CSE 473: Artificial Intelligence

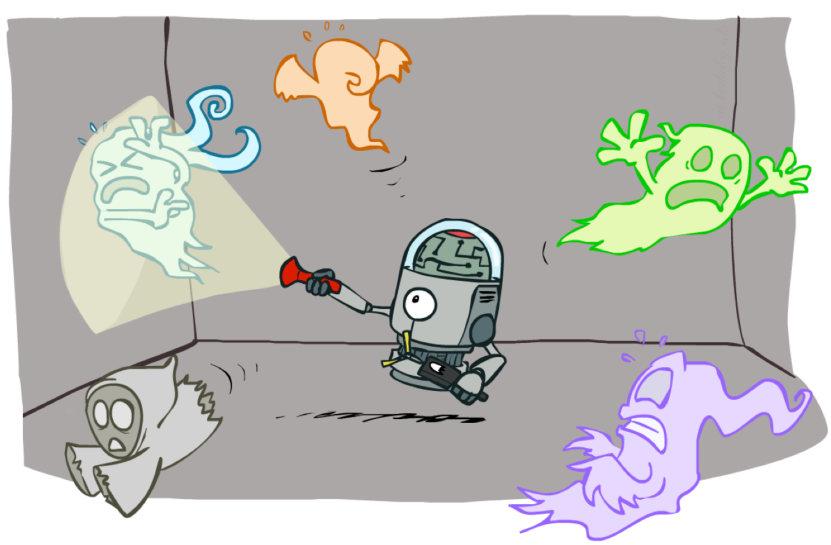
Hidden Markov Models

Daniel Weld

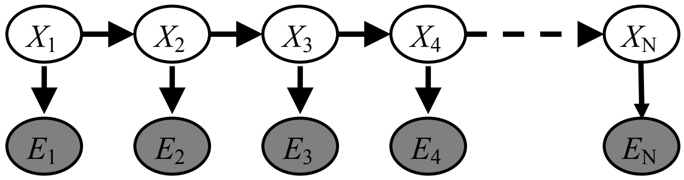
University of Washington

[Many of 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.]

Hidden Markov Models



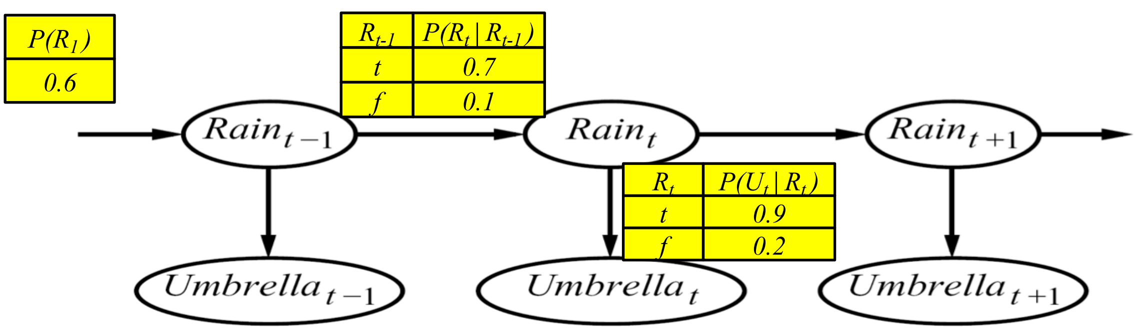
Hidden Markov Models



- Defines a joint probability distribution:

$$P(X_1, \ldots, X_n, E_1, \ldots, E_n) =$$
$$P(X_{1:n}, E_{1:n}) =$$
$$P(X_1)P(E_1|X_1)\prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)$$

Hidden Markov Model: Example

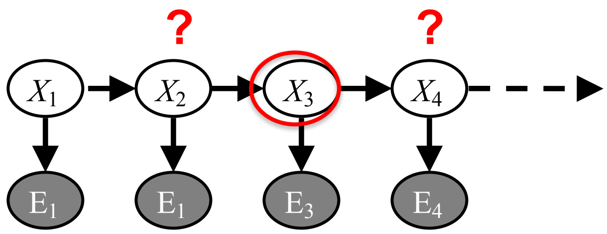


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Conditional Independence

HMMs have two important independence properties:

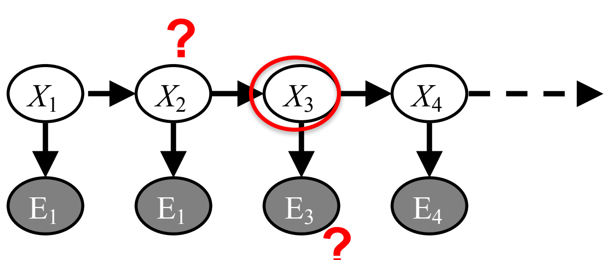
- **Future independent of past given the present**



Conditional Independence

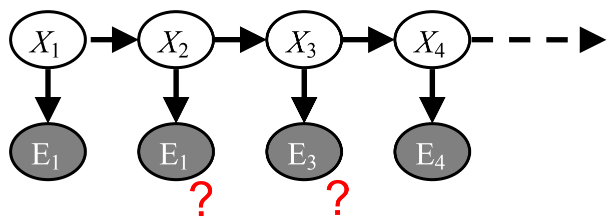
HMMs have two important independence properties:

- Future independent of past given the present
- **Current observation independent of all else given current state**



Conditional Independence

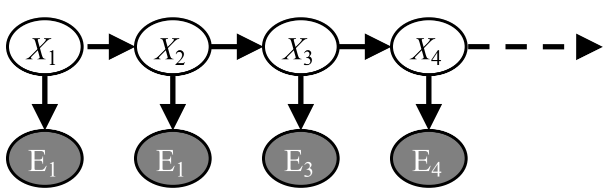
- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
  - Current observation independent of all else given current state



- Quiz: does this mean that observations are ***independent*** given no evidence?
  - [No, correlated by the hidden state]

Ghostbusters HMM

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

Etc…

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

P(E|X)

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

Etc… (must specify for other distances)

HMM Computations

- Given
  - parameters
  - evidence $E_{1:n} = e_{1:n}$
- Inference problems include:
  - Filtering, find $P(X_t|e_{1:t})$ for some t
  - Most probable explanation, for some t find

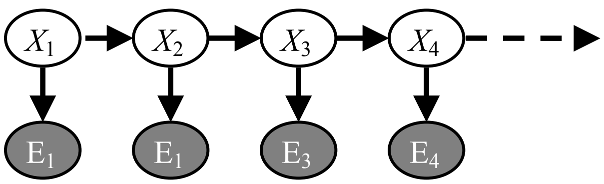
    $x^*_{1:t} = \text{argmax}_{x_{1:t}} P(x_{1:t}|e_{1:t})$
  - Smoothing, find $P(X_t|e_{1:n})$ for some $t < n$

Filtering (aka Monitoring)

- **The task of tracking the agent's belief state, B(x), over time**
  - B(x) is a distribution over world states – repr agent knowledge
  - We start with B(X) in an initial setting, usually uniform
  - As time passes, or we get observations, we update B(X)
- **Many algorithms for this:**
  - Exact probabilistic inference
  - Particle filter approximation
  - Kalman filter (a method for handling continuous Real-valued random vars)
    - invented in the 60'for Apollo Program – real-valued state, Gaussian noise

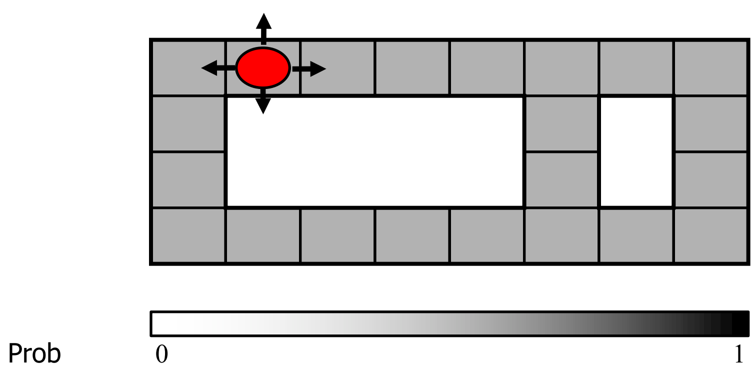
HMM Examples

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



Example: Robot Localization

Example from Michael Pfeiffer

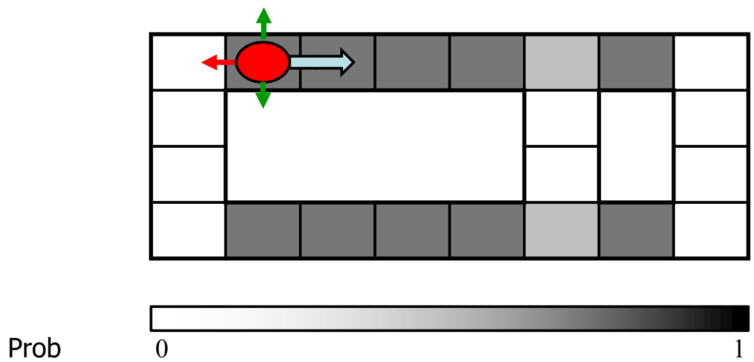


T=1

Sensor model: never more than 1 mistake

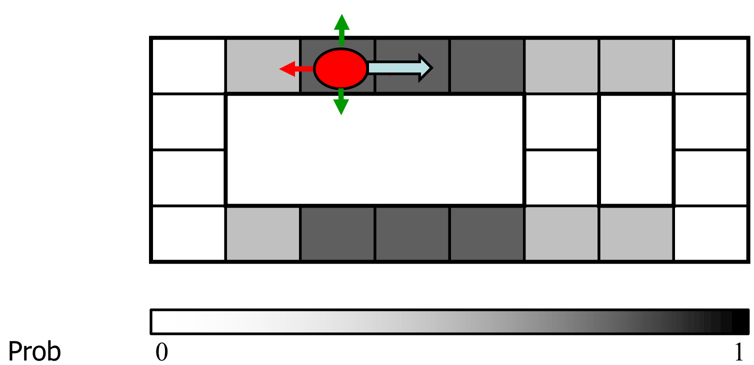
Motion model: may not execute action with small prob.

Example: Robot Localization



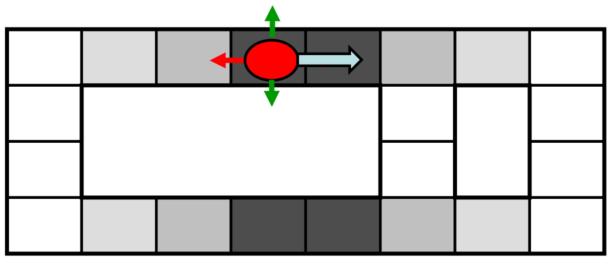
t=1

Example: Robot Localization



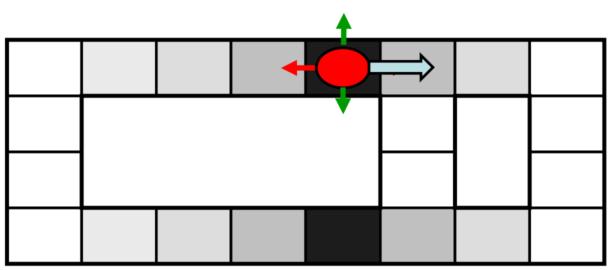
t=2

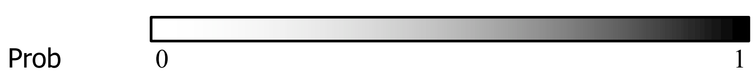
Example: Robot Localization



t=3

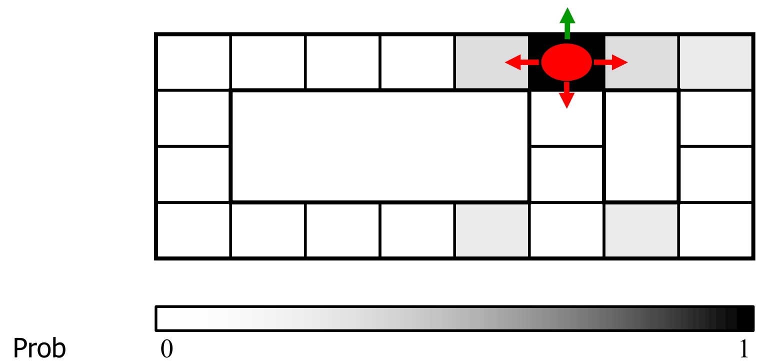
Example: Robot Localization





t=4

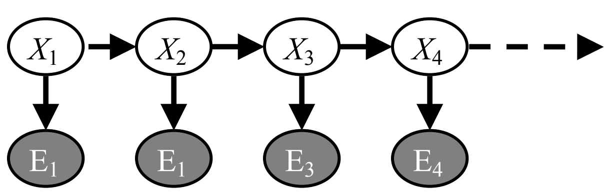
Example: Robot Localization



t=5

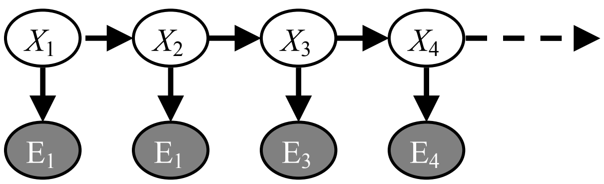
Other Real HMM Examples

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



Other Real HMM Examples

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



Filtering (aka Monitoring)

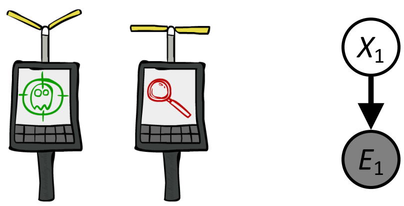
- Filtering, or monitoring, is the task of tracking the distribution B(X) (called "the belief state") over time

- We start with $B_0(X)$ in an initial setting, usually uniform

- We update $B_t(X)$ *computing $B_{t+1}(X)$*
  1. As time passes, and *using prob model of how ghosts move*
  2. As we get observations *using prob model of how noisy sensors work*

Filtering: Base Cases

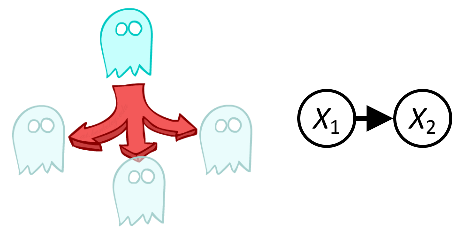
"Observation"



$$P(X_1|e_1)$$

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

"Passage of Time"



$$P(X_2)$$

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

Forward Algorithm

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

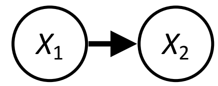
- t = 0
- $B(X_t)$ = initial distribution
- Repeat forever
  - $B'(X_{t+1})$ = Simulate passage of time from $B(X_t)$
  - Observe e_{t+1}
  - $B(X_{t+1})$ = Update $B'(X_{t+1})$ based on probability of e_{t+1}

23

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

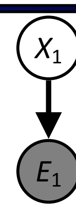
Observation

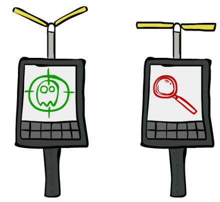
- Assume we have current belief P(X | previous evidence):

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

- Then, after evidence comes in:

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





- 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"

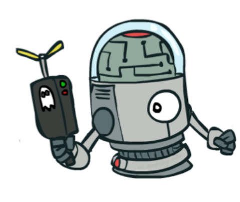
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

Before observation

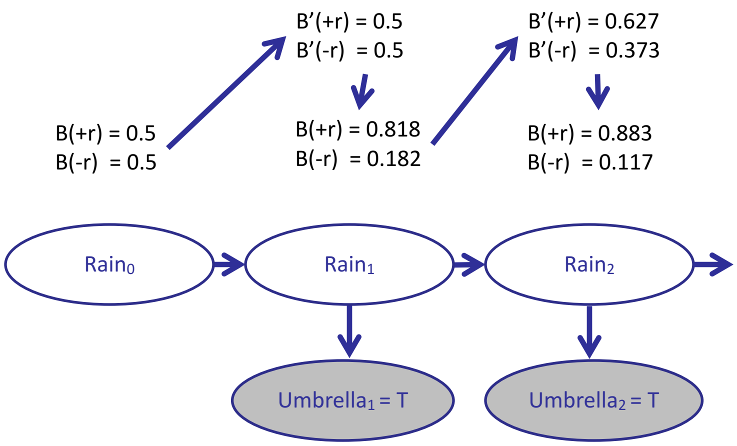
<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

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



Example: Weather HMM



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

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

Video of Demo Pacman – Sonar (with beliefs)

Summary: Online Belief Updates

Every time step, we start with current P(X | evidence)

1. We update for time:

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

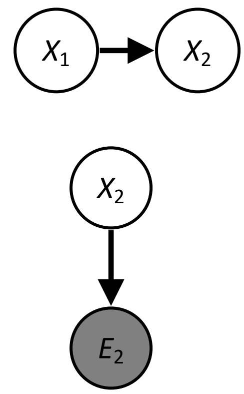
2. We update for evidence:

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

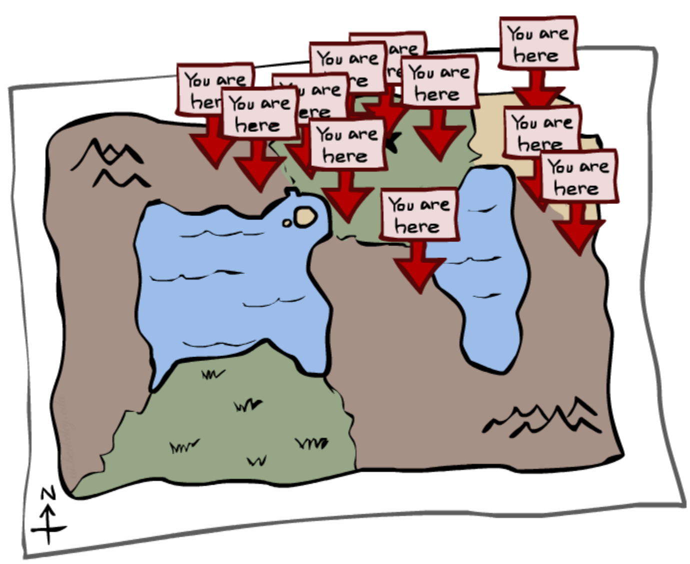
The forward algorithm does both at once (and doesn't normalize)

Computational complexity?

O(X^2 +XE) time & O(X+E) space



Particle Filtering



Particle Filtering Overview

- ***Approximation technique*** to solve filtering problem
- Represents P distribution with ***samples***
- Filtering still operates in two steps
  - Elapse time
  - Incorporate observations
    - (But this part has two sub-steps: weight & resample)

35

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
- Solution: approximate inference
  - Track ***samples of X***, not exact distribution of 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
- Particle is just new name for ***sample***
- This is how robot localization works in practice

Remember…

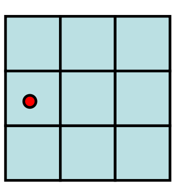
We'll start by looking at this

An HMM is defined by:

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

Here's a Single Particle

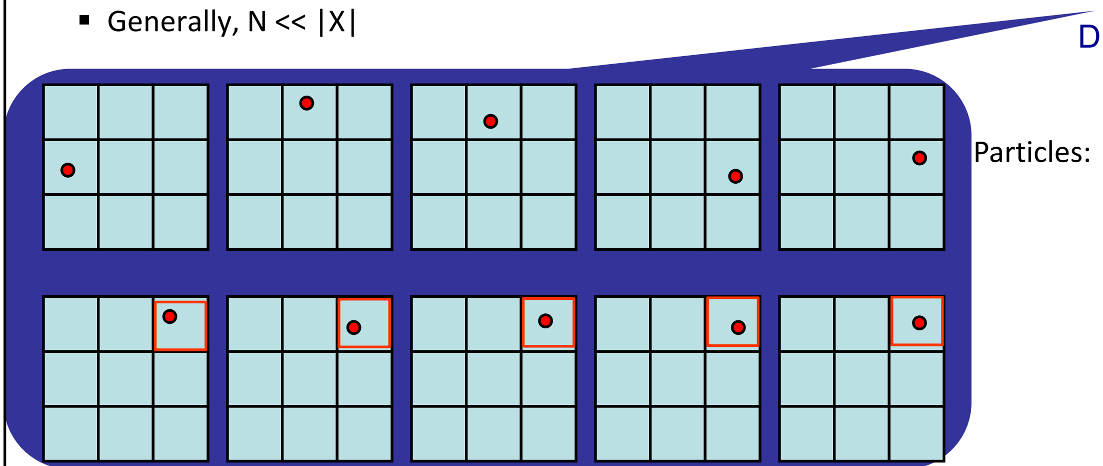
- It represents a hypothetical state where the robot is in (1,2)



Particles Approximate Distribution

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|

P(x) Distribution

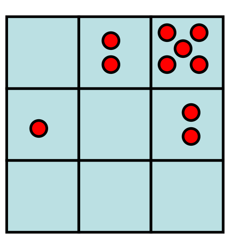


Particles: **(3,3)**
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

P(x=<3,3>) = 5/10 = 50%

Particle Filtering

A more compact view ***overlays*** the samples:

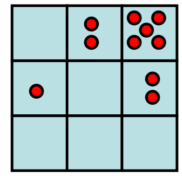


Encodes →

0.0	0.2	0.5
0.1	0.0	0.2
0.0	0.2	0.5

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 purpose



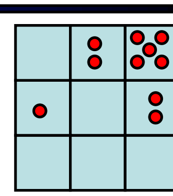
- P(x) approximated by **(number of particles with value x) / N**
  - More particles, more accuracy

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

- What is P((3,3))? 5/10 = 50%

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 purpose



- P(x) approximated by **(number of particles with value x) / N**
  - More particles, more accuracy

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

- What is P((2,2))? 0/10 = 0%

- In fact, many x may have P(x) = 0!