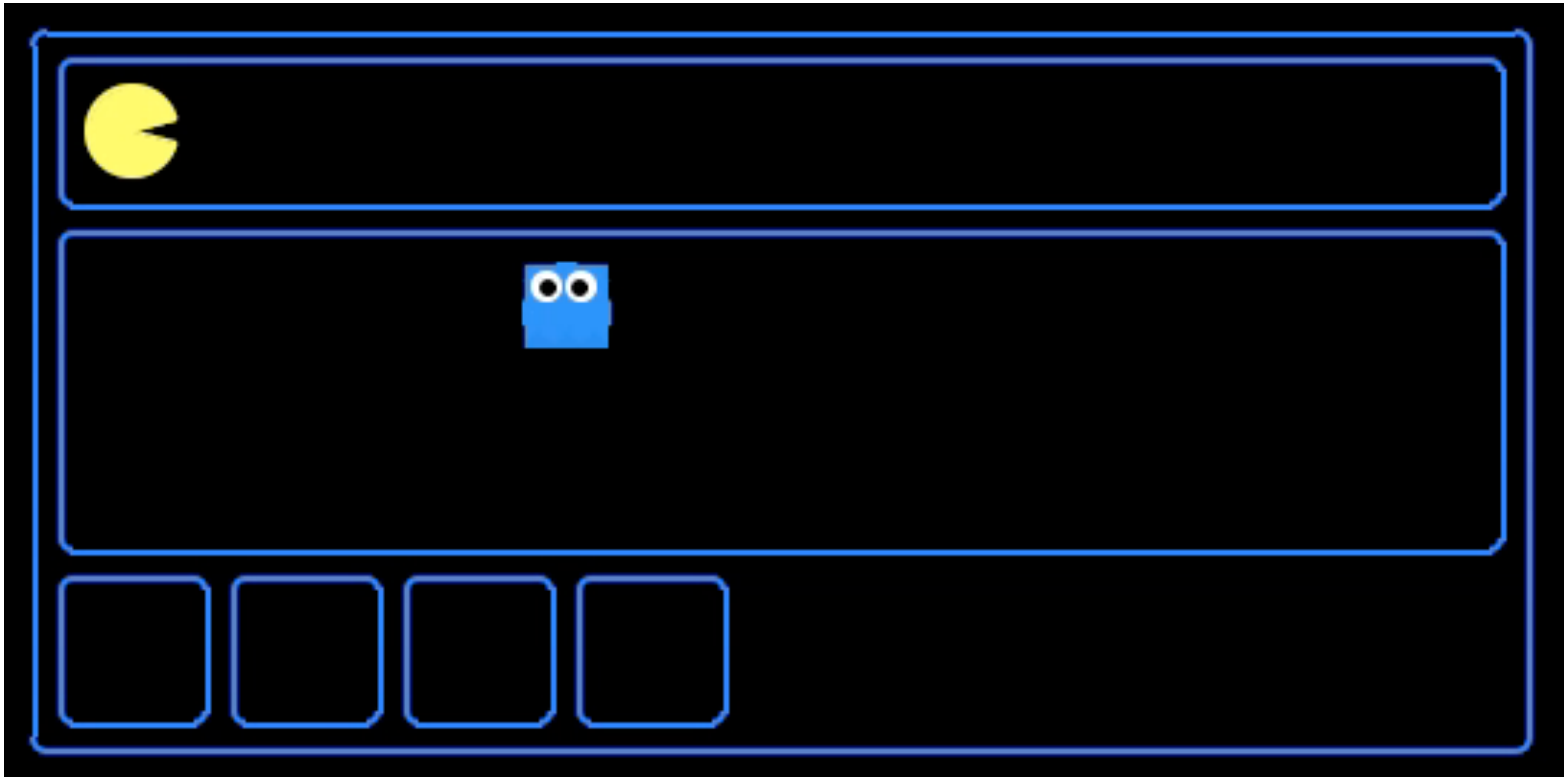
Stationary Distributions

- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!

- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the stationary distribution of the chain
  - Usually, can only predict a short time out

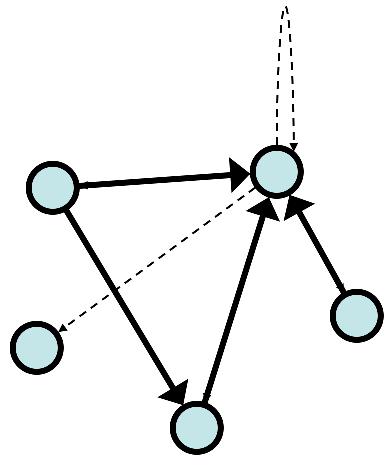
Pac-man Markov Chain

Pac-man knows the ghost's initial position, but gets no observations!



Web Link Analysis

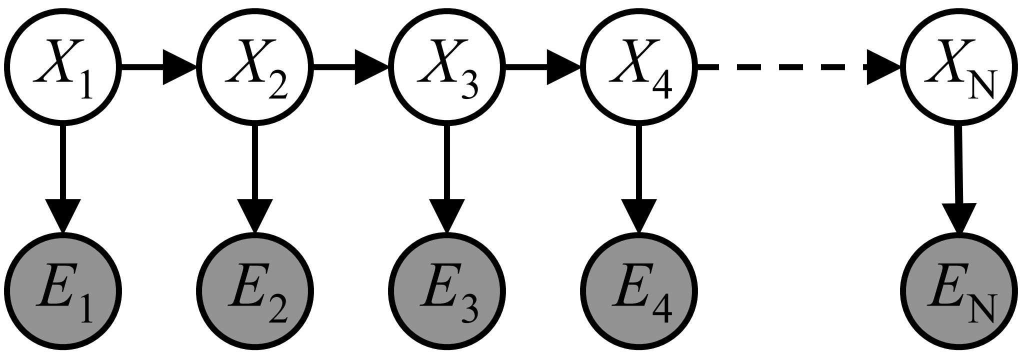
- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (dotted lines, not all shown)
    - With prob. 1-c, follow a random outlink (solid lines)



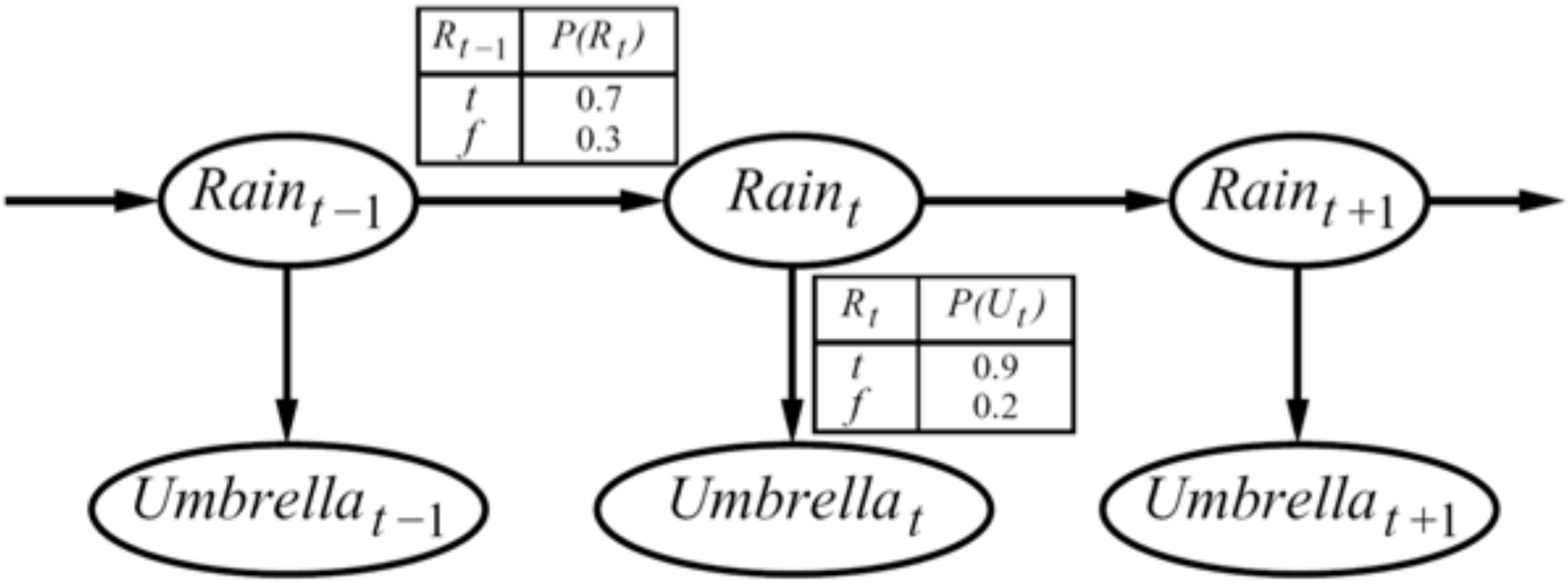
- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page
  - Somewhat robust to link spam
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis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

Hidden Markov Models

- Markov chains not so useful for most agents
  - Eventually you don't know anything anymore
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states S
  - You observe outputs (effects) at each time step
  - POMDPs without actions (or rewards).
  - As a Bayes' net:

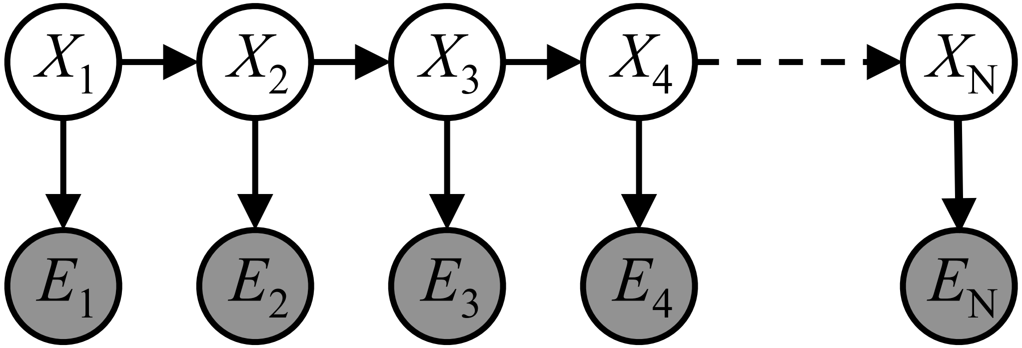


Example



- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Hidden Markov Models



- Defines a joint probability distribution:

$$P(X_1, \ldots, X_n, E_1, \ldots, E_n) =$$

$$P(X_{1:n}, E_{1:n}) =$$

$$P(X_1)P(E_1|X_1)\prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)$$

Ghostbusters HMM

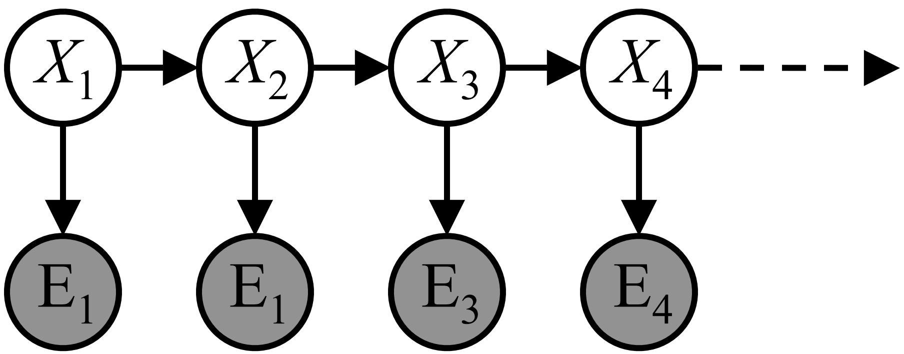
- $P(X_1)$ = uniform
- $P(X'|X)$ = usually move clockwise, but sometimes move in a random direction or stay in place
- $P(E|X)$ = same sensor model as before: red means close, green means far away.

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

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



HMM Computations

- Given
  - joint $P(X_{1:n}, E_{1:n})$
  - evidence $E_{1:n} = e_{1:n}$

- Inference problems include:
  - Filtering, find $P(X_t|e_{1:t})$ for all t
  - Smoothing, find $P(X_t|e_{1:n})$ for all t
  - Most probable explanation, find

$$x^*_{1:n} = \text{argmax}_{x_{1:n}} P(x_{1:n}|e_{1:n})$$

Real HMM Examples

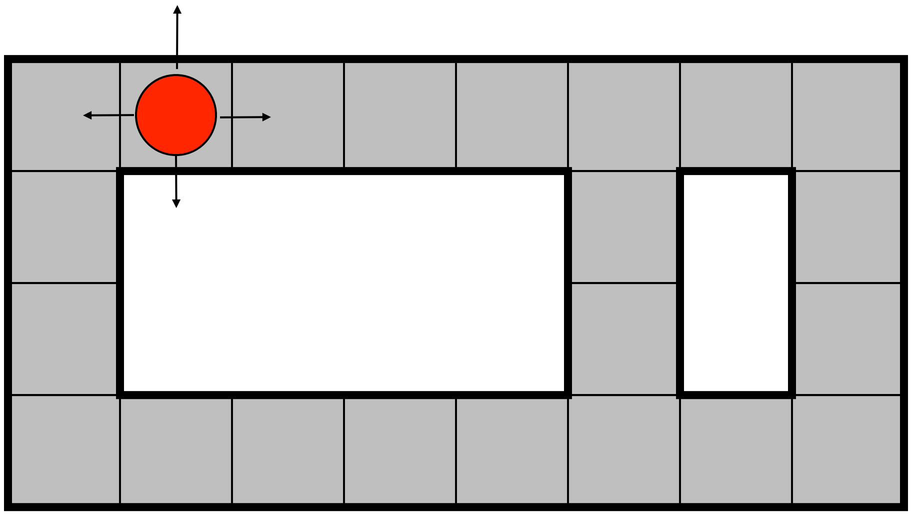
- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options
- Robot tracking:
  - Observations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

Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution B(X) (the belief state) over time
- We start with B(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



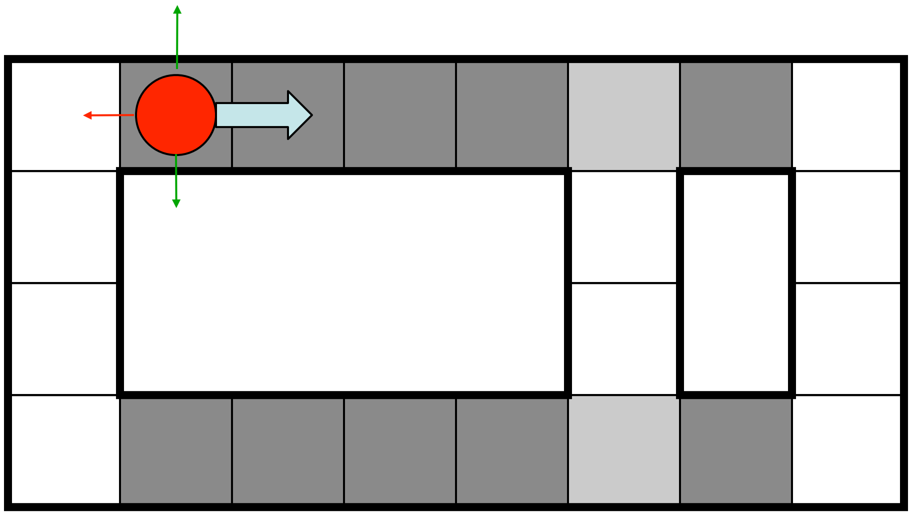
Prob 0 1

t=0

Sensor model: never more than 1 mistake

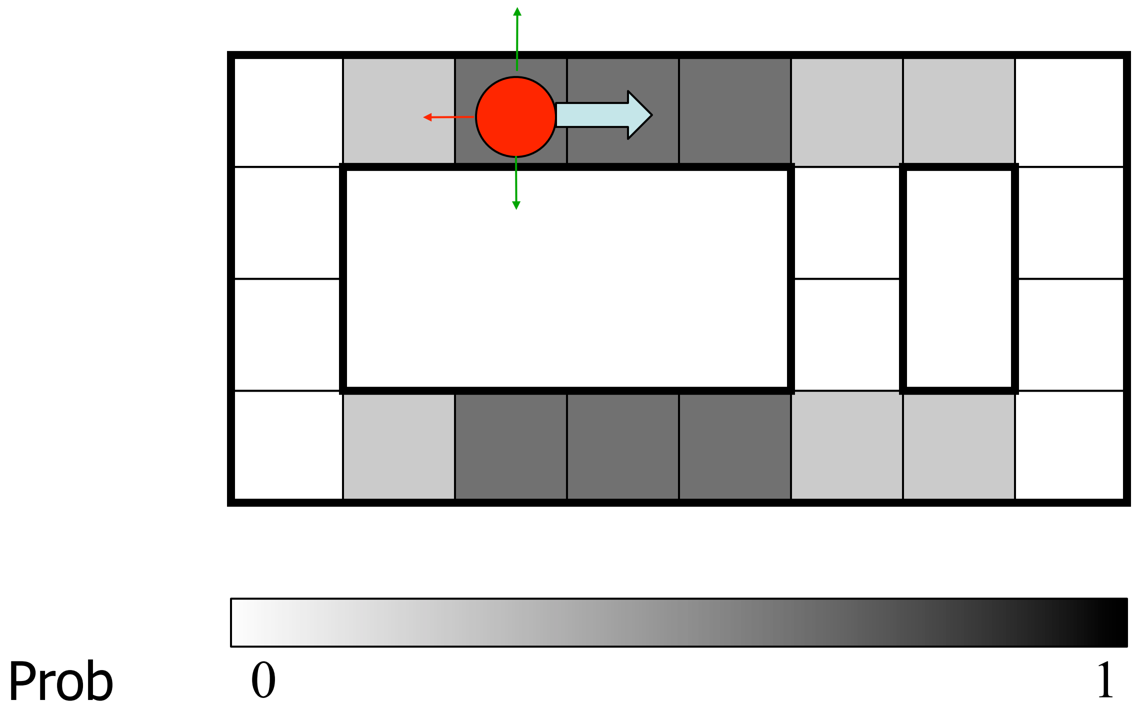
Motion model: may not execute action with small prob.

Example: Robot Localization



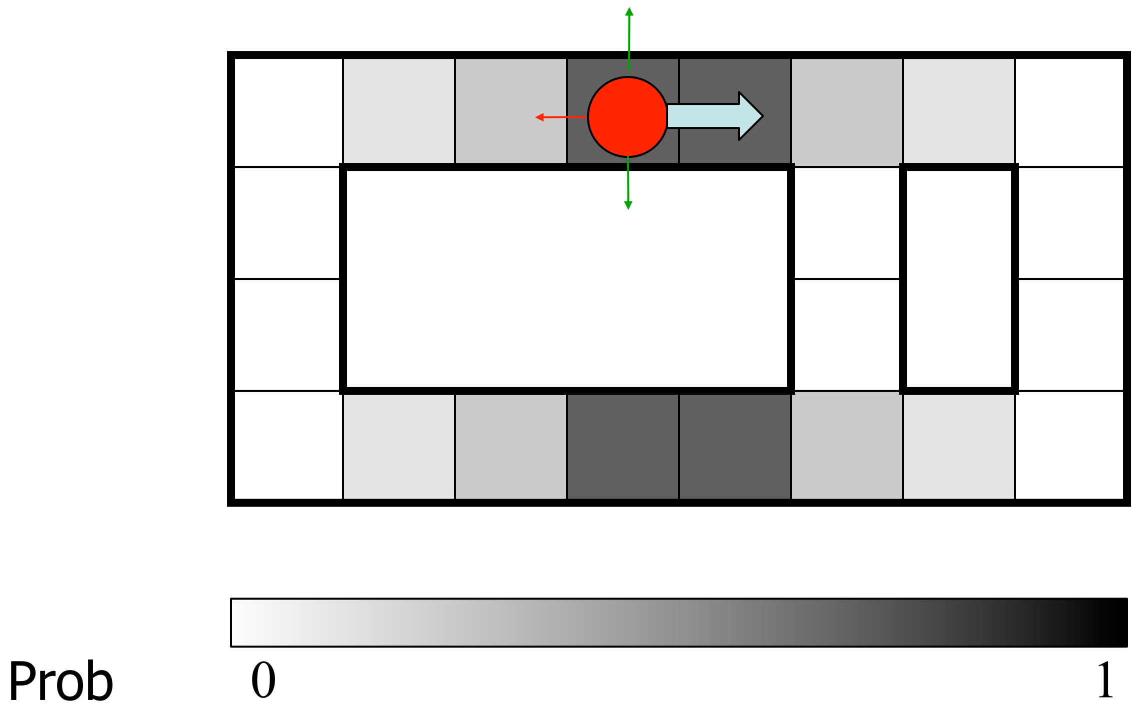
t=1

Example: Robot Localization



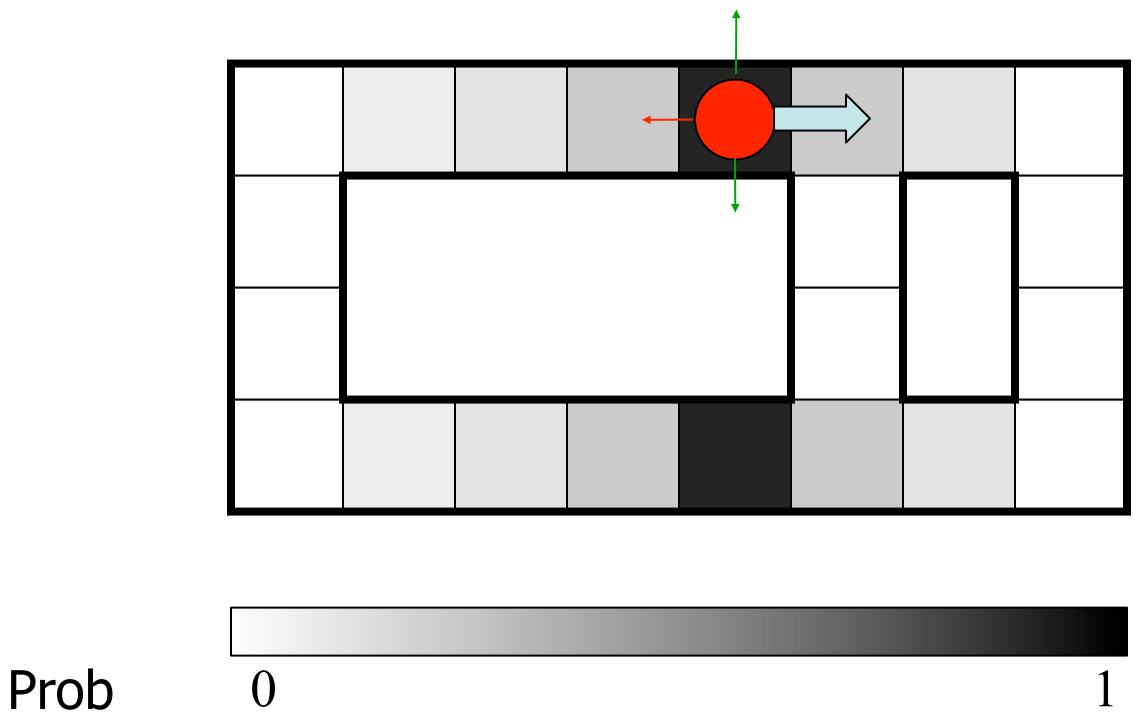
t=2

Example: Robot Localization



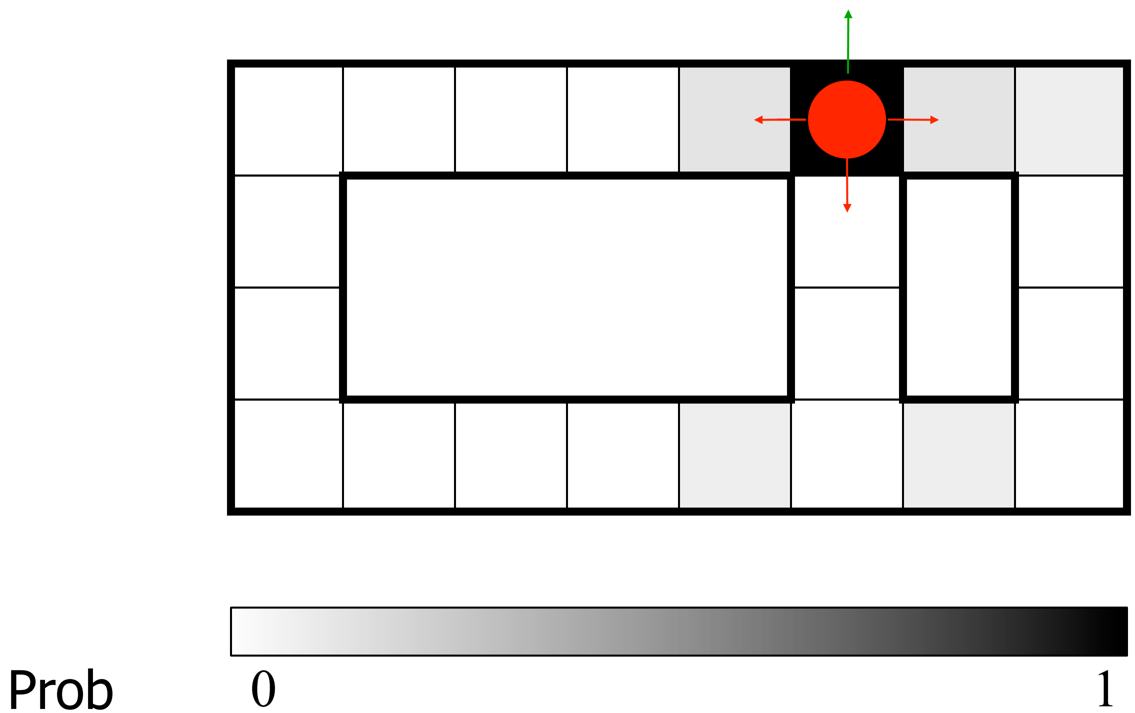
t=3

Example: Robot Localization



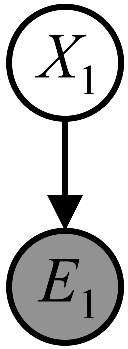
t=4

Example: Robot Localization



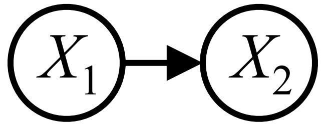
t=5

Inference: Simple Cases



$$P(X_1|e_1)$$

$$P(x_1|e_1) = P(x_1, e_1)/P(e_1)$$
$$\propto_{X_1} P(x_1, e_1)$$
$$= P(x_1)P(e_1|x_1)$$

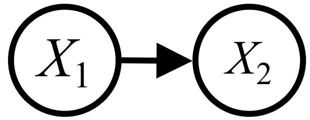


$$P(X_2)$$

$$P(x_2) = \sum_{x_1} P(x_1, x_2)$$
$$= \sum_{x_1} P(x_1)P(x_2|x_1)$$

Online Belief Updates

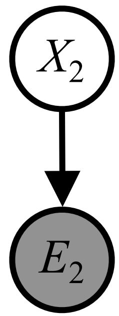
- Every time step, we start with current P(X | evidence)
- We update for time:



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

- We update for evidence:

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

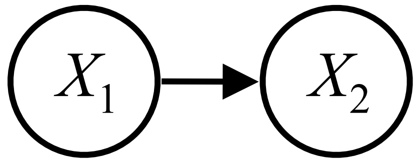


Passage of Time

- 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:



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

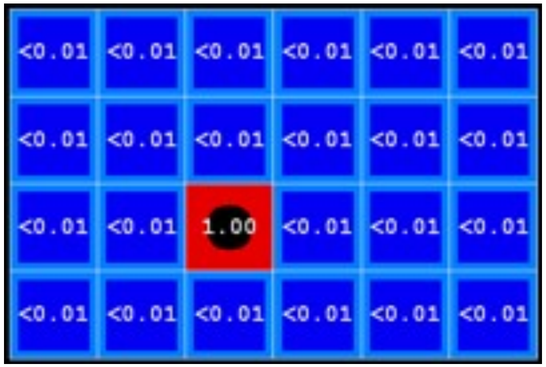
- Or, compactly:

$$B'(X') = \sum_{x} P(X'|x)B(x)$$

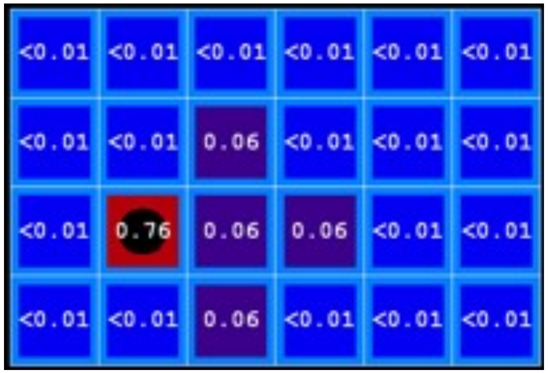
- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Example: Passage of Time

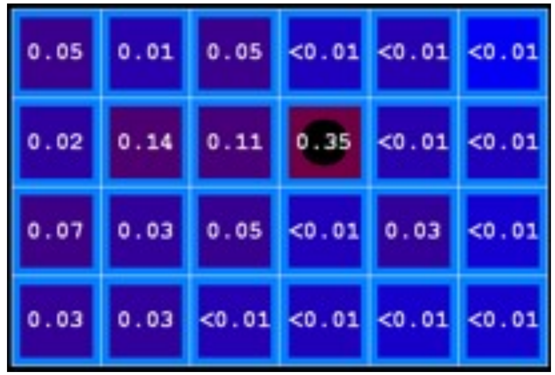
- As time passes, uncertainty “accumulates”



T = 1



T = 2



T = 5

$$B'(X') = \sum_x P(X'|x)B(x)$$

Transition model: ghosts usually go clockwise

Observation

- Assume we have current belief P(X | previous evidence):

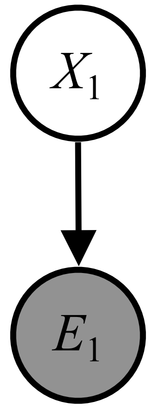
$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- Then:

$$P(X_{t+1}|e_{1:t+1}) \propto P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t})$$

- Or:

$$B(X_{t+1}) \propto P(e|X)B'(X_{t+1})$$

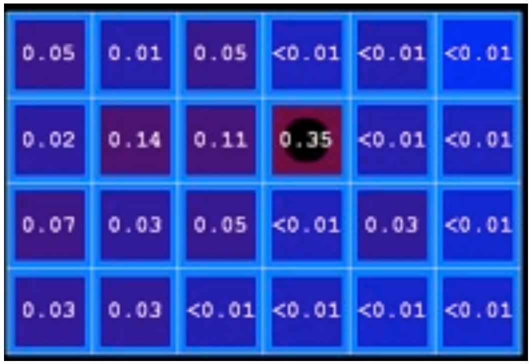


- Basic idea: beliefs reweighted by likelihood of evidence

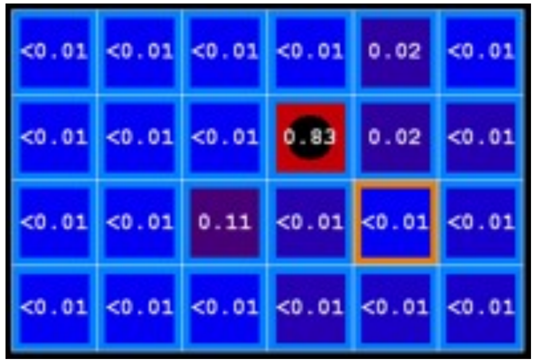
- Unlike passage of time, we have to renormalize

Example: Observation

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



Before observation



After observation

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

The Forward Algorithm

- We to know: $B_t(X) = P(X_t|e_{1:t})$
- We can derive the following updates

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

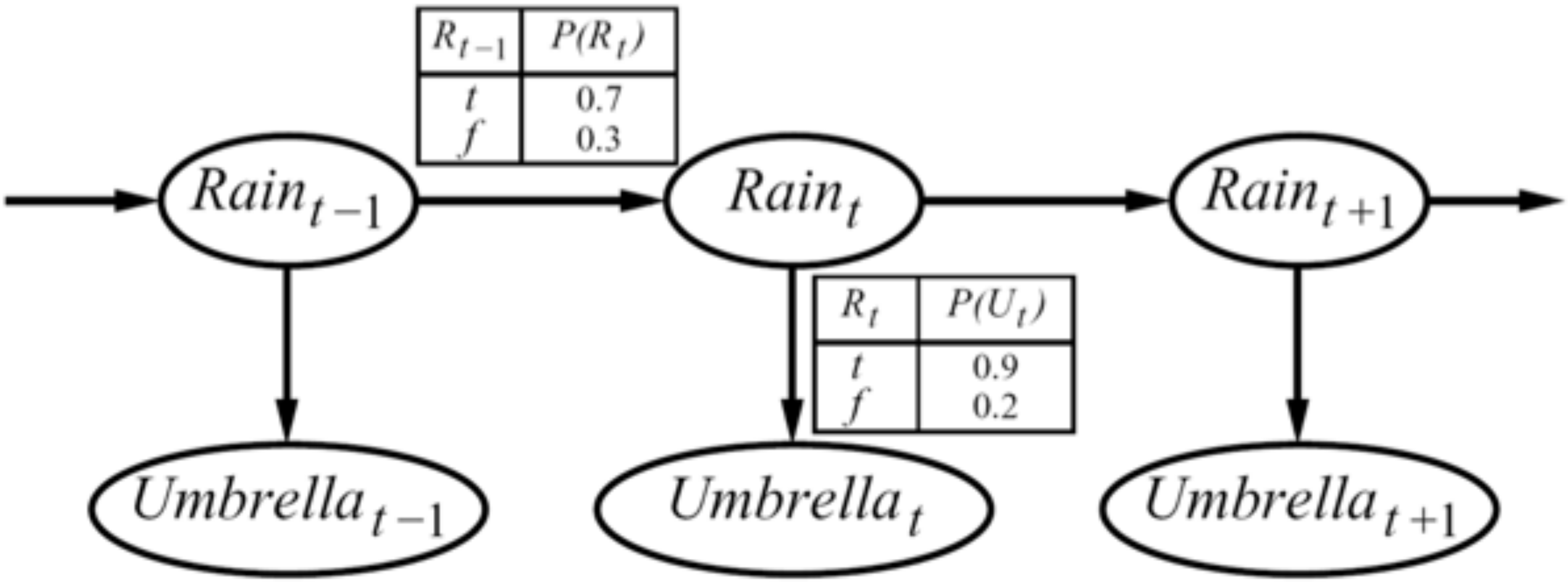
$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})$$

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

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

- To get $B_t(X)$, compute each entry and normalize

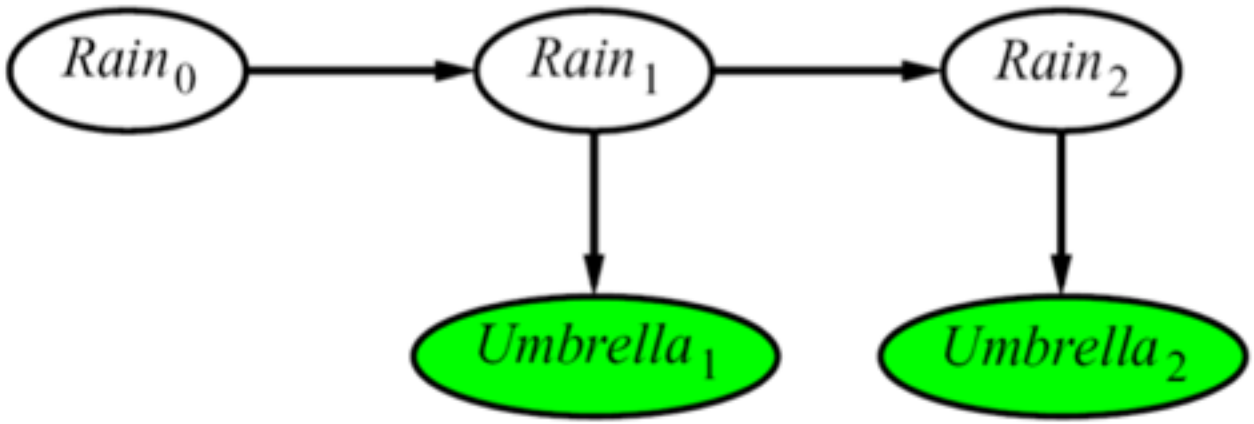
Example: Run the Filter



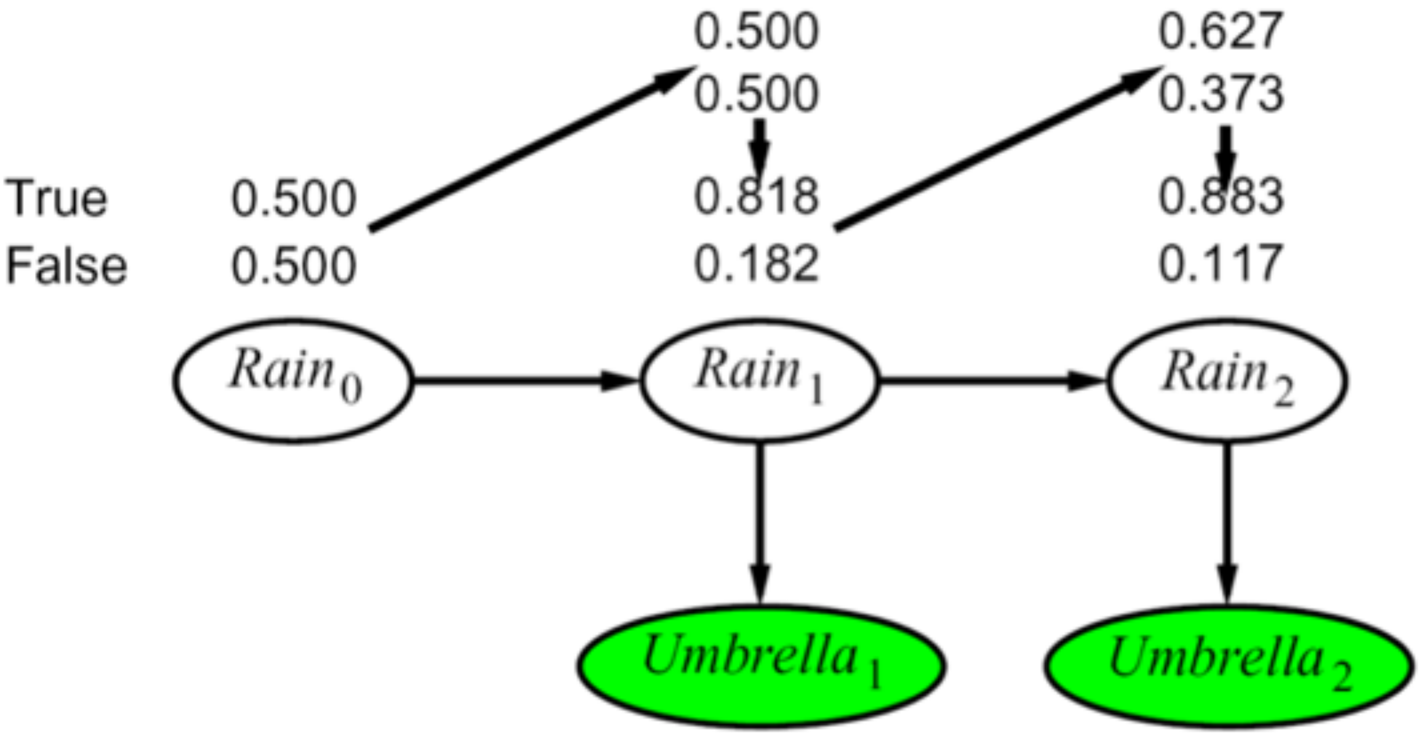
- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Example HMM

True	0.500
False	0.500

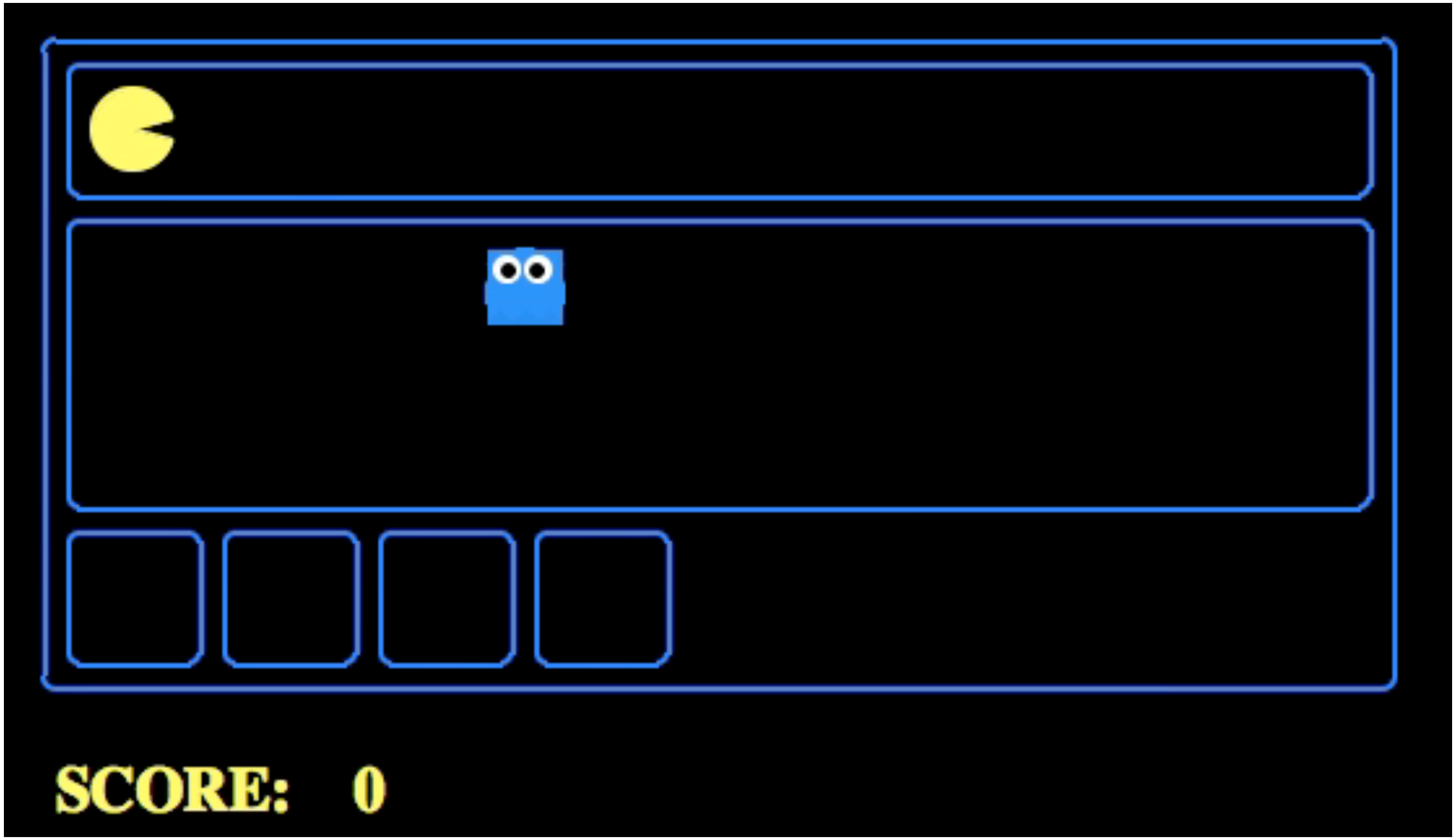


Example HMM



0.500
0.500
0.627
0.373
True
False
0.500
0.500
0.818
0.182
0.883
0.117
$Rain_0$
$Rain_1$
$Rain_2$
$Umbrella_1$
$Umbrella_2$

Example Pac-man



Summary: Filtering

- Filtering is the inference process of finding a distribution over X_T given e_1 through e_T : $P(X_T \mid e_{1:t})$
- We first compute $P(X_1 \mid e_1)$: $P(x_1|e_1) \propto P(x_1) \cdot P(e_1|x_1)$
- For each t from 2 to T, we have $P(X_{t-1} \mid e_{1:t-1})$
- **Elapse time:** compute $P(X_t \mid e_{1:t-1})$

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

- **Observe:** compute $P(X_t \mid e_{1:t-1} , e_t) = P(X_t \mid e_{1:t})$

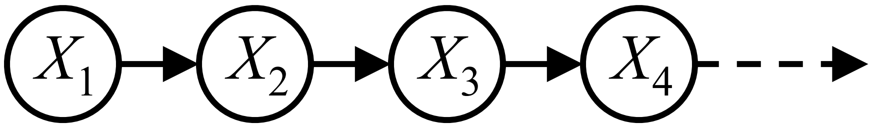
$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

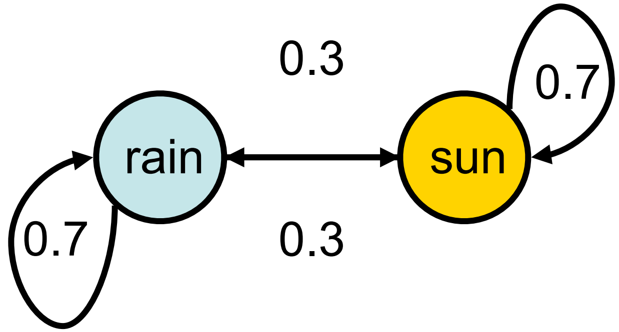
Filtering Complexity

- Problem size:
  - |X| states, |E| observations, T time steps
- Each Time Step
  - Computation: $O(|X|^2)$
  - Space: $O(|X|)$
- Total
  - Computation: $O(|X|^2T)$
  - Space: $O(|X|)$

Recap: Reasoning Over Time

- Stationary Markov models

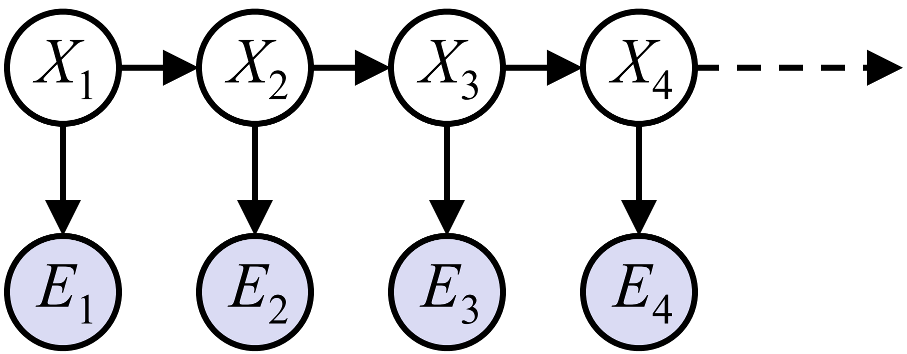




$P(X_1)$ $P(X|X_{-1})$

$P(E|X)$

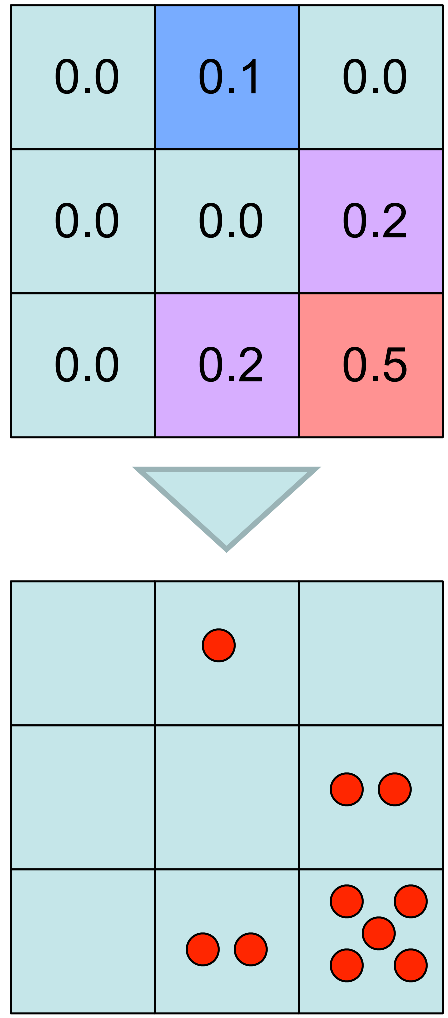
- Hidden Markov models



X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

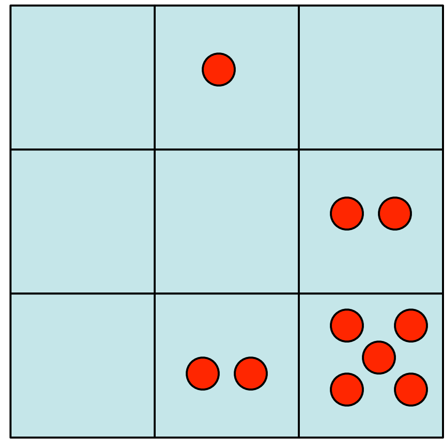
Particle Filtering

- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
  - $|X|^2$ may be too big to do updates
- 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



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 will 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)
(2,1)
(3,3)
(3,3)
(2,1)

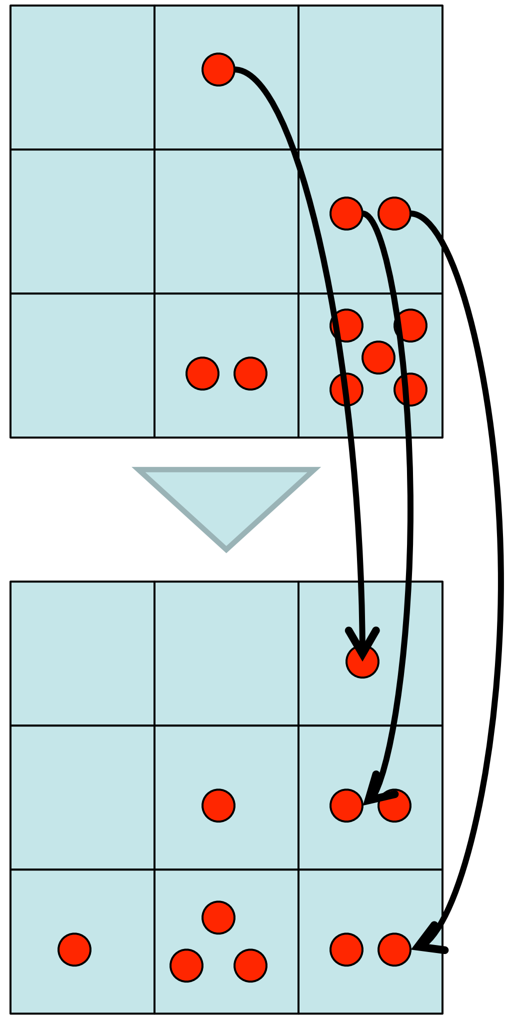
Particle Filtering: Elapse Time

- 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 probs
  - Here, most samples move clockwise, but some move in another direction or stay in place

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



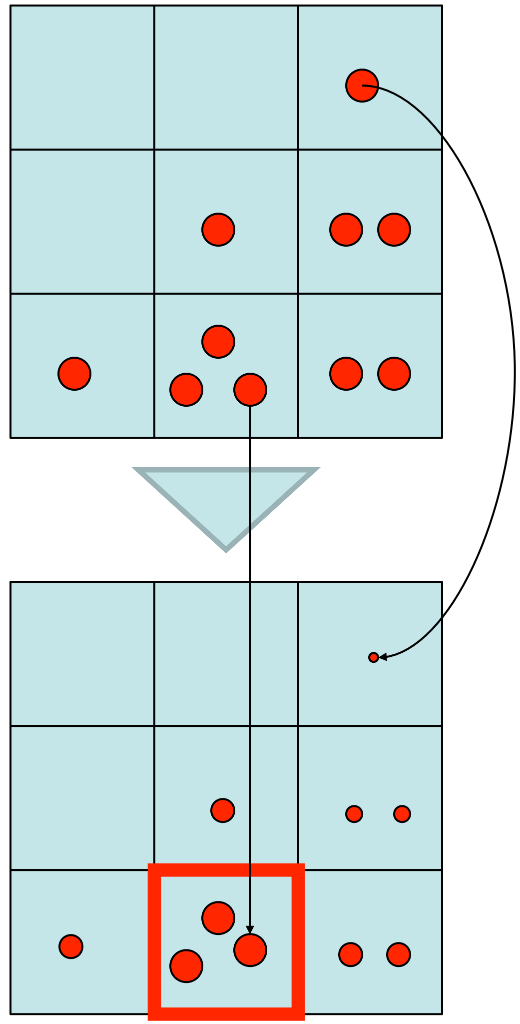
Particle Filtering: Observe

- Slightly trickier:
  - Don't do rejection sampling (why not?)
  - We don't sample the observation, we fix it
  - This is similar to likelihood weighting, so we downweight our samples based on the evidence

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

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

  - Note that, as before, the probabilities don't sum to one, since most have been downweighted (in fact they sum to an approximation of P(e))

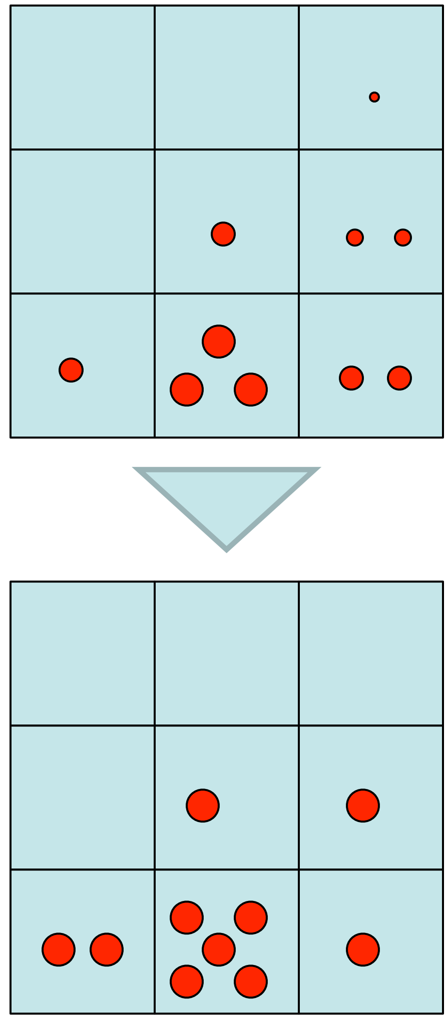


Particle Filtering: Resample

- 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

Old Particles:
(3,3) w=0.1
(2,1) w=0.9
(2,1) w=0.9
(3,1) w=0.4
(3,2) w=0.3
(2,2) w=0.4
(1,1) w=0.4
(3,1) w=0.4
(2,1) w=0.9
(3,2) w=0.3

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



Particle Filtering Summary

- Represent current belief P(X | evidence to date) as set of *n* samples (actual assignments X=x)
- For each new observation e:
  1. Sample transition, once for each current particle x

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

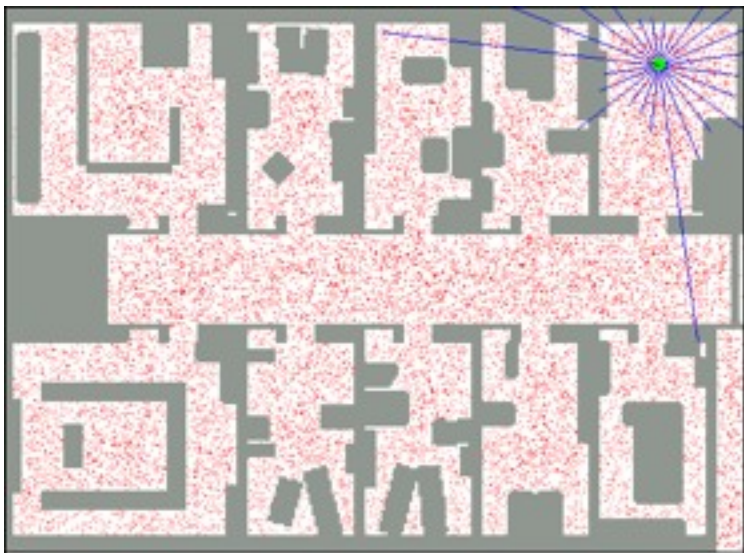
  2. For each new sample x', compute importance weights for the new evidence e:

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

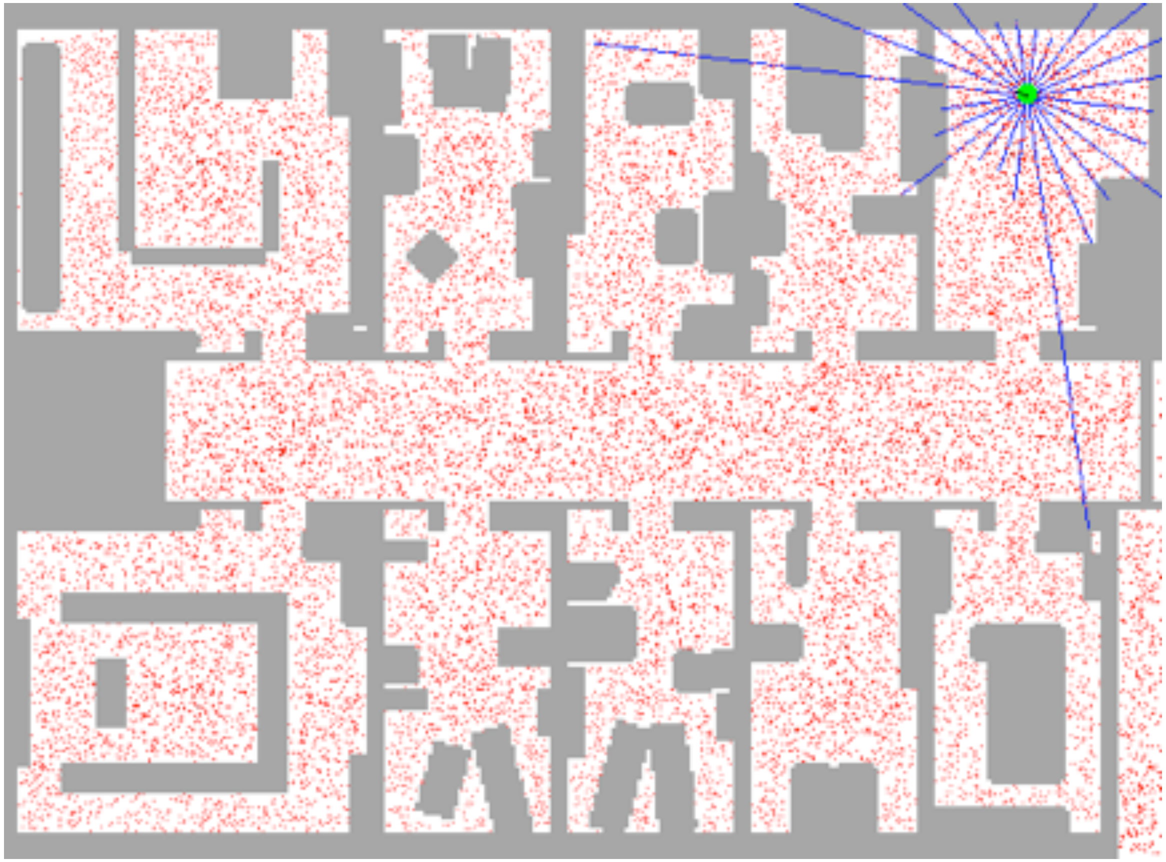
  3. Finally, normalize the importance weights and resample N new particles

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

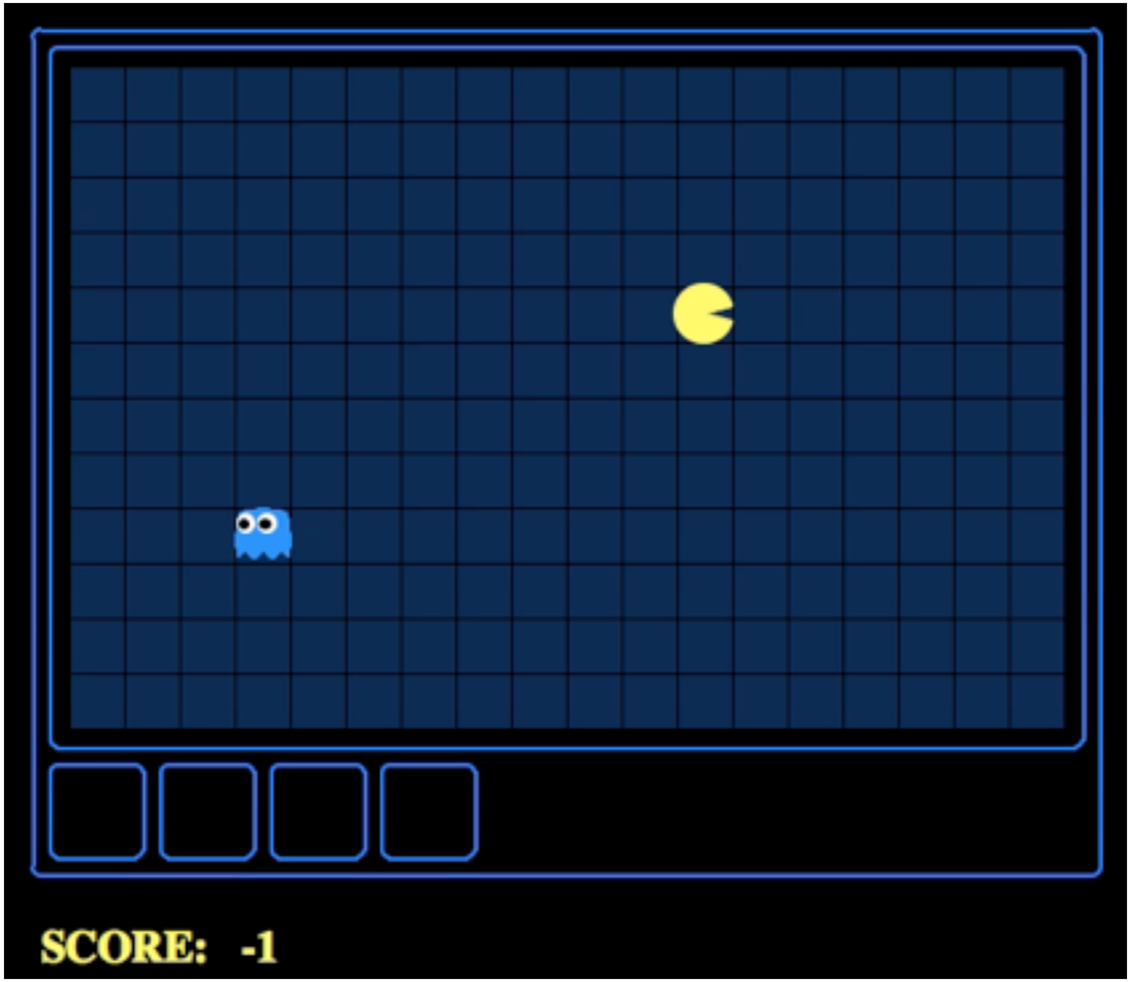


Robot Localization



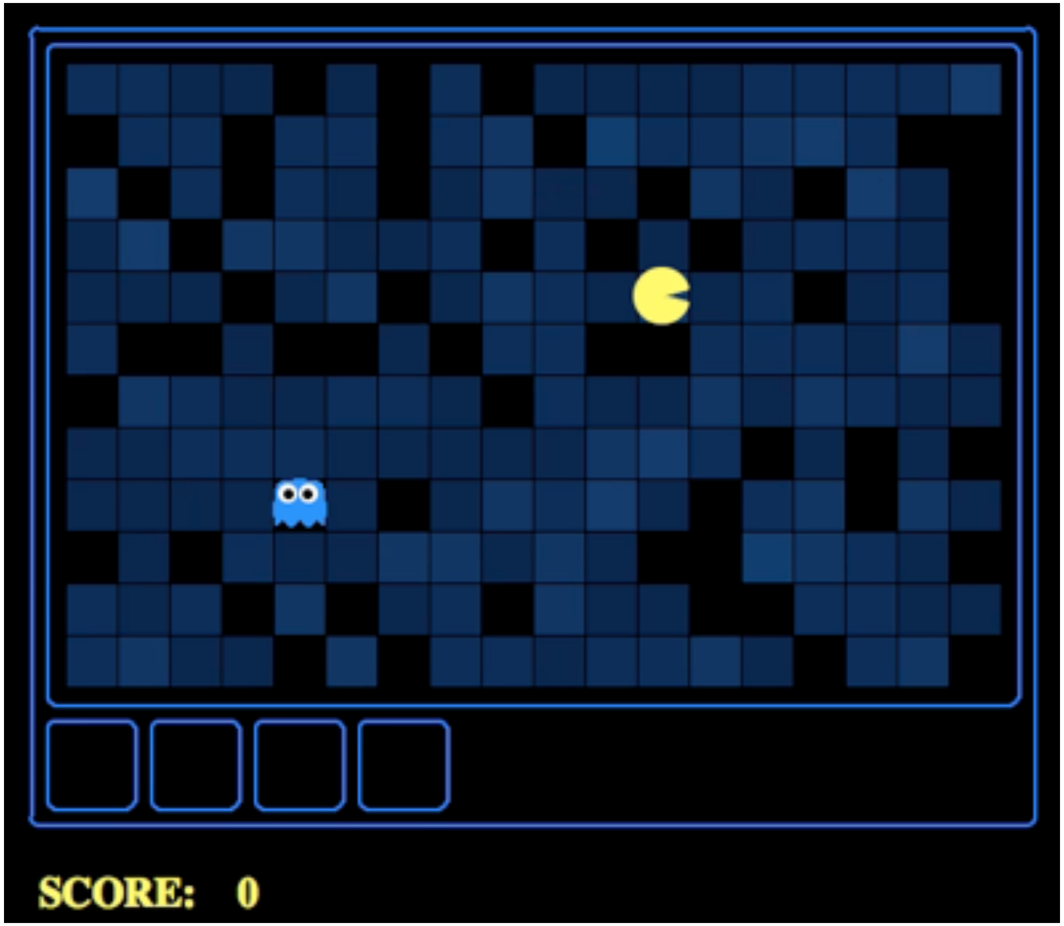
Which Algorithm?

Exact filter, uniform initial beliefs



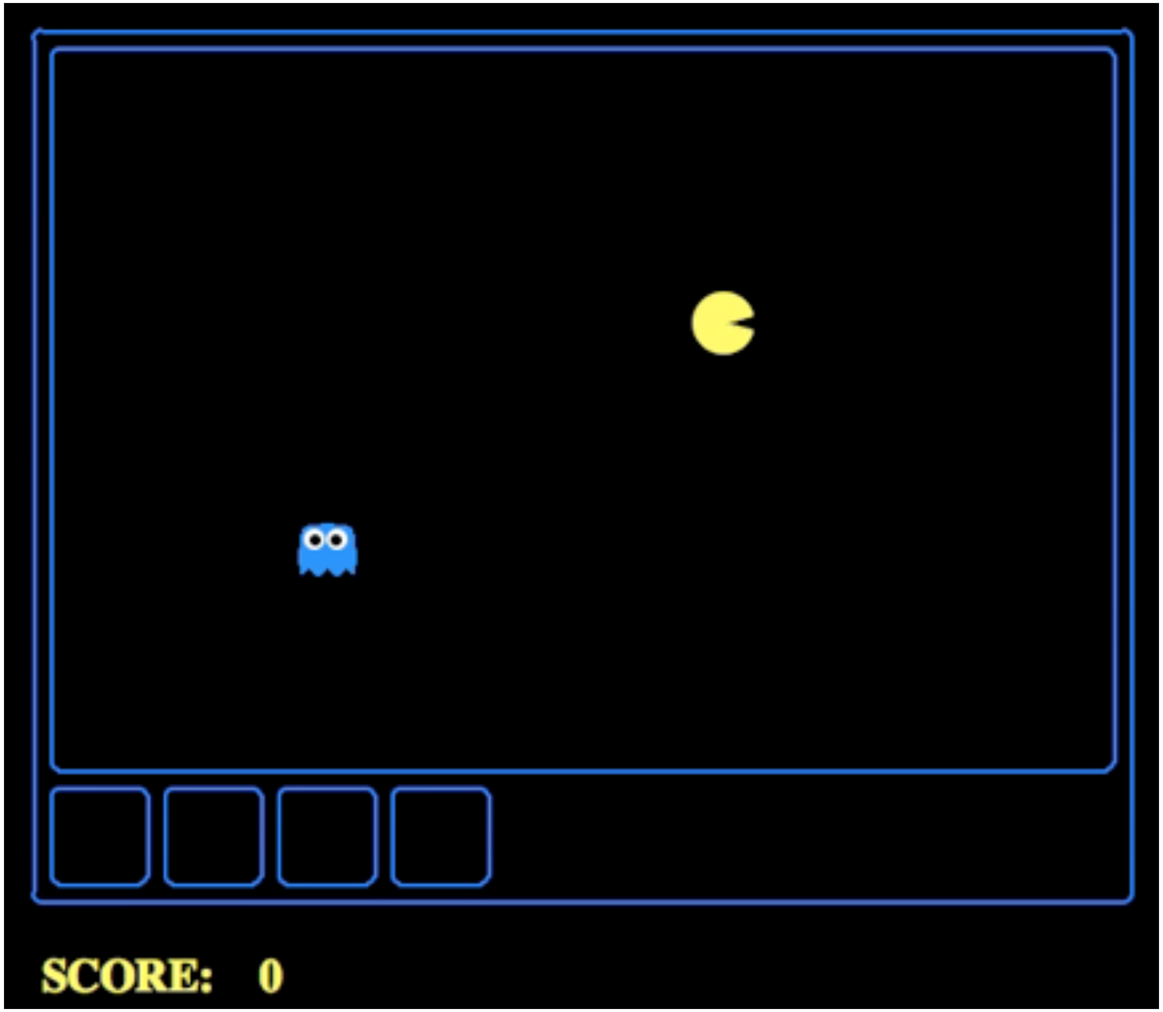
Which Algorithm?

Particle filter, uniform initial beliefs, 300 particles



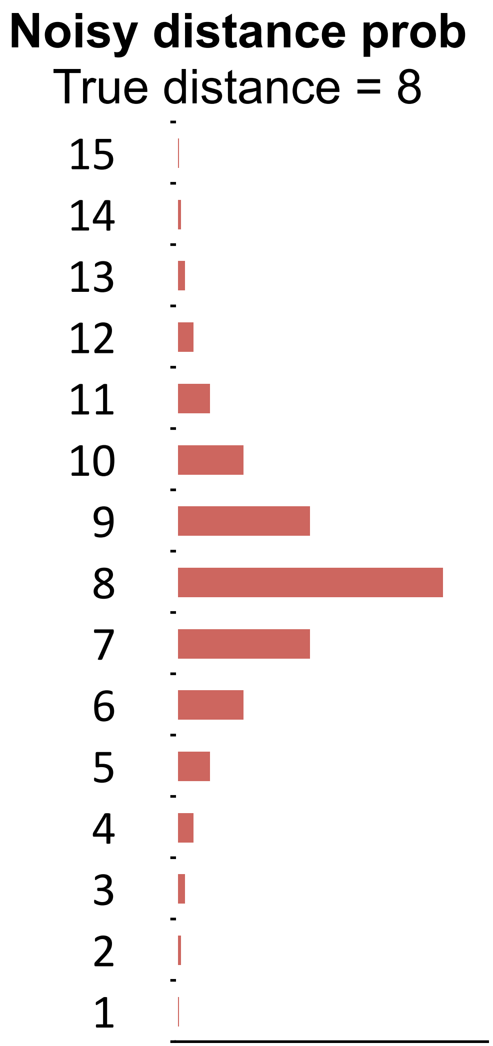
Which Algorithm?

Particle filter, uniform initial beliefs, 25 particles



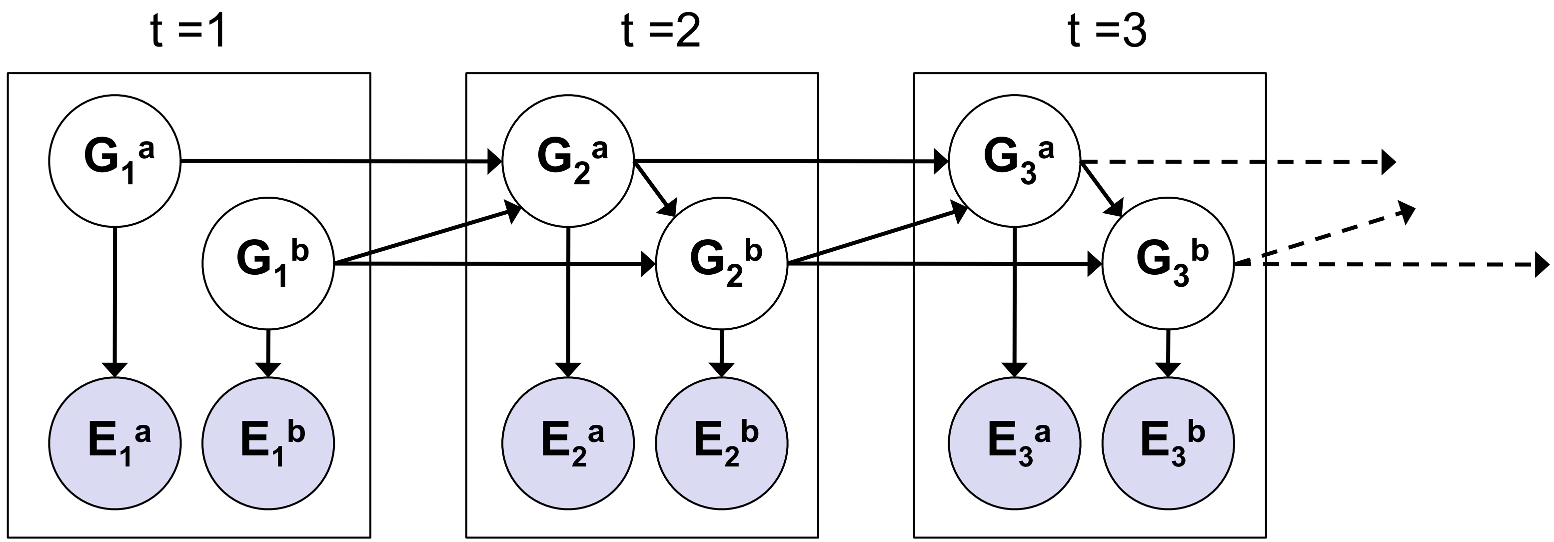
P4: Ghostbusters

- **Plot:** Pacman's grandfather, Grandpac, learned to hunt ghosts for sport.
- He was blinded by his power, but could hear the ghosts' banging and clanging.
- **Transition Model:** All ghosts move randomly, but are sometimes biased
- **Emission Model:** Pacman knows a "noisy" distance to each ghost



Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*



- Discrete valued dynamic Bayes nets are also HMMs

DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G_1^a}$ = (3,3) $\mathbf{G_1^b}$ = (5,3)

- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G_2^a}$ = (2,3) $\mathbf{G_2^b}$ = (6,3)
- **Observe**: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E_1^a} \mid \mathbf{G_1^a}) * P(\mathbf{E_1^b} \mid \mathbf{G_1^b})$

- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood