CS 188: Artificial Intelligence

Filtering and Applications

University of California, Berkeley

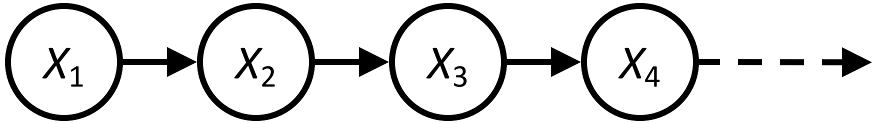
Today’s Topics

- Recap of Hidden Markov Models (HMMs) and **exact inference**
- Approximate Inference in HMMs via **Particle Filtering**
- **Applications** in Robot Localization and Mapping
- Brief overview of **Dynamic Bayes Nets**

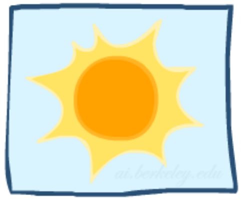
[Demo: Ghostbusters Markov Model (L15D1)]

Recap: Reasoning Over Time

- Markov models



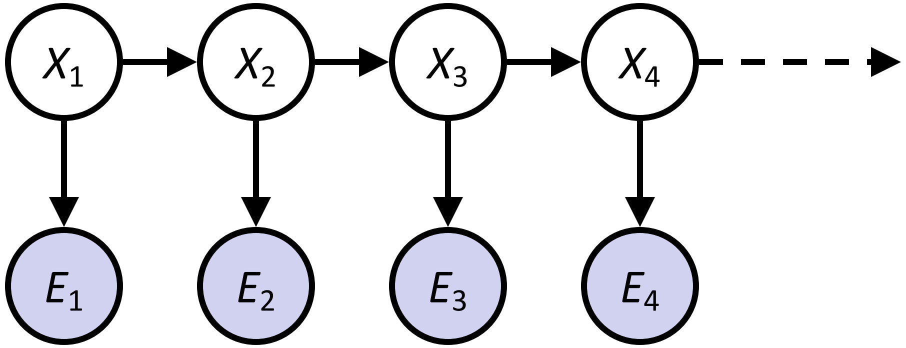
$P(X_1) \quad P(X_t|X_{t-1})$



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

X_{t-1}	X_t	P
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

- Hidden Markov models



$P(E|X)$

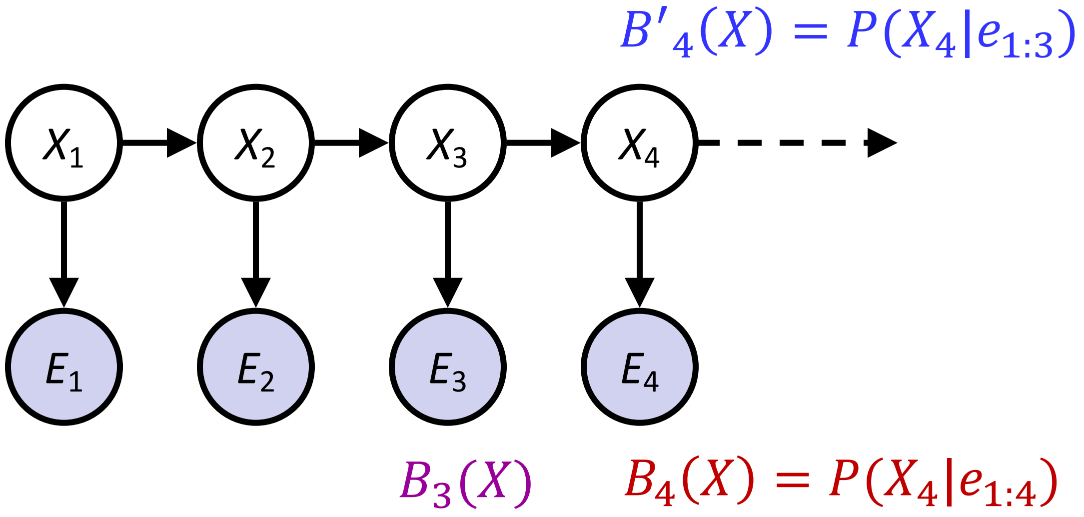
X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

HMM 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 $B_t(X)$ in terms of $B_{t-1}(X)$
  - Two steps: **Passage of Time** & **Observation**

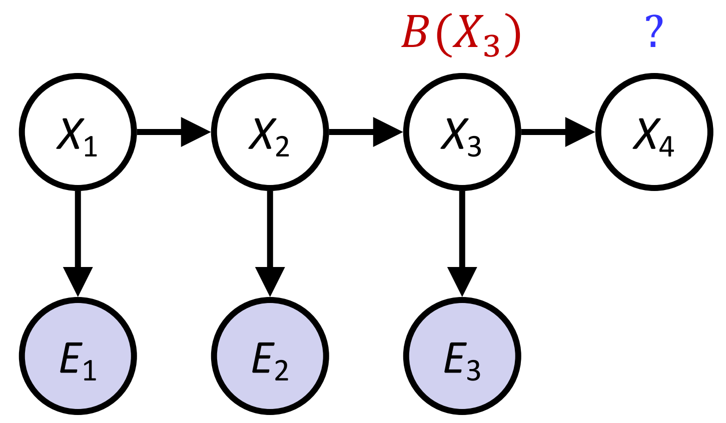


Passage of Time

- Assume we have current belief P(X | evidence to date) and transition prob.

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

Ex:



- 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:

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

Example: Passage of Time

- As time passes, uncertainty "accumulates"

(Transition model: ghosts usually go counter-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

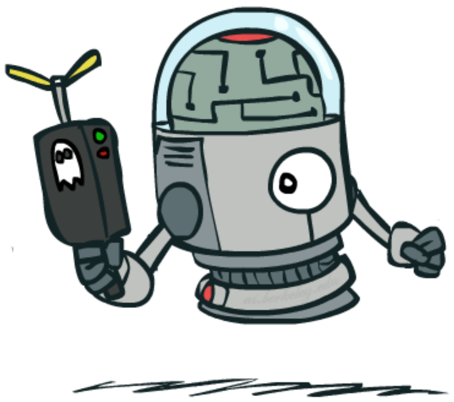
T = 1

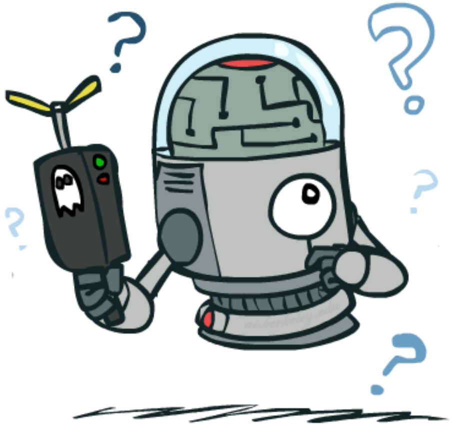
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

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 = 4

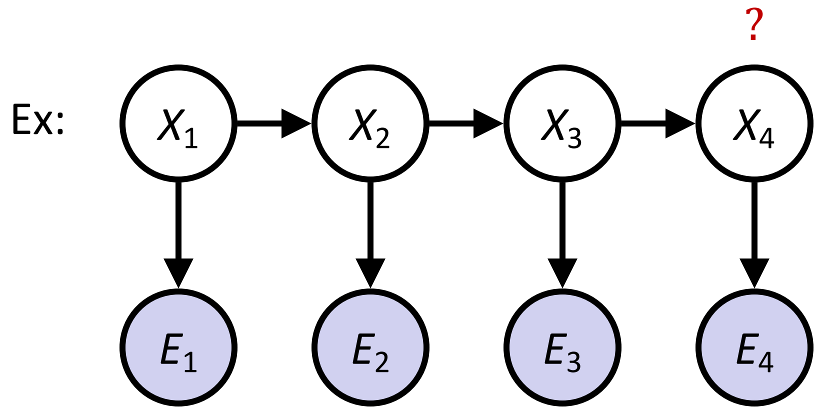




Observation

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

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



- Then, after evidence comes in:

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

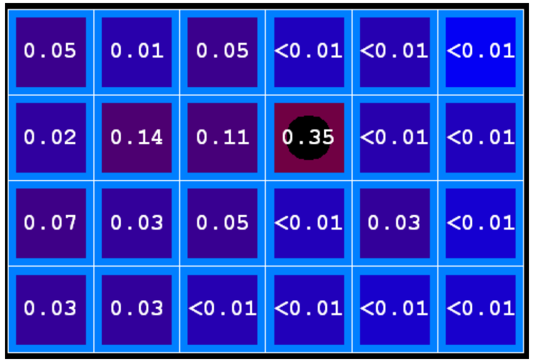
- Or, compactly:

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

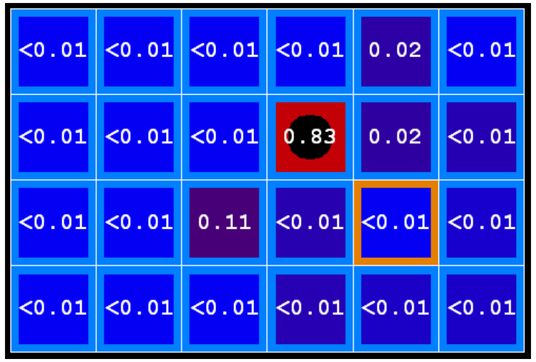
- Basic idea: beliefs “reweighted” by likelihood of evidence
- Unlike passage of time, we have to renormalize

Example: Observation

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

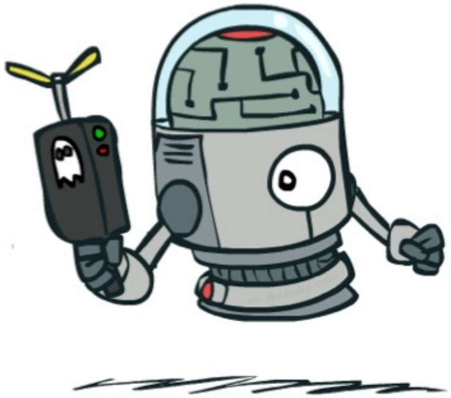


Before observation



After observation

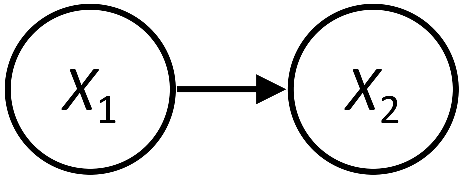
$$B(X) \propto P(e|X)B'(X)$$



Online Belief Updates

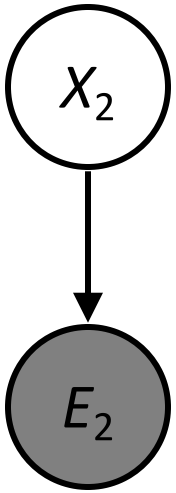
- Every time step, we start with current P(X | evidence)
- We update for time:

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



- We update for evidence:

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



- This is our updated belief $B_t(X) = P(X_t|e_{1:t})$
- 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

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

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

[Demo: Ghostbusters Exact Filtering (L15D2)]

Example: Weather HMM

B'(+r) = ?
B'(-r) = ?

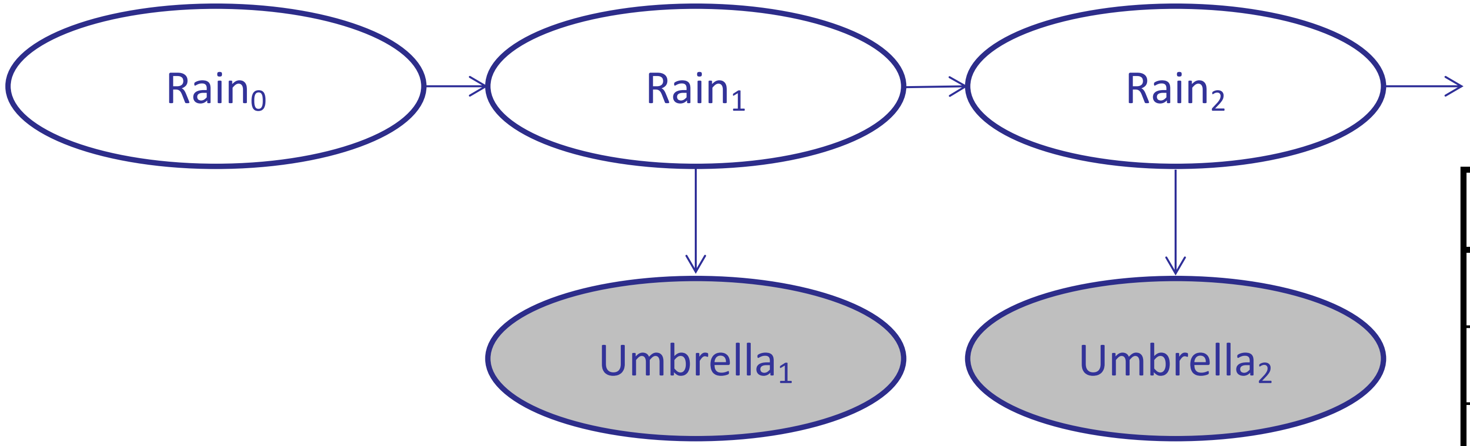
B(+r) = 0.5
B(-r) = 0.5

Passage of Time:

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

Observation:

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



$P(X_{t+1}|X_t)$

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

$P(E_t|X_t)$

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

Example: Weather HMM

$B'(+r) = 0.5*0.7 + 0.5*0.3 = 0.5$
$B'(-r) = 0.5*0.3 + 0.5*0.7 = 0.5$

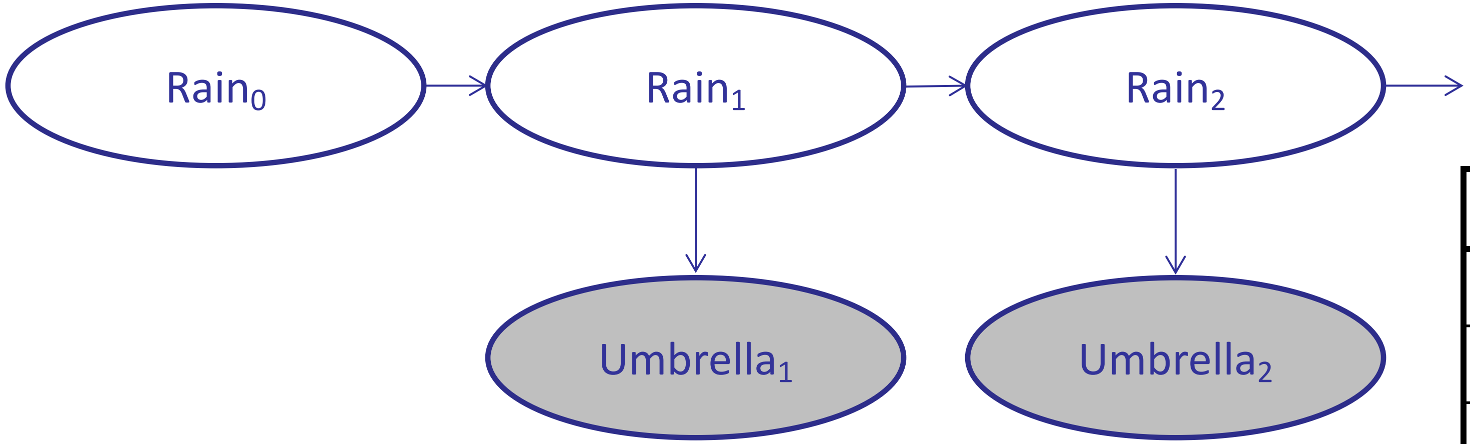
$B(+r) = 0.5$
$B(-r) = 0.5$

Passage of Time:

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

Observation:

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



$P(X_{t+1}|X_t)$

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

$P(E_t|X_t)$

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

Example: Weather HMM


$B(+r) = 0.5$
$B(-r) = 0.5$

$B'(+r) = 0.5$
$B'(-r) = 0.5$

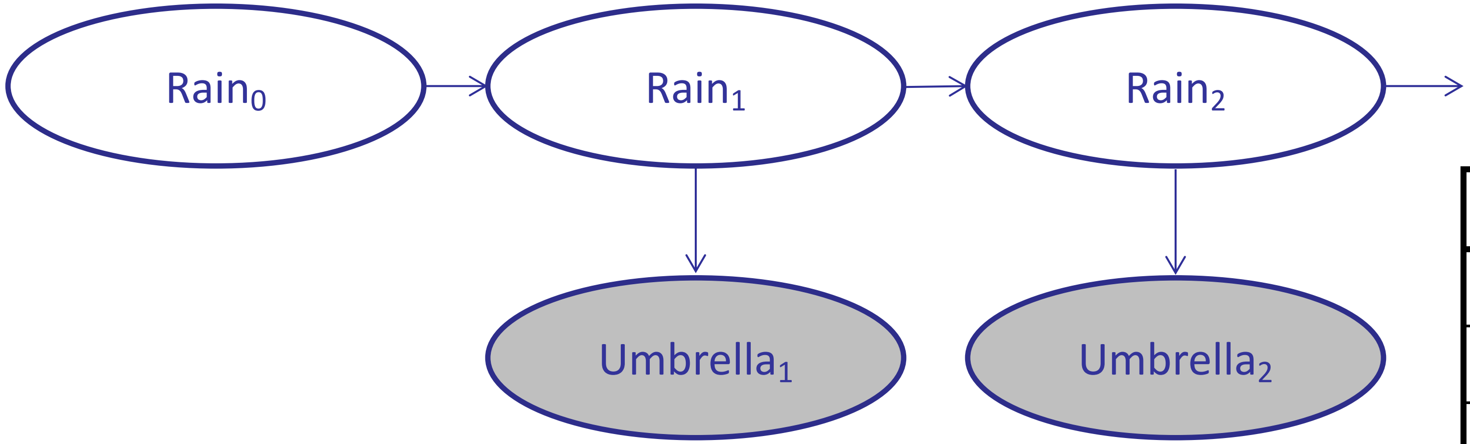
$B(+r) = ?$
$B(-r) = ?$

Passage of Time:

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

Observation:

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



$P(X_{t+1}|X_t)$

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

$P(E_t|X_t)$

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

Example: Weather HMM

$B(+r) = 0.5$
$B(-r) = 0.5$

$B'(+r) = 0.5$
$B'(-r) = 0.5$

$B(+r) = 0.9 * 0.5 = 0.45$
$B(-r) = 0.2 * 0.5 = 0.10$

normalize

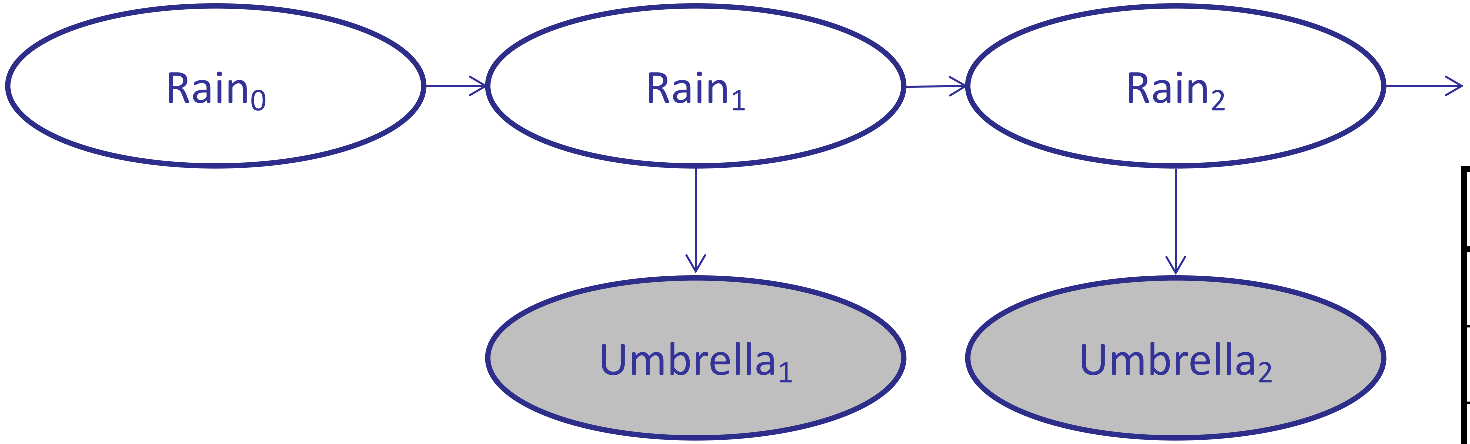
0.818
0.182

Passage of Time:

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

Observation:

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



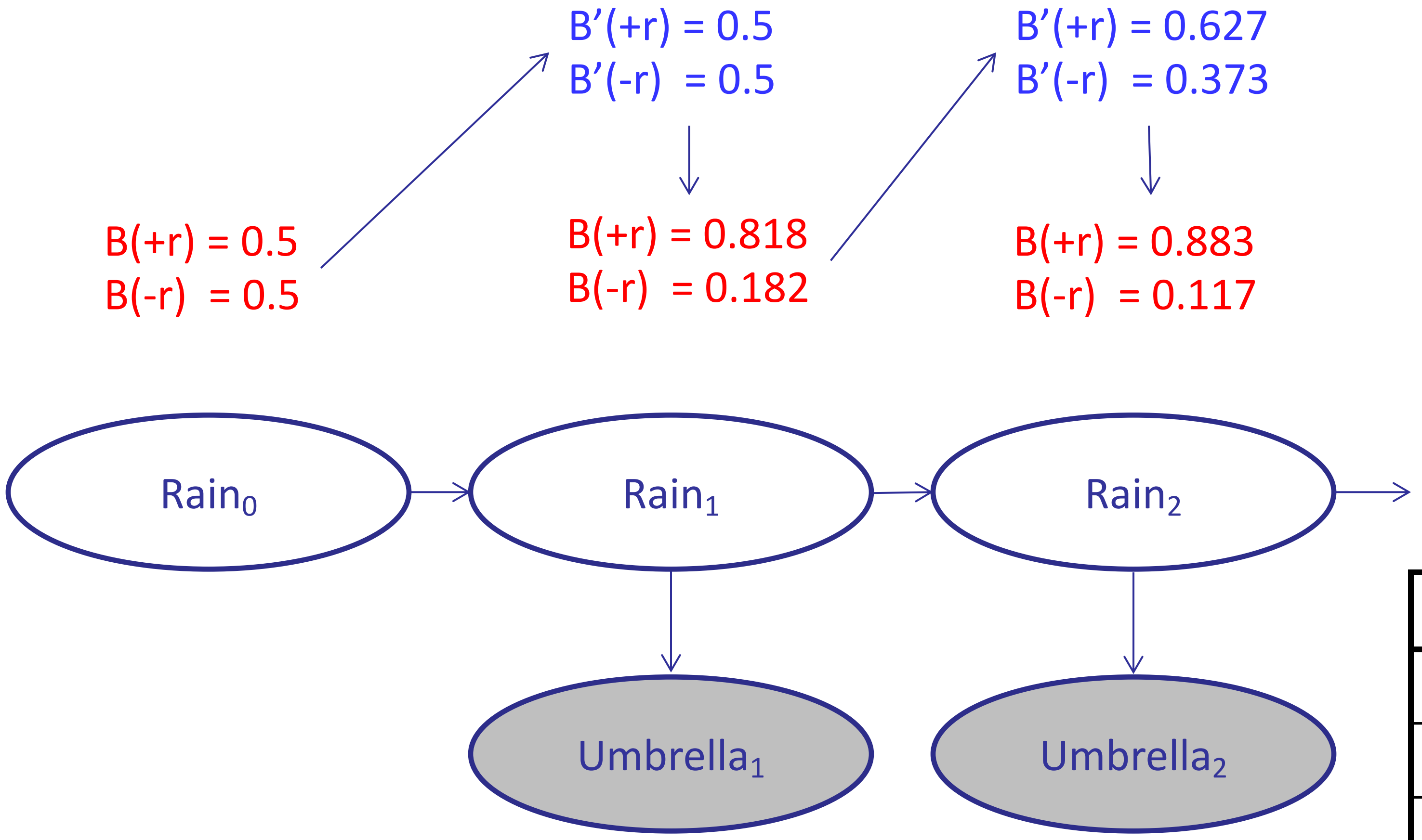
$P(X_{t+1}|X_t)$

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

$P(E_t|X_t)$

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

Example: Weather HMM



Passage of Time:

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

Observation:

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

$P(X_{t+1}|X_t)$

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

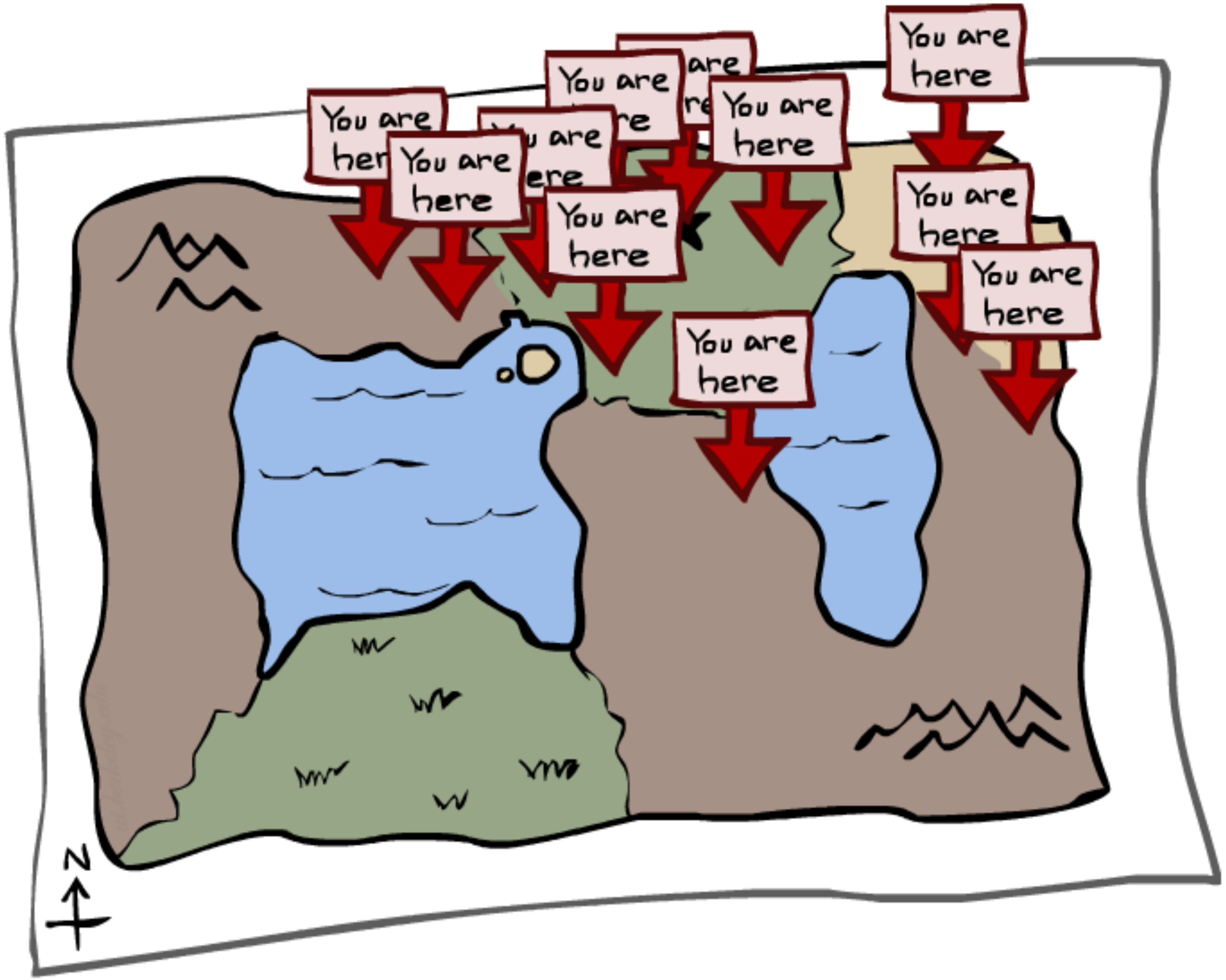
$P(E_t|X_t)$

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 Ghostbusters Filtering

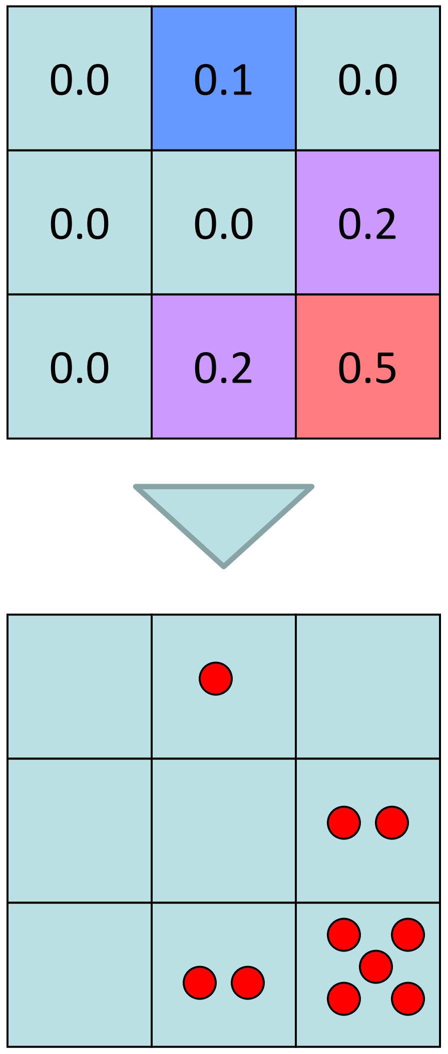
How can we support large state spaces?

Particle Filtering



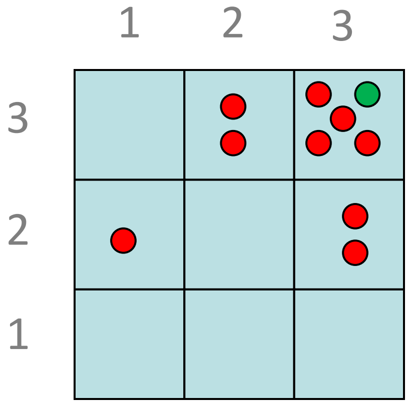
Particle Filtering

- 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

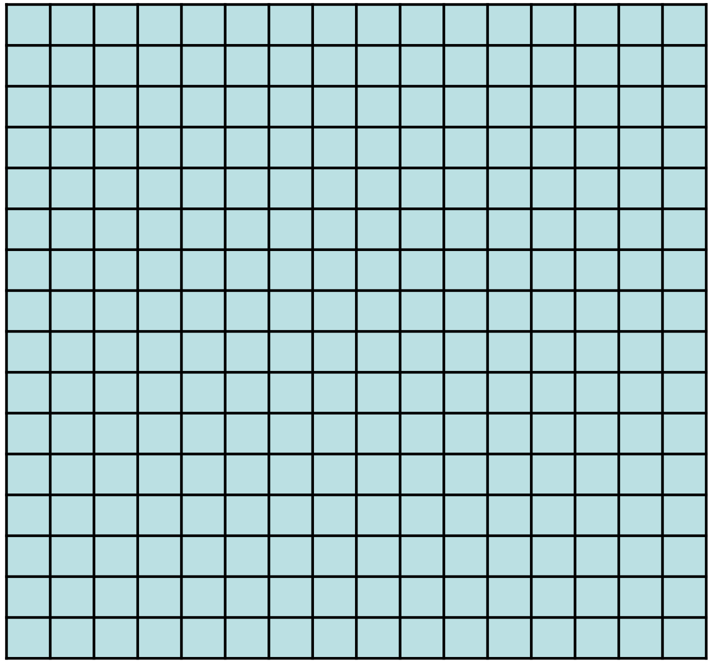


Representation: Particles

- 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
  - Example: if we want to infer location on 16x16 grid



Store 256 numbers:



VS

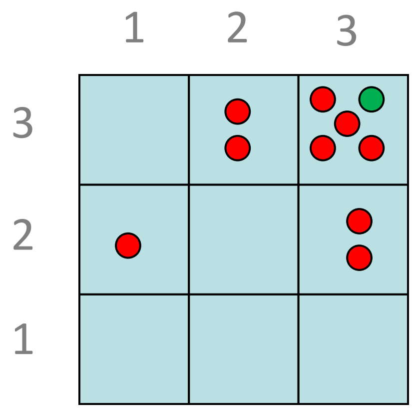
Store 10 numbers:

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

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

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

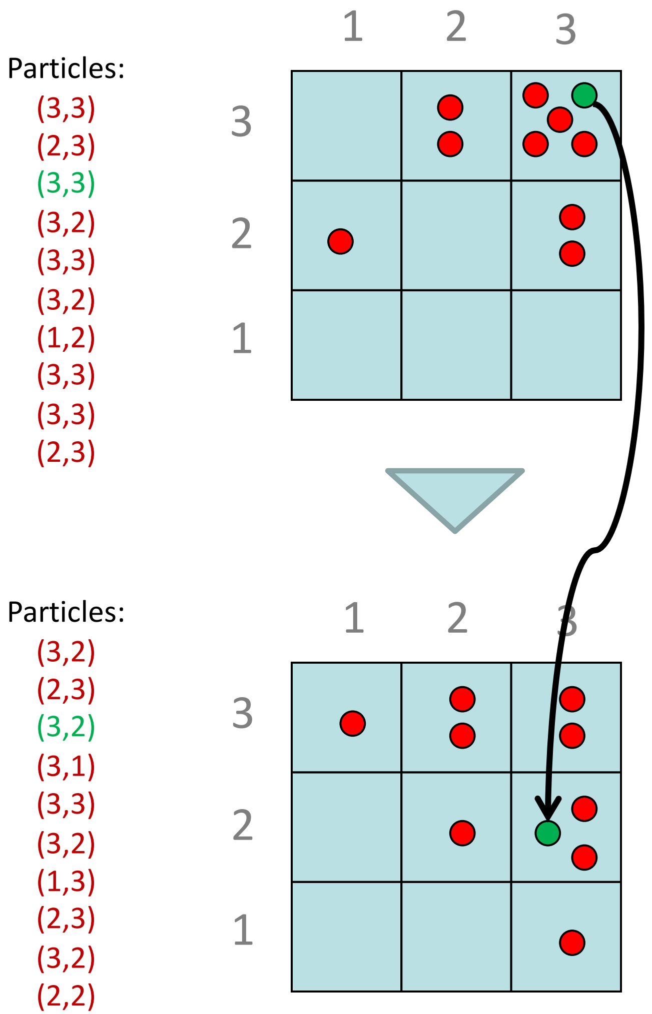


Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particle Filtering: Passage of Time

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

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



Particle Filtering: Passage of Time

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

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

For example:

sample(

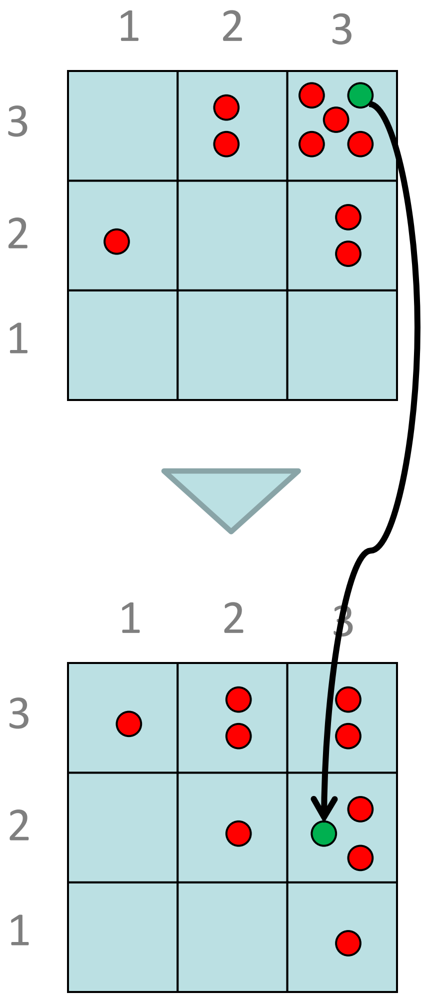
X’	P(X’|X=(3,3))
(3,2)	0.8
(3,3)	0.1
(2,3)	0.1

)

most likely returns (3,2) but may return (3,3) or (2,3)

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)



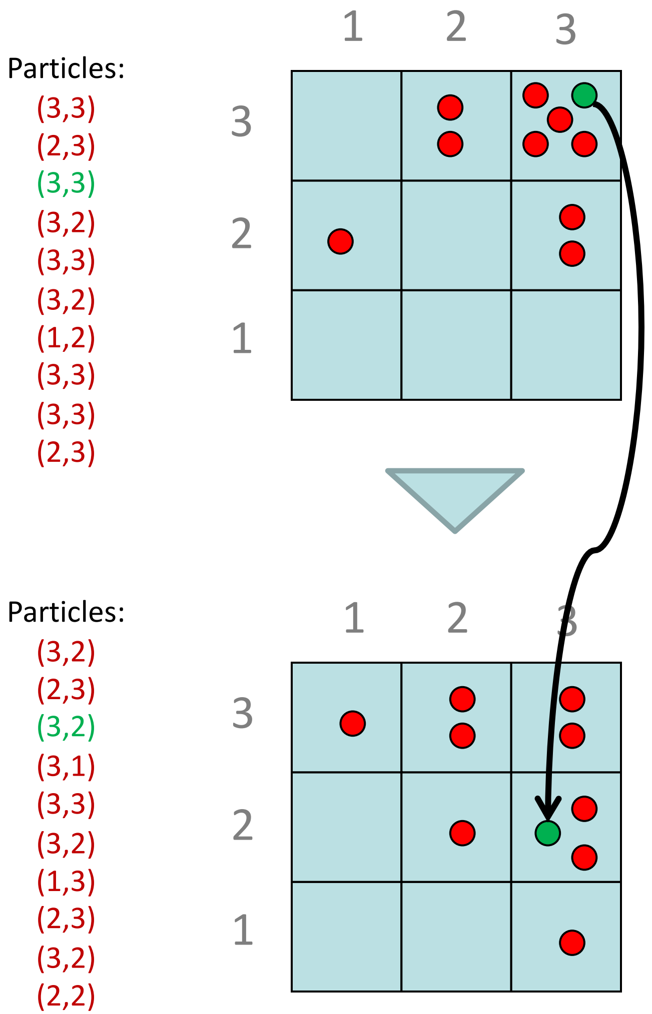
Particle Filtering: Passage of Time

- 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)



Particle Filtering: Observe

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

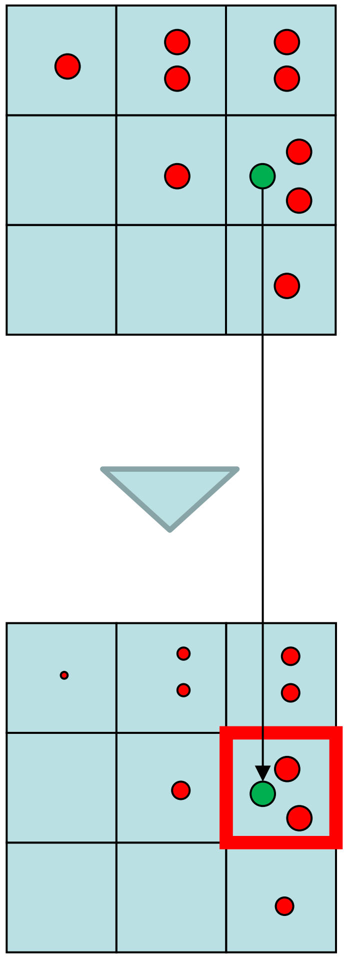
$$w(x) = P(e|x)$$

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

  - As before, the probabilities don't sum to one, since all have been down-weighted (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

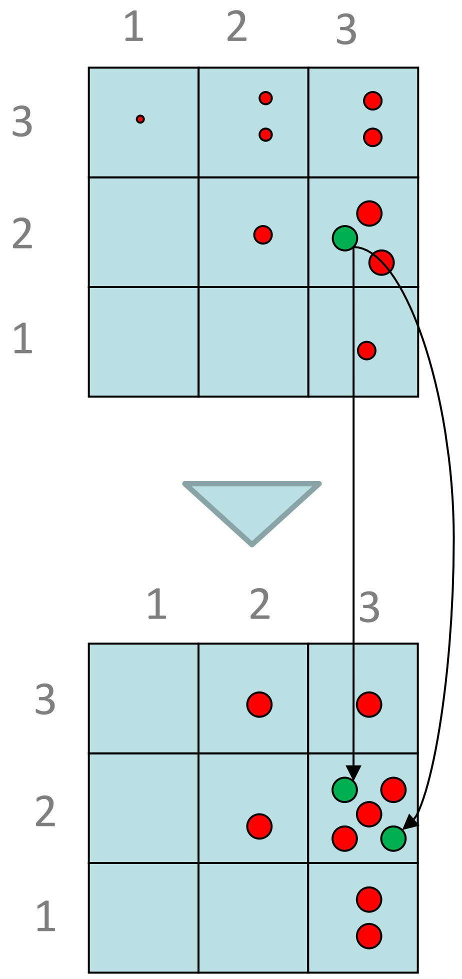


Particle Filtering: Resample

- 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

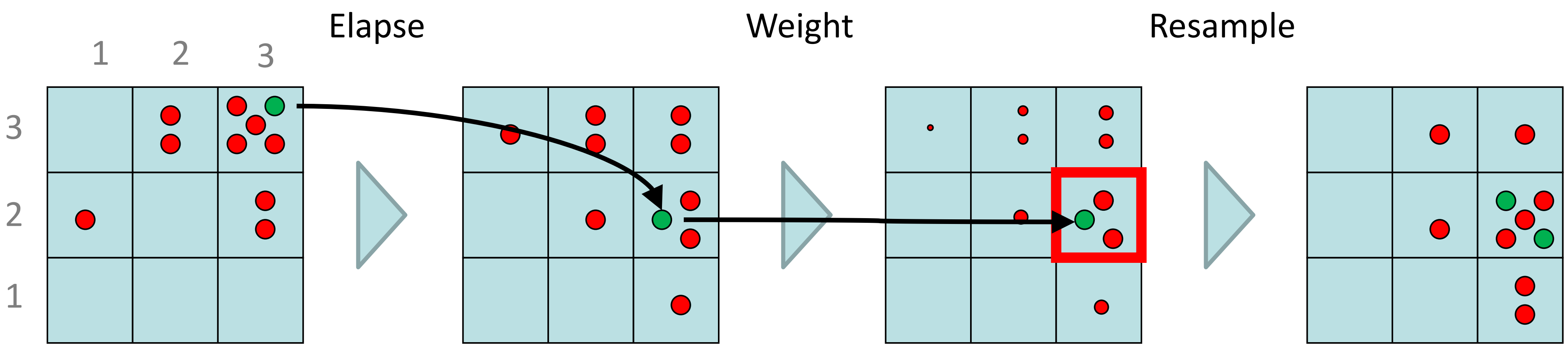
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

(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



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)

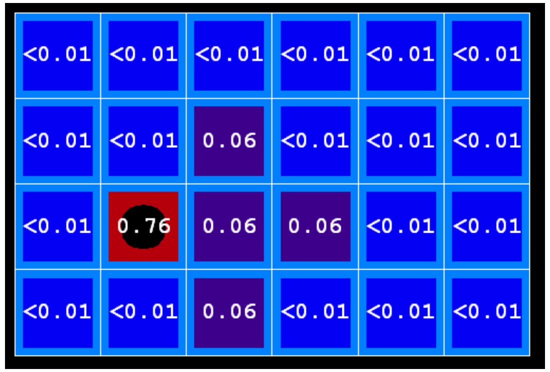
[Demos: ghostbusters particle filtering (L15D3,4,5)]

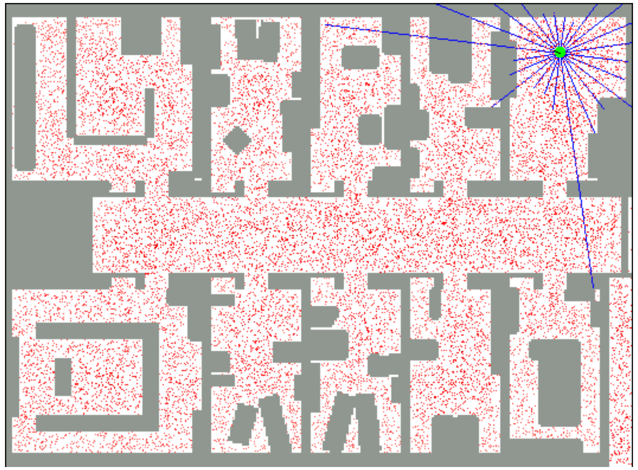
Video of Demo – Moderate Number of Particles

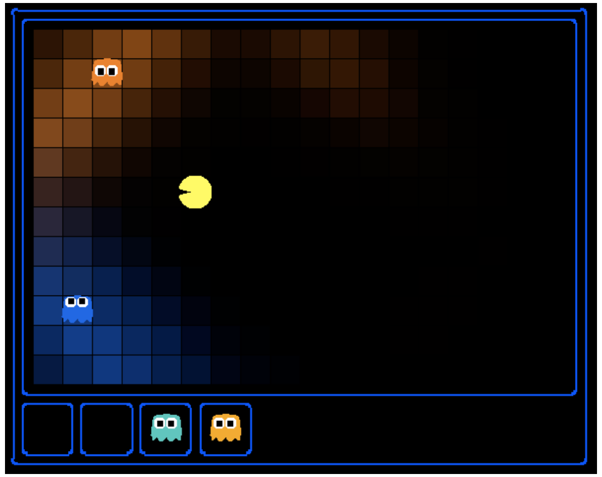
Video of Demo – One Particle

Video of Demo – Huge Number of Particles

More Demos!

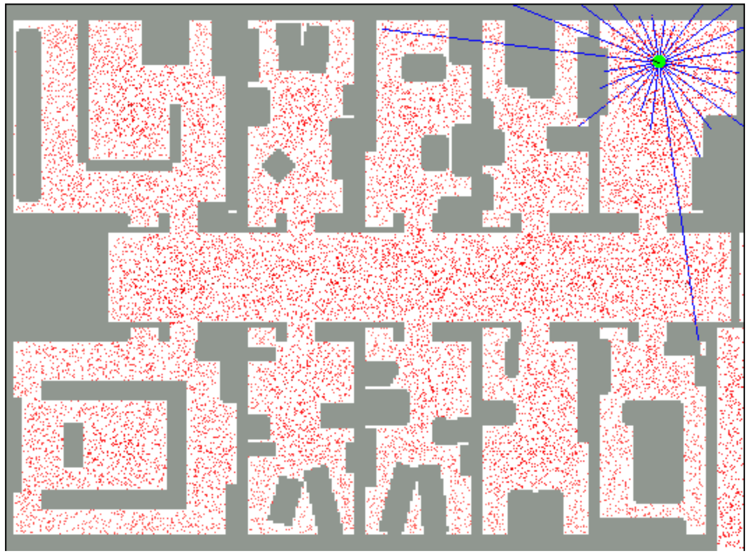

<0.01 <0.01 <0.01 <0.01 <0.01 <0.01
<0.01 <0.01 0.06 <0.01 <0.01 <0.01
<0.01 0.76 0.06 0.06 <0.01 <0.01
<0.01 <0.01 0.06 <0.01 <0.01 <0.01

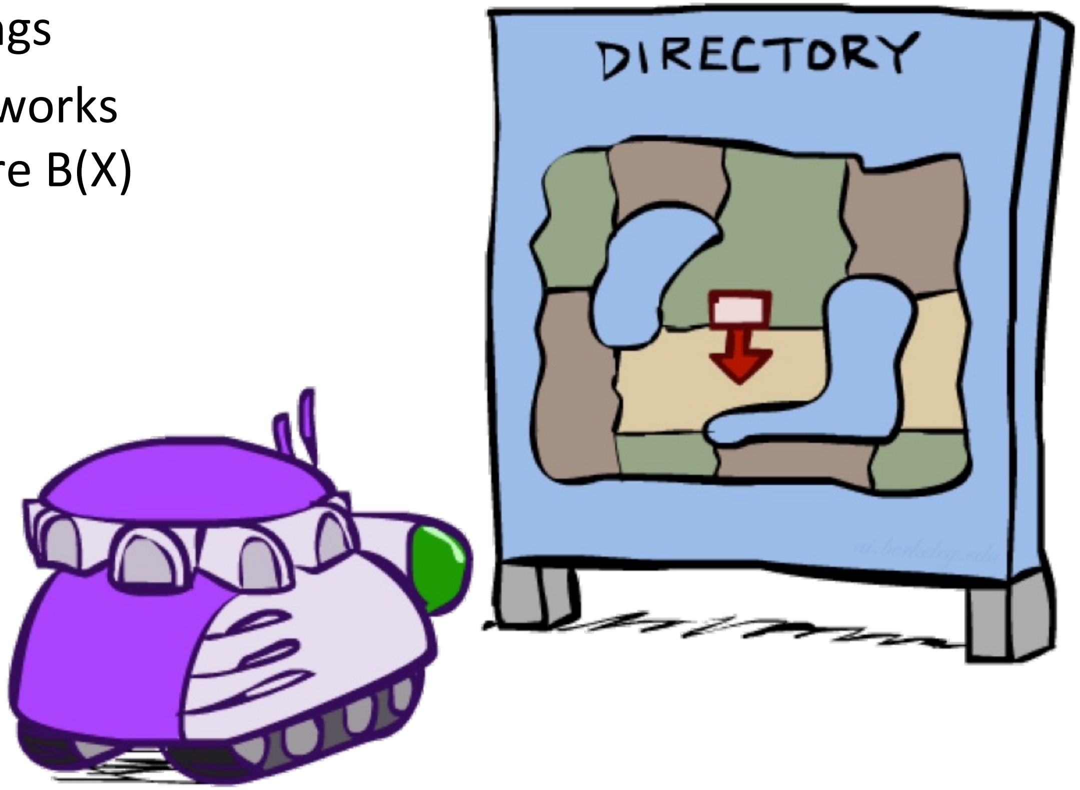




Robot Localization

- 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

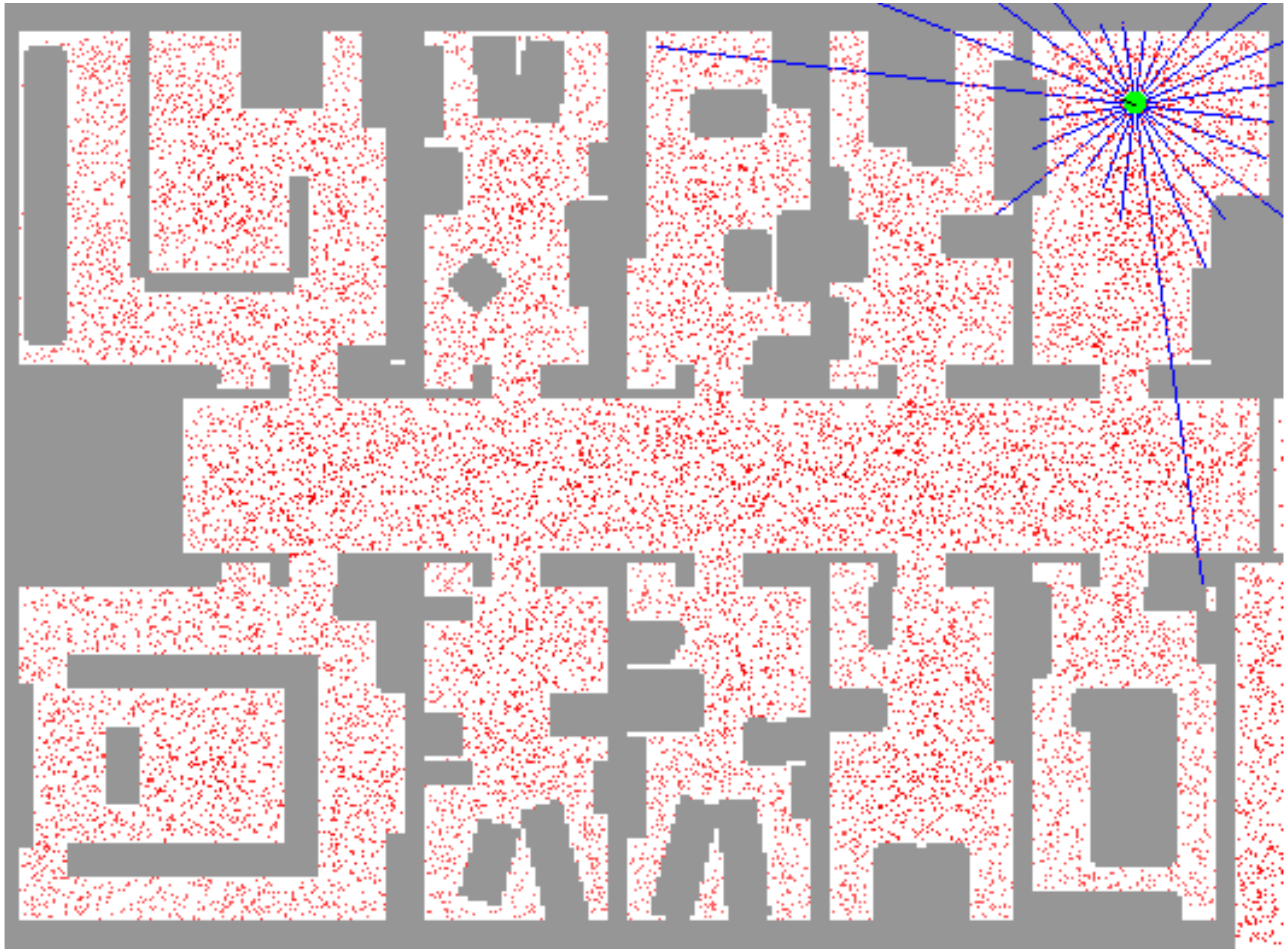




Particle Filter Localization (Sonar)

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

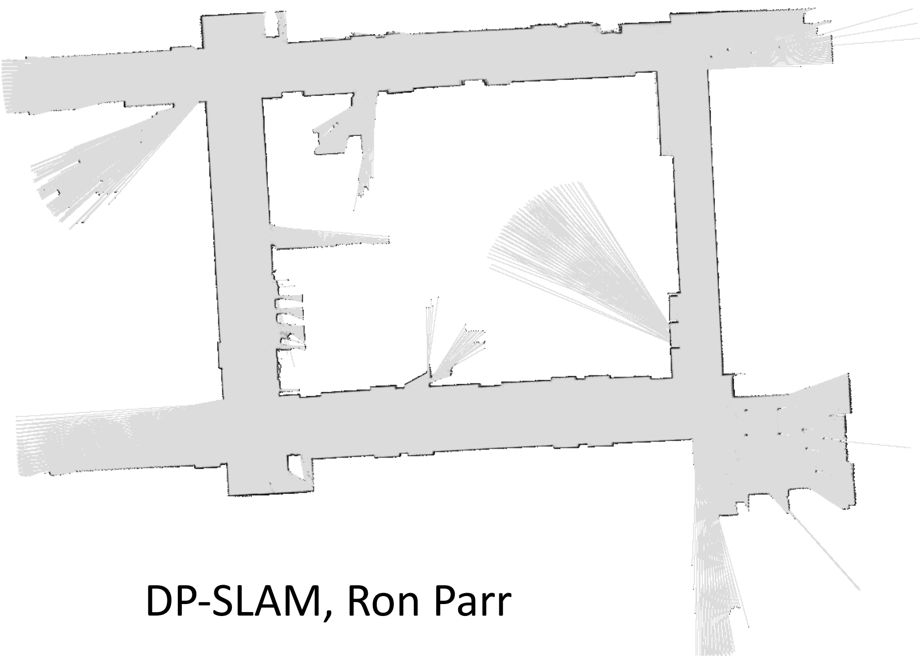
Particle Filter Localization (Laser)



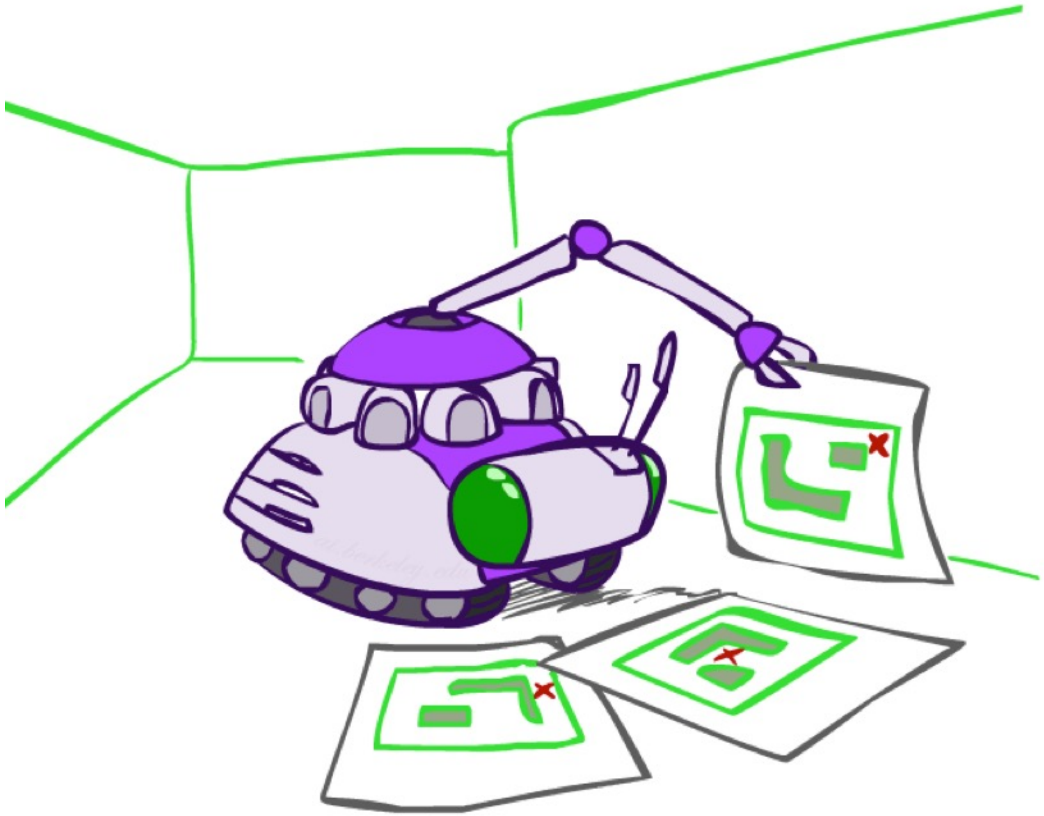
[Video: global-floor.gif]

Robot Mapping

- SLAM: Simultaneous Localization And Mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

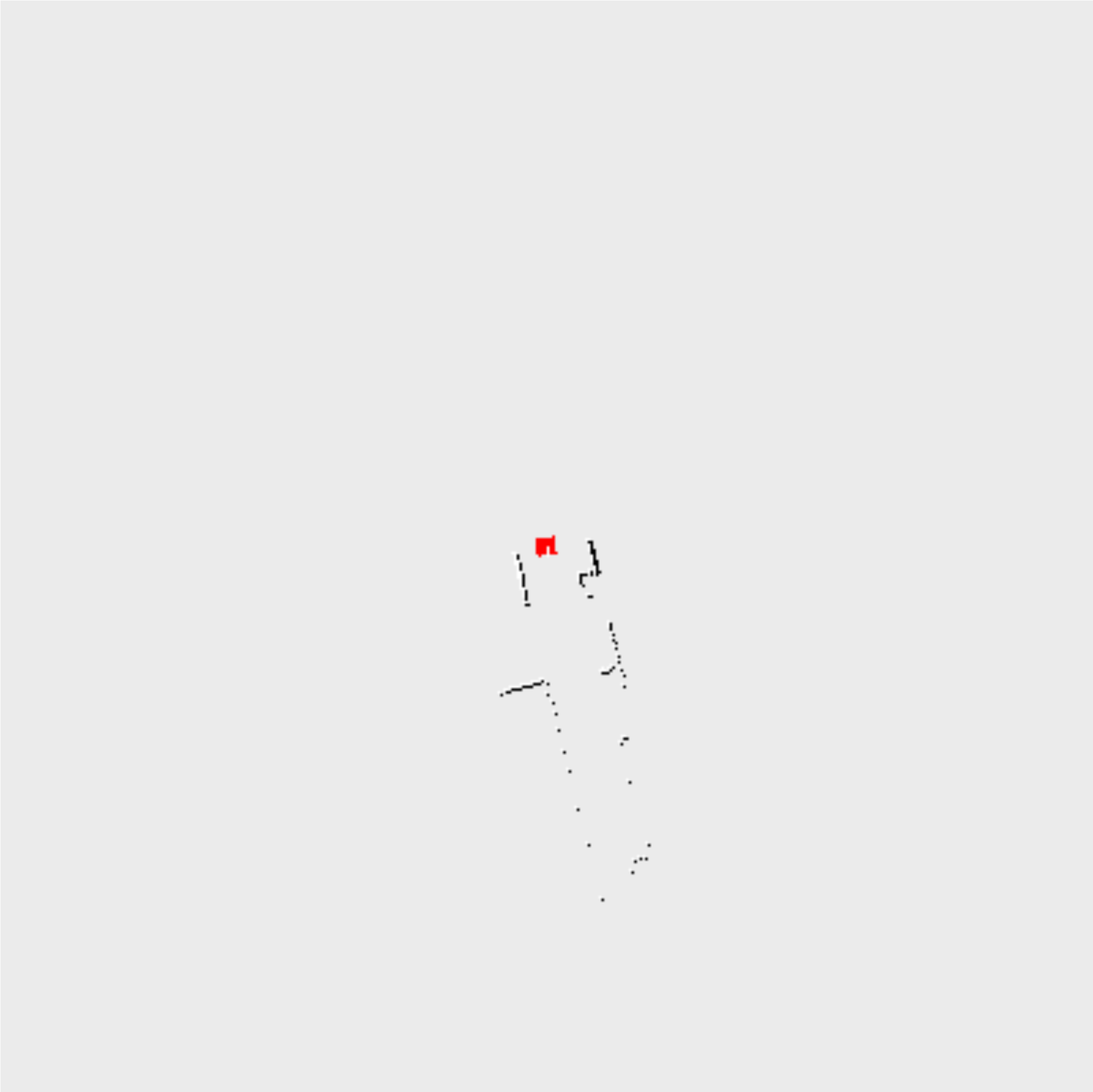


DP-SLAM, Ron Parr



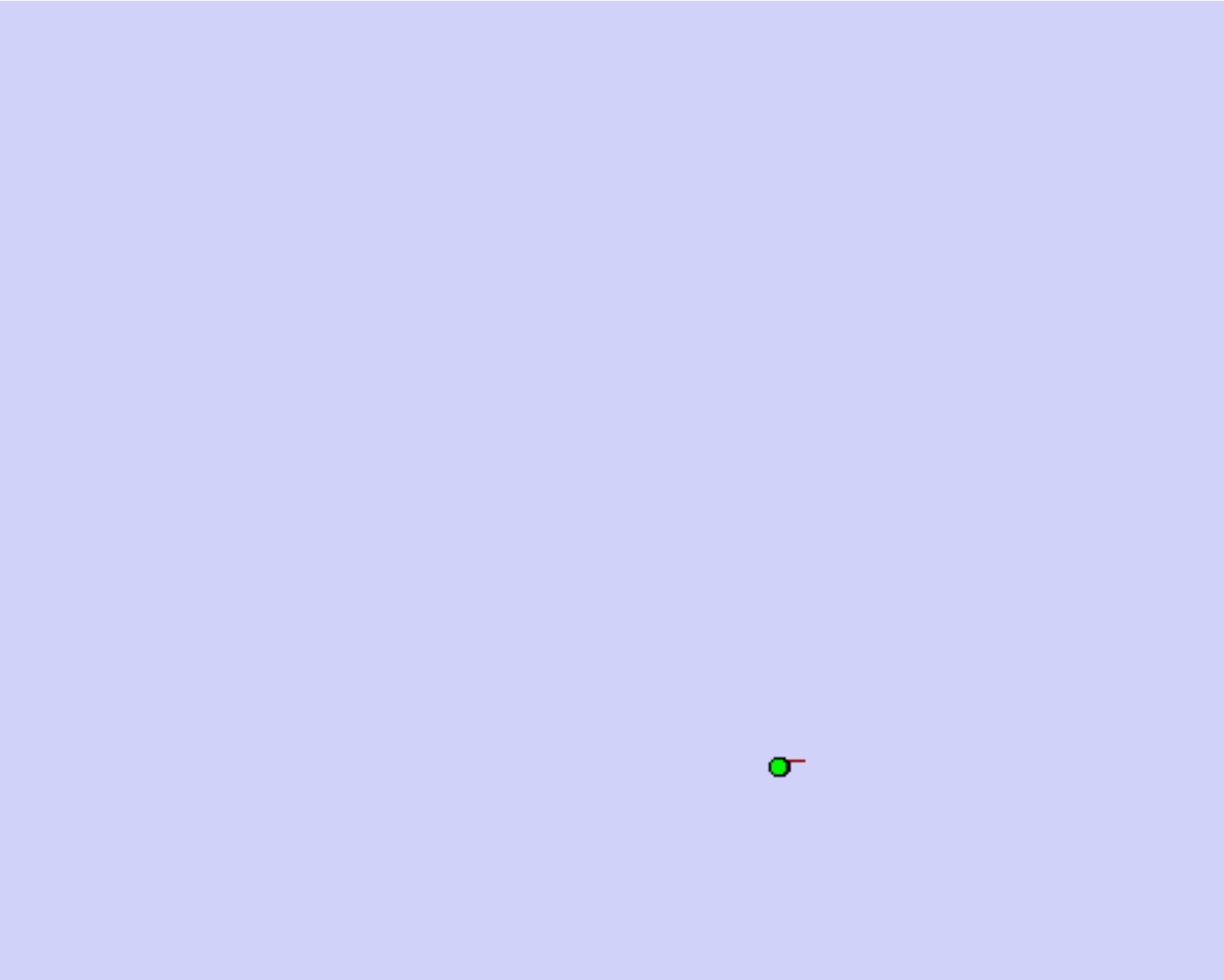
[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 1



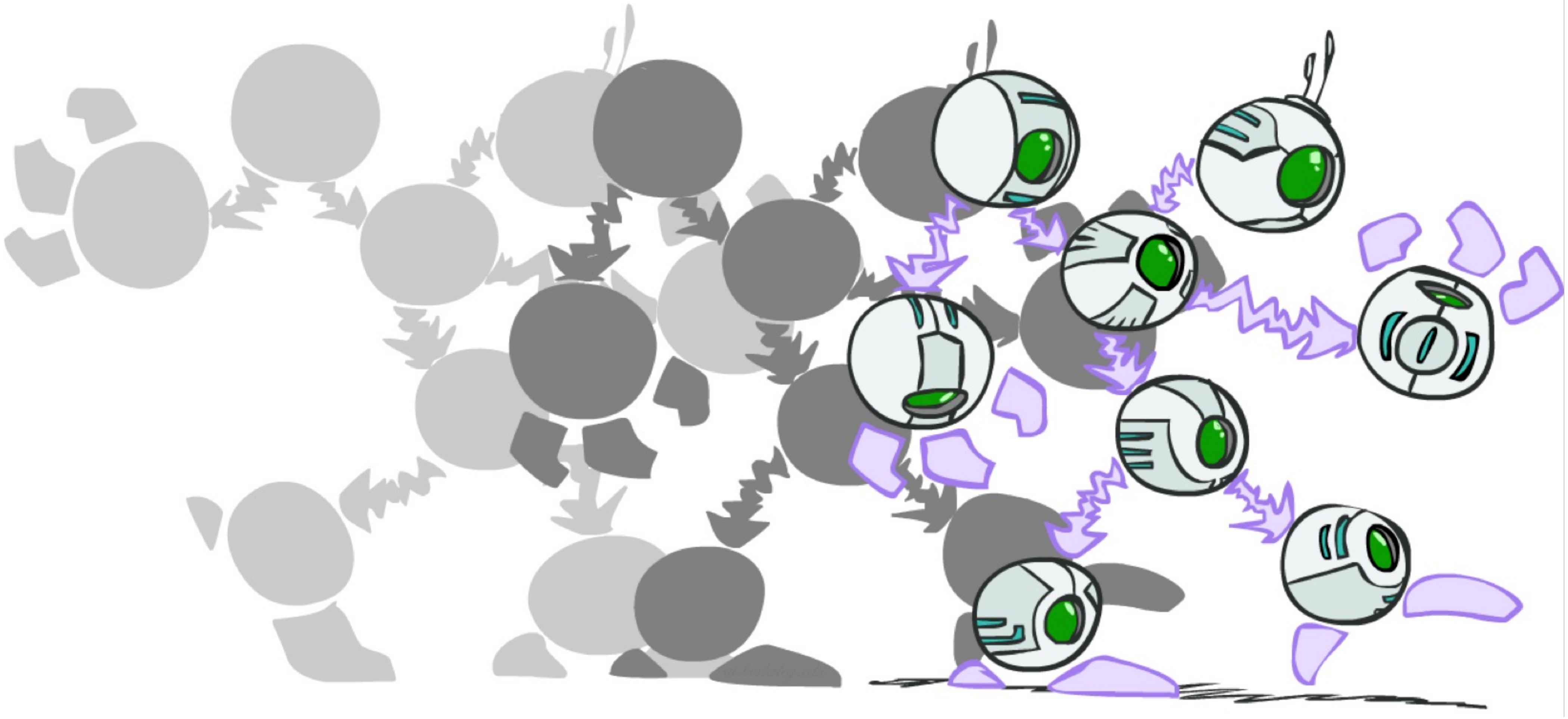
[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 2



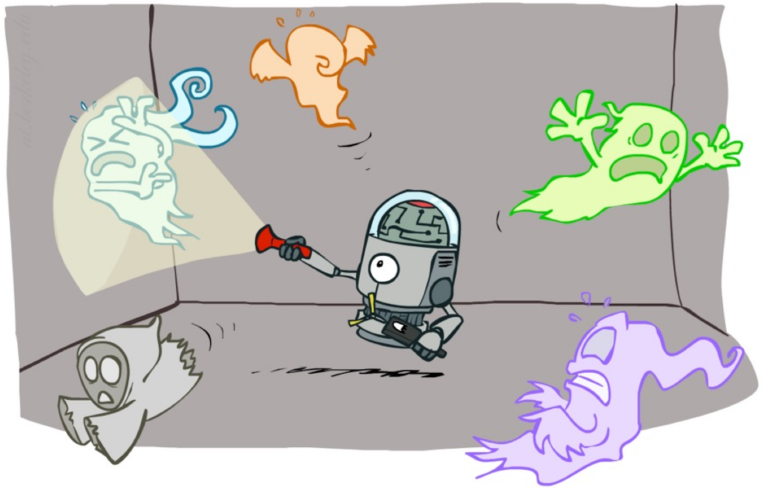
[Demo: PARTICLES-SLAM-fastslam.avi]

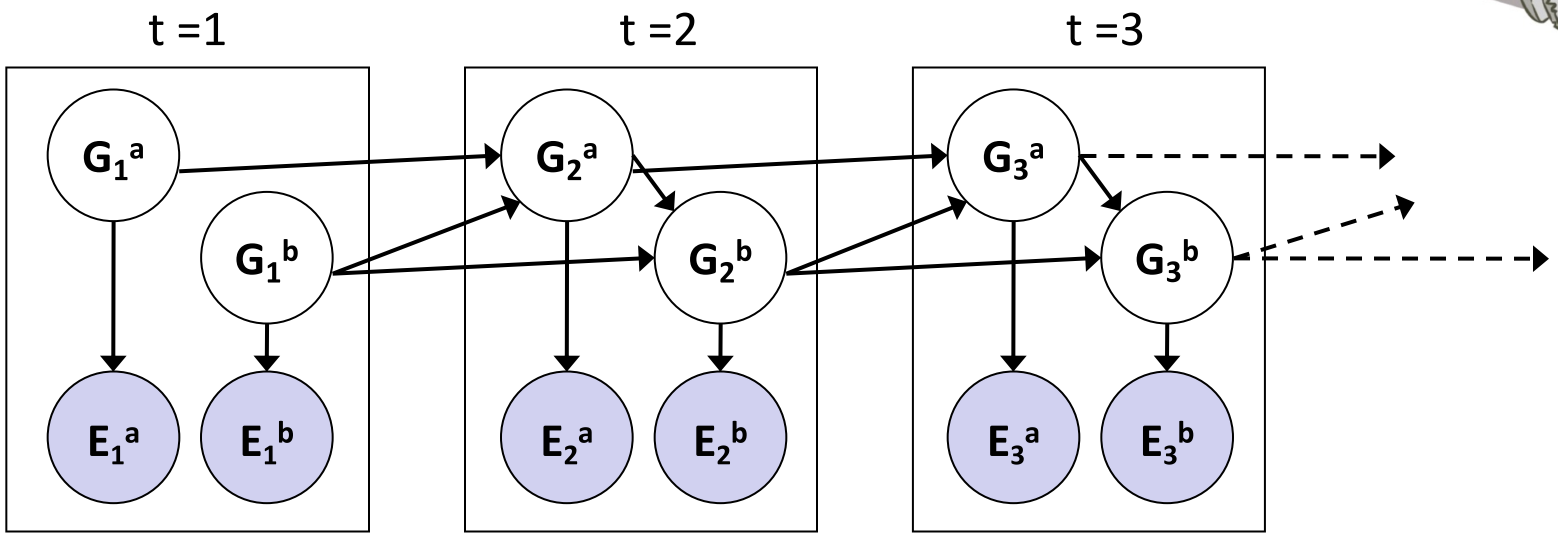
Dynamic Bayes Nets



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*

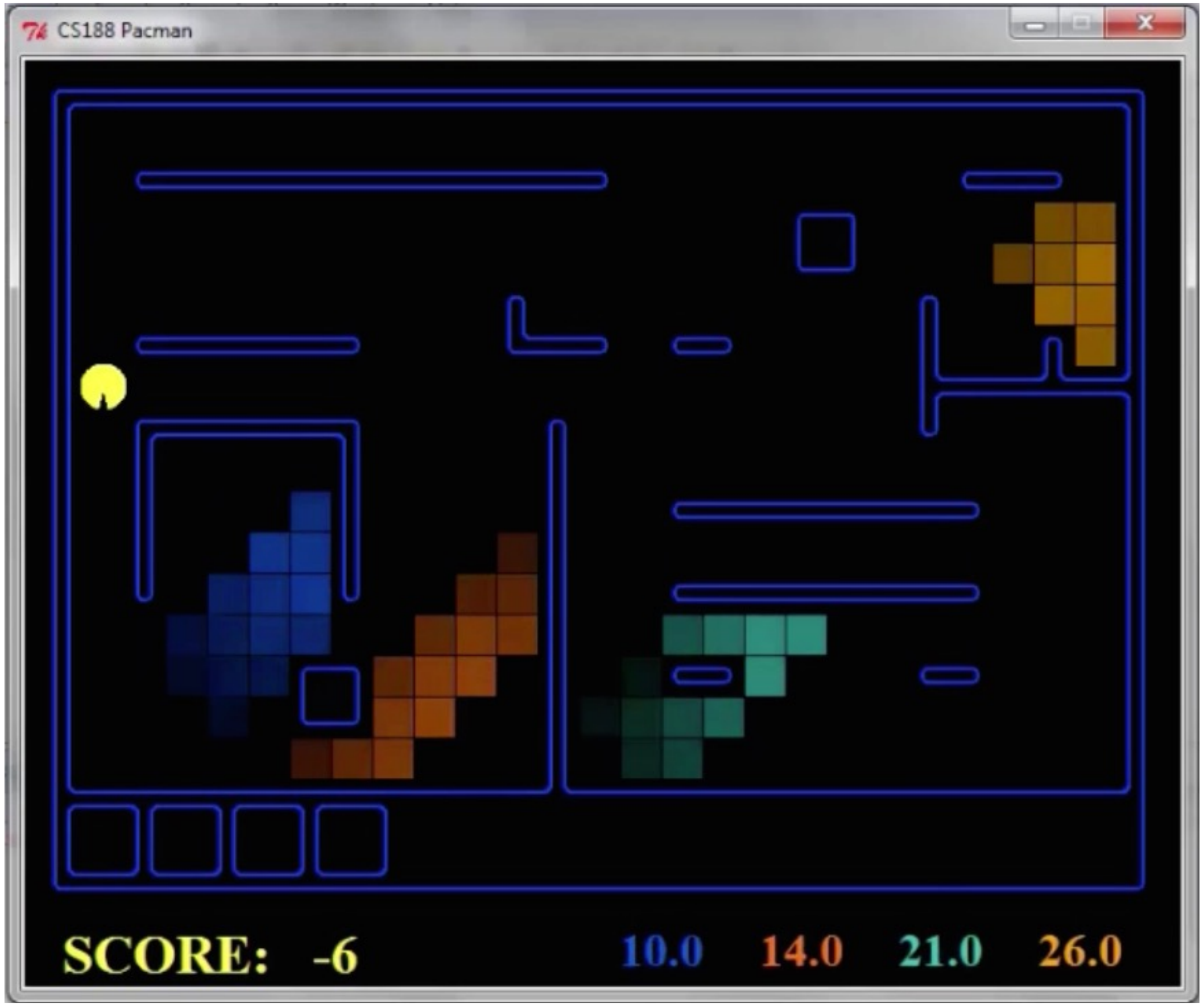




- Dynamic Bayes nets are a generalization of HMMs

[Demo: pacman sonar ghost DBN model (L15D6)]

Pacman – Sonar



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

Video of Demo Pacman Sonar Ghost DBN Model

Conclusion

- We're done with Part II: Uncertainty!

- We've seen methods for:
  - Representing uncertainty structure via **Bayes Nets** and multiple ways of doing inference
  - Incorporating decision-making with uncertainty via **Decision Nets**
  - Exploiting special structure of sequences / time via **Markov Models** and **Hidden Markov Models** and exact and approximate inference (**Particle Filtering**)

- Next up: Part III: Machine Learning!