Announcements

Assignments

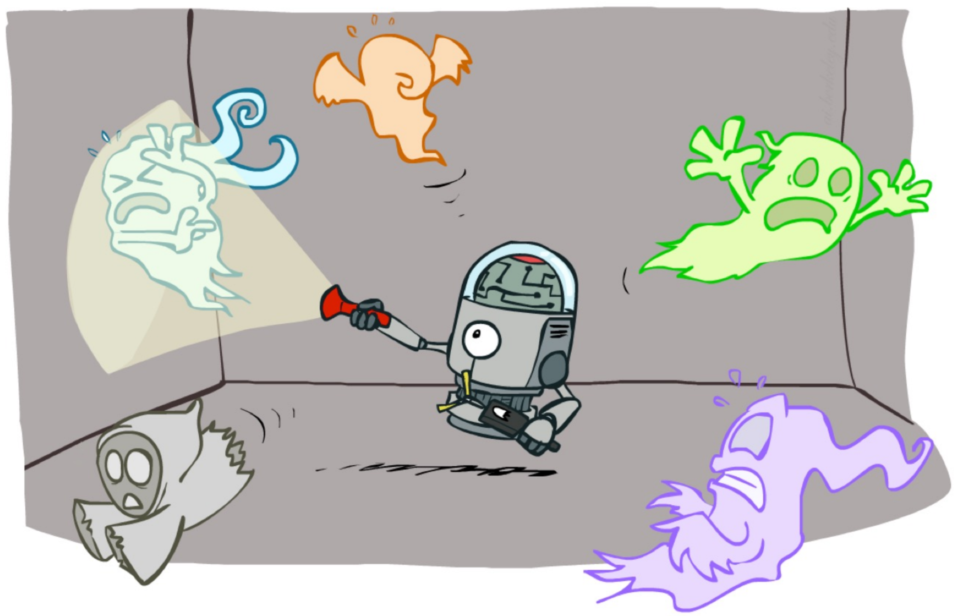
- HW10
  - Due 4/25, 10 pm
- P5
  - Due 4/27, 10 pm

Late days can be used on either/both assignments

Final Exam – 5/4 5:30pm (location TBA)

AI: Representation and Problem Solving

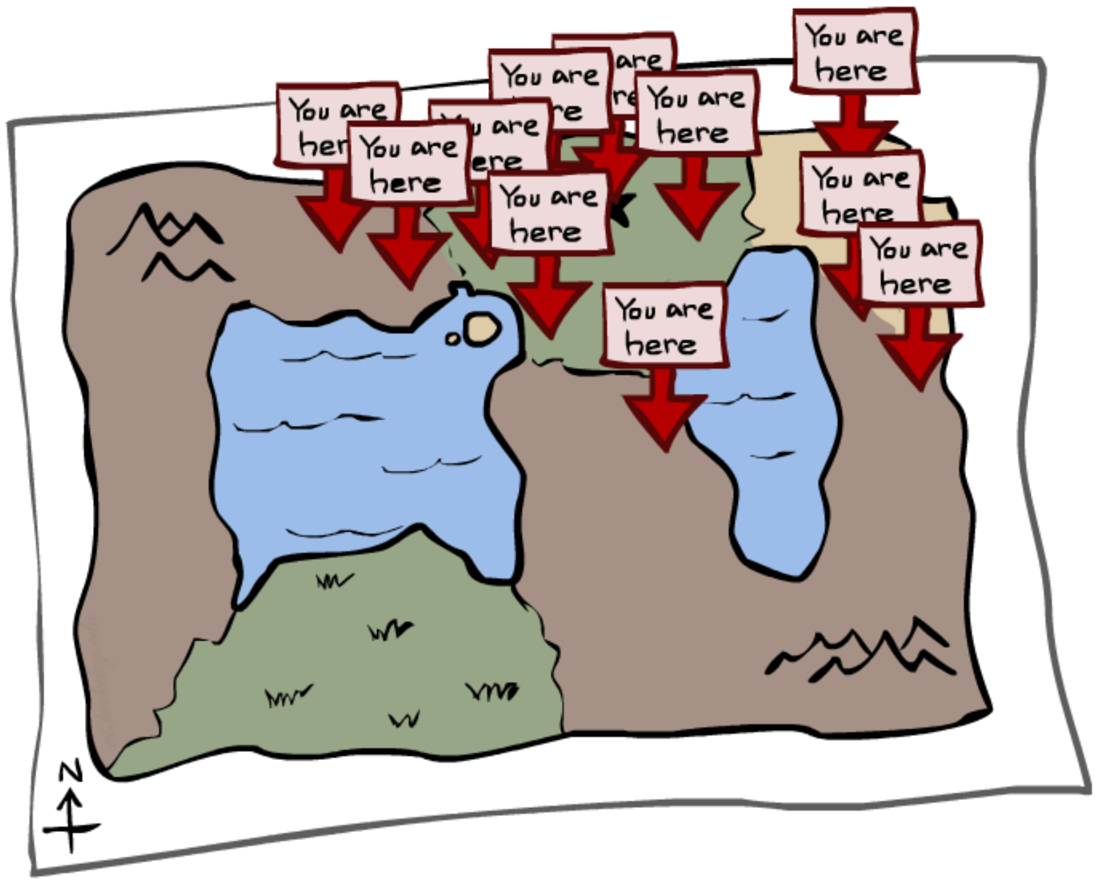
Particle Filtering



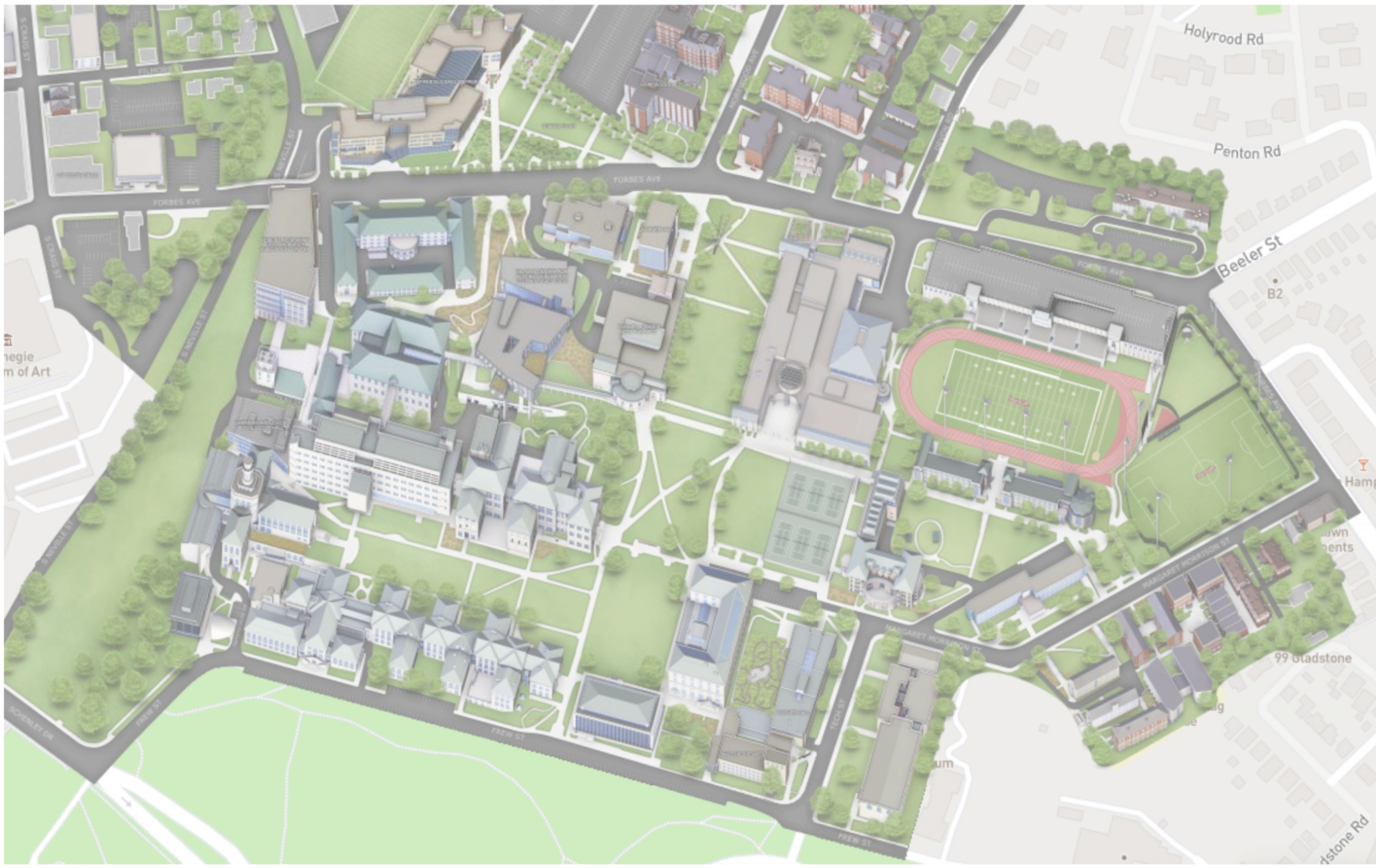
Instructor: Stephanie Rosenthal

Slide credits: CMU AI and http://ai.berkeley.edu

Particle Filtering



Particle Filtering



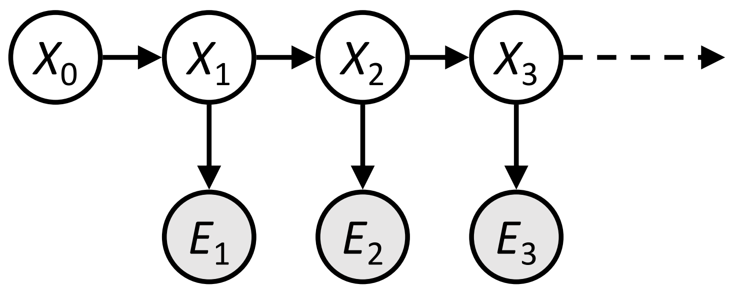
Belief States

When we're predicting the actual location we're in at each time step X_k

Really, what we're doing is maintaining a probability distribution over all possible states

This distribution is called a **belief state**, it represents the belief of where we are

We denote the belief state for X at time 3 is $\mathbf{b(X_3)=P(X_3)}$

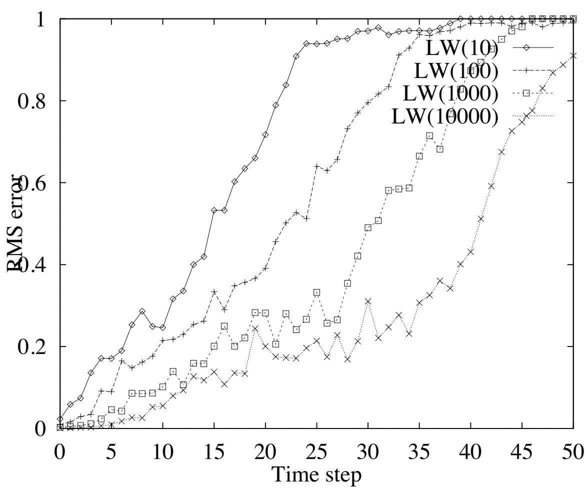


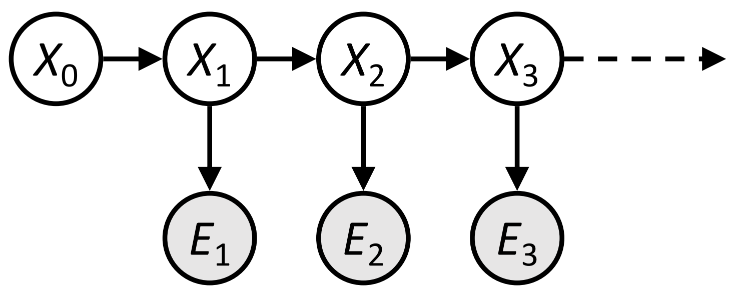
We need a new algorithm!

When |*X*| is more than 10^6 or so (e.g., 3 ghosts in a 10x20 world), exact inference to compute the belief state becomes infeasible

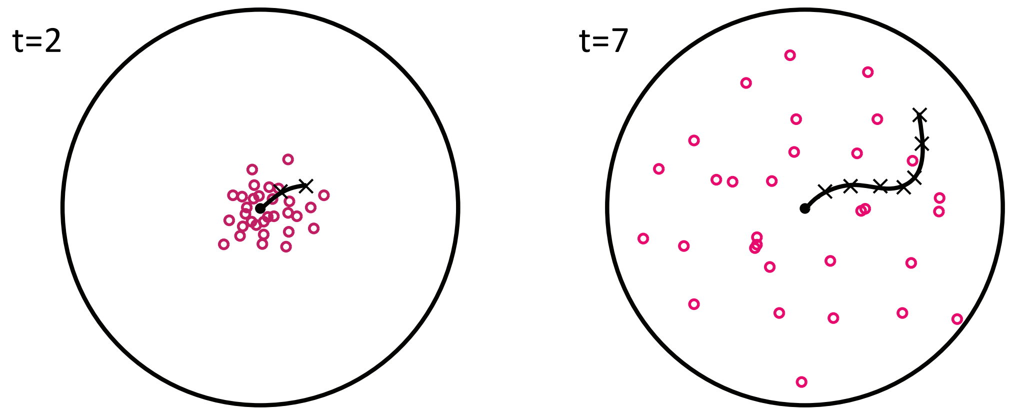
We could try to sample our Bayes net to compute b(X)

Likelihood weighting fails completely – number of samples needed grows ***exponentially*** with *T*





We need a new idea!



Idea: Sample in the first state, and then move those samples by sampling the transition function

The problem: sample state trajectories go off into low-probability regions, ignoring the evidence; too few "reasonable" samples

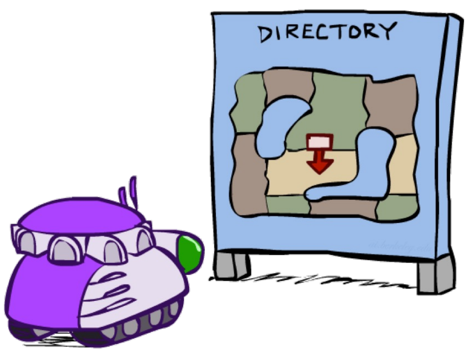
Solution: kill the bad ones, make more of the good ones. This way the population of samples stays in the high-probability region.

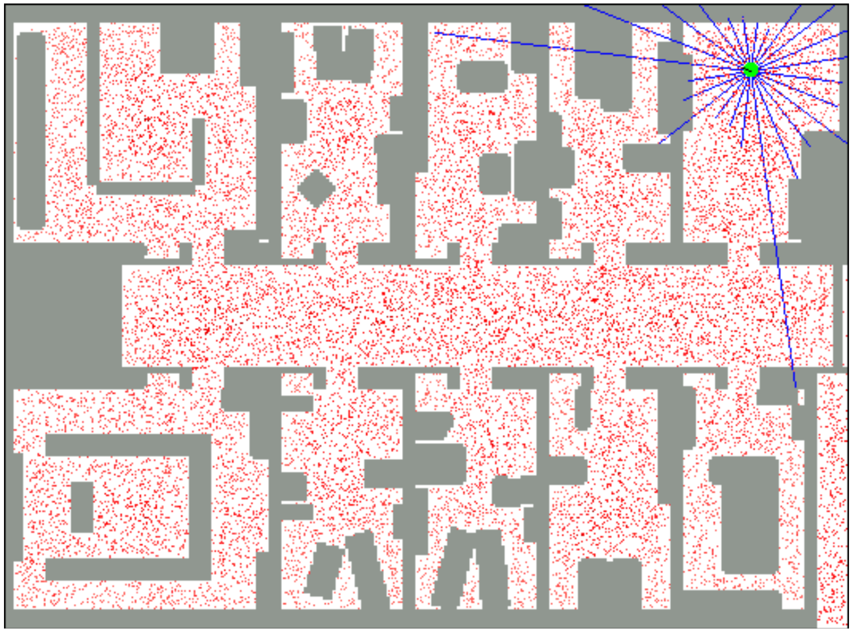
This is called resampling or survival of the fittest

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





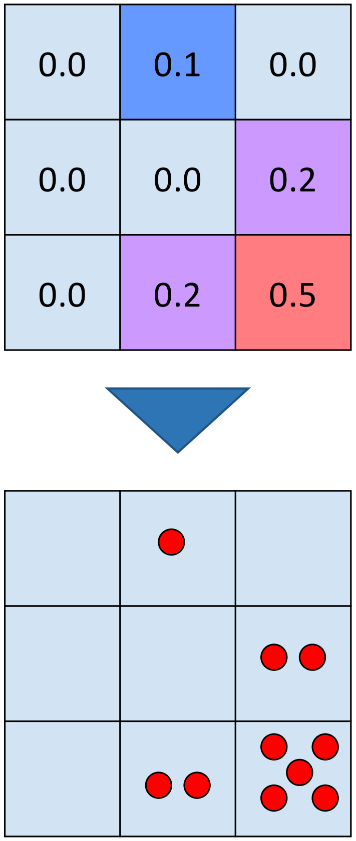
Particle Filter Localization (Sonar)

[Dieter Fox, et al.]

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

Particle Filtering

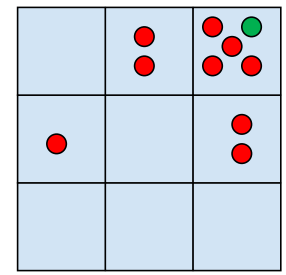
- Represent belief state by a set of samples
  - Samples are called ***particles***
  - Time per step is linear in the number of samples
  - But: number needed may be large
- This is how robot localization works in practice



Representation: Particles

Our representation of P(X) is now a list of N particles (samples)

- Generally, N << |X|
- Storing dictionary mapping from X to counts would defeat the point



P(x) approximated by number of particles with value x

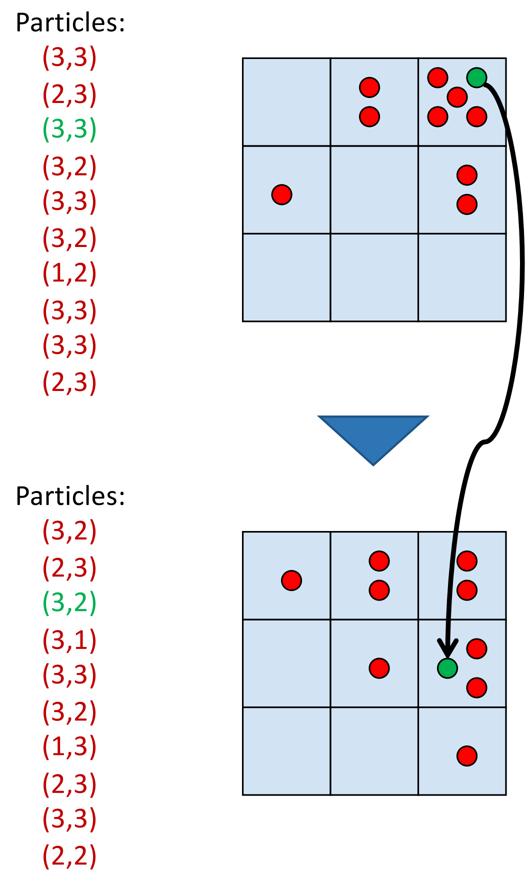
- So, many x may have P(x) = 0!
- More particles, more accuracy
- Usually we want a low-dimensional marginal
  - E.g., “Where is ghost 1?” rather than “Are ghosts 1,2,3 in {2,6}, [5,6], and [8,11]?”

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: Propagate forward ("Predict")

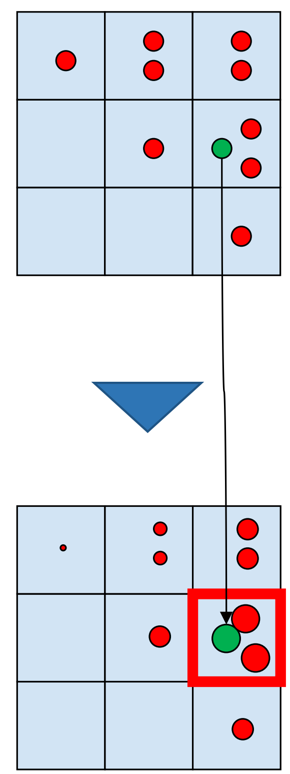
- A particle in state x_t is moved by sampling its next position directly from the transition model:
  - $x_{t+1} \sim P(X_{t+1} \mid x_t)$
  - Here, most samples move clockwise, but some move in another direction or stay in place
- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)



Particle Filtering: Observe/Weight ("Update" part 1)

- Slightly trickier:
  - Don't sample observation, fix it
  - Similar to likelihood weighting, weight samples based on the evidence
    - $W = P(e_t | x_t)$
  - Normalize the weights: particles that fit the data better get higher weights, others get lower weights

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
(3,3) w=.4
(2,2) w=.4

Particle Filtering: Resample ("Update" part 2)

Rather than tracking weighted samples, we ***resample***

We have an updated belief distribution based on the weighted particles

We sample N new particles from the weighted belief distributions

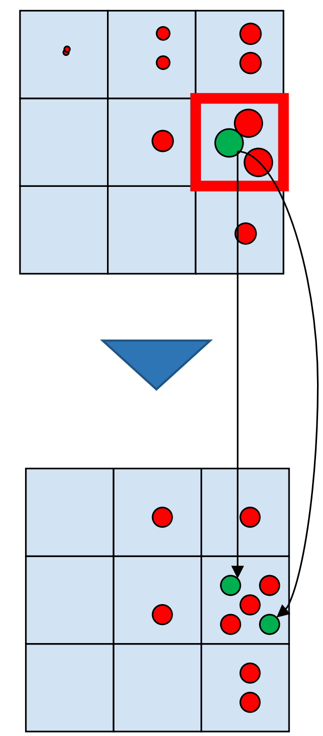
Now the update is complete for this time step, continue with the next one

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
(3,3) w=.4
(2,2) w=.4

(New) Particles:

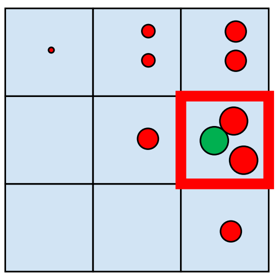
(3,2)
(2,2)
(3,2)
(2,3)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(3,2)



Weighting and Resampling

How to compute a belief distribution given weighted particles

Weight

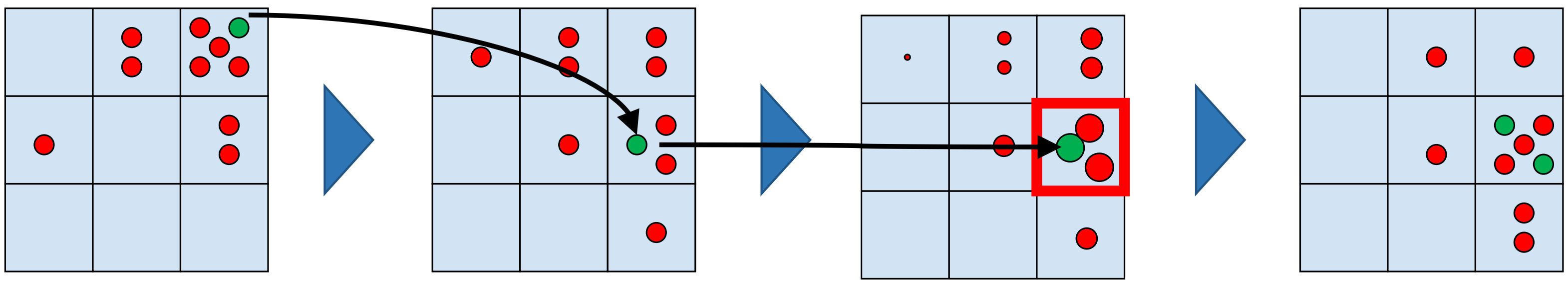


Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,3) w=.4
(2,2) w=.4

Summary: Particle Filtering

Particles: track samples of states rather than an explicit distribution

Propagate forward based on transition function | Weight based on observation function | Resample using weighted particles

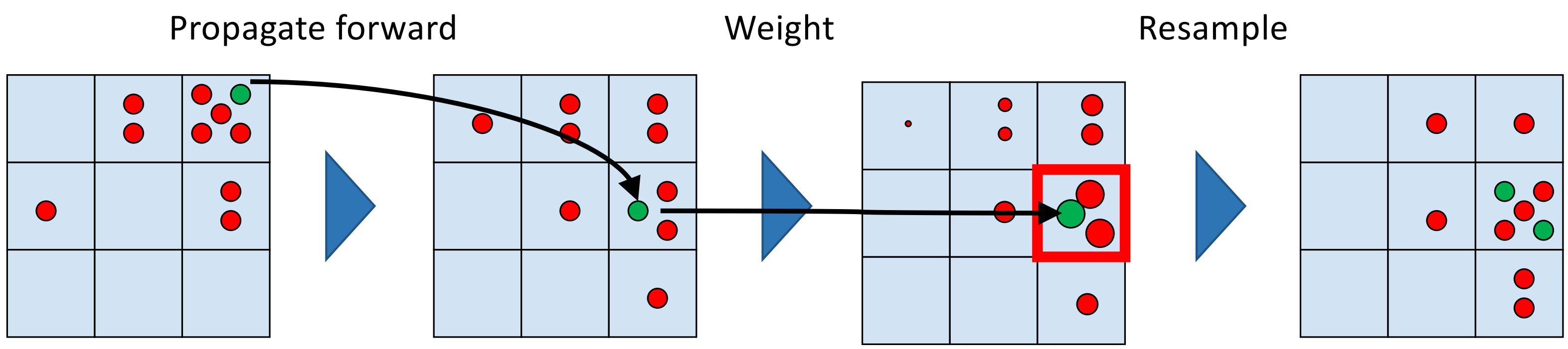


Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,3) w=.4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

Consistency: see proof in AIMA Ch. 14

Poll 1

If we only have one particle which of these steps are unnecessary?

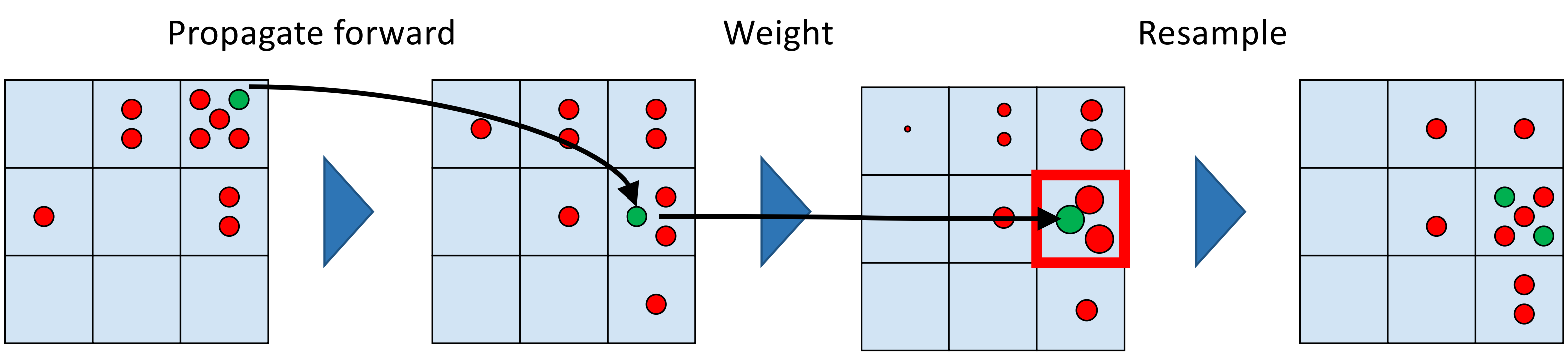


Select all that are unnecessary.

A. Propagate forward
B. Weight
C. Resample
D. None of the above

Poll 1

If we only have one particle which of these steps are unnecessary?

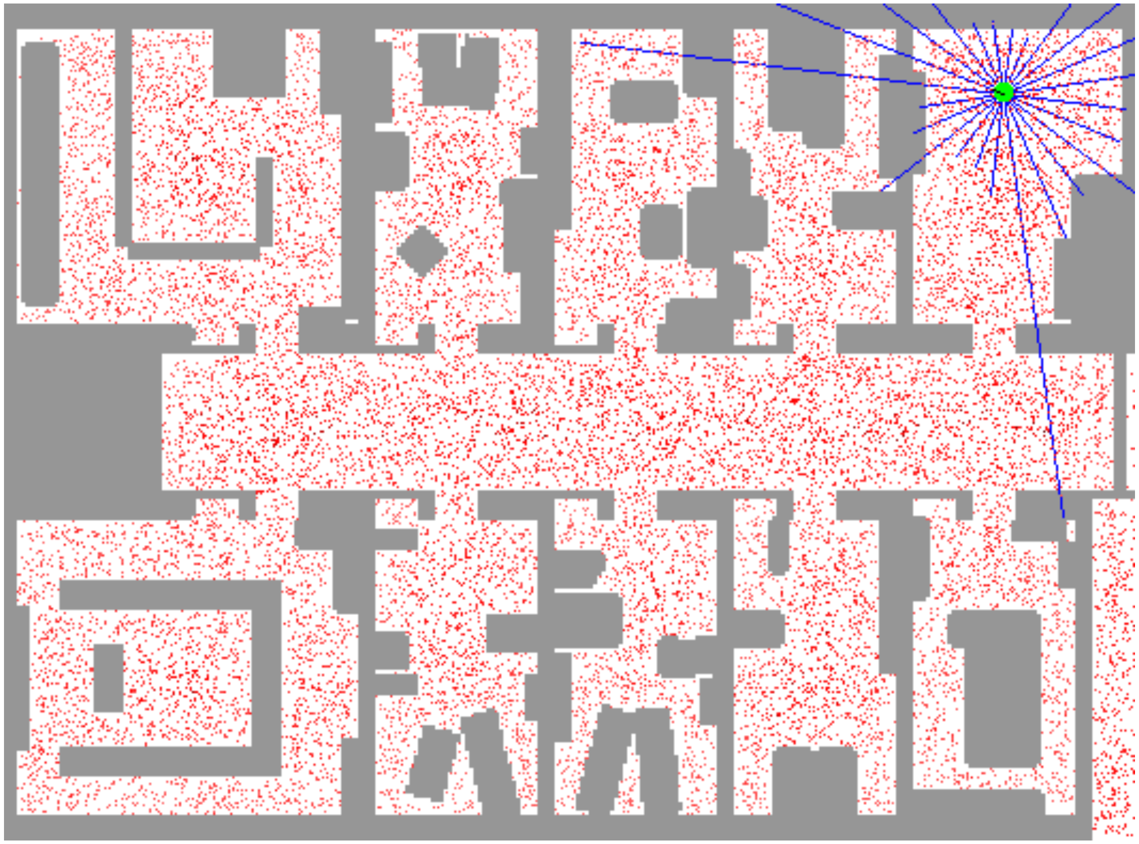


Select all that are unnecessary.

A. Propagate forward
B. Weight
C. Resample
D. None of the above

Unless the weight is zero, in which case, you'll want to resample from the beginning ☹

Particle Filter Localization (Laser)



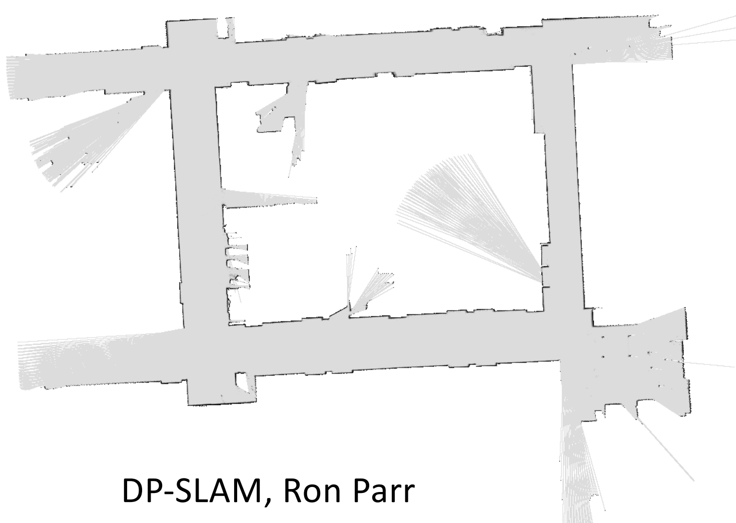
[Dieter Fox, et al.]

[Video: global-floor.gif]

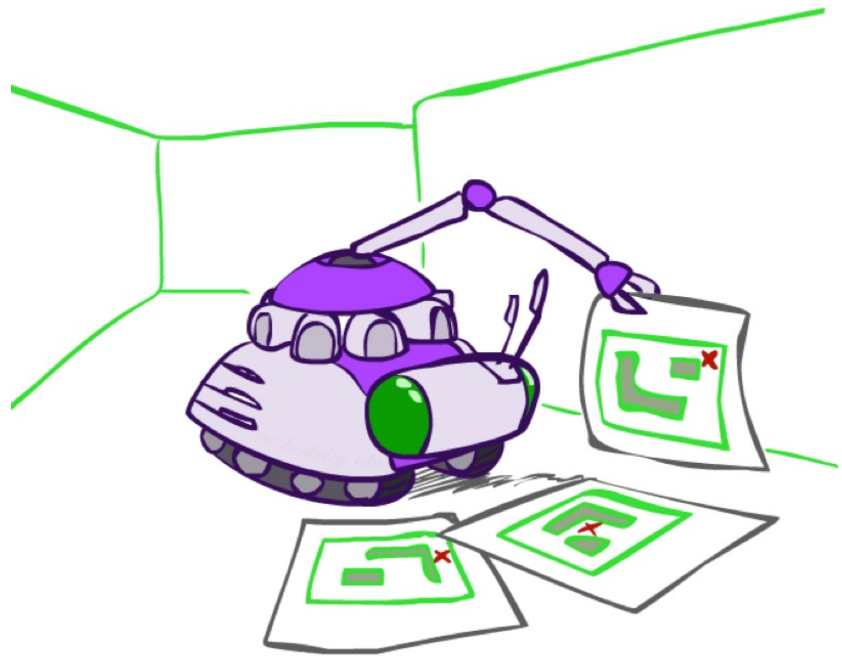
Robot Mapping

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

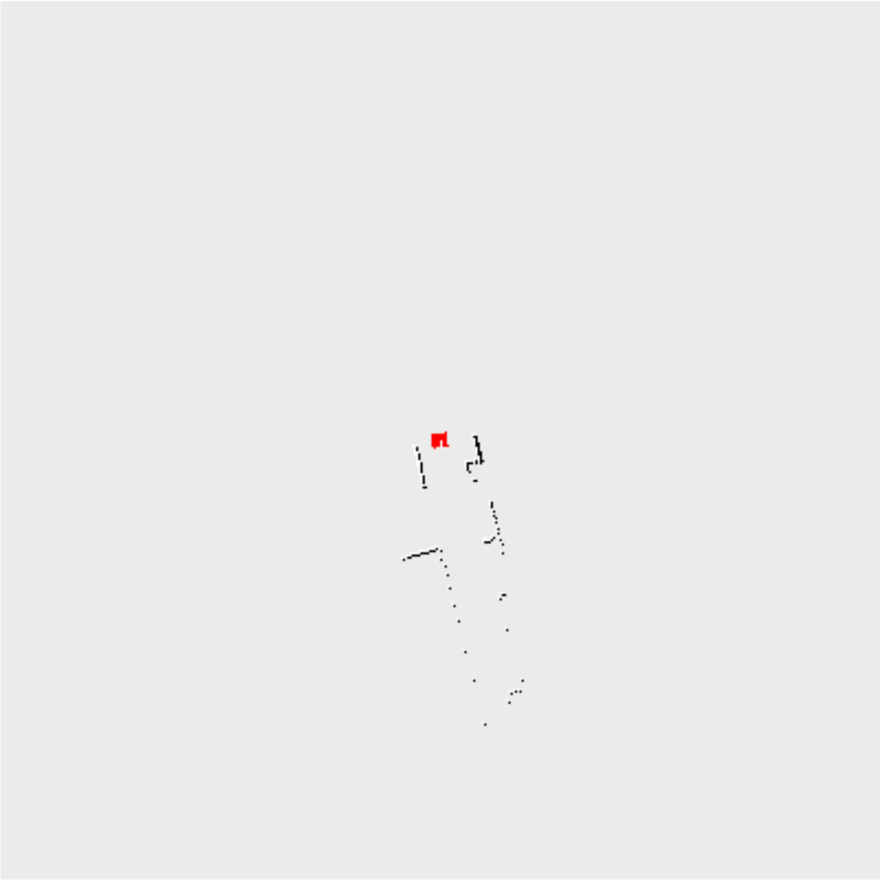


DP-SLAM, Ron Parr



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

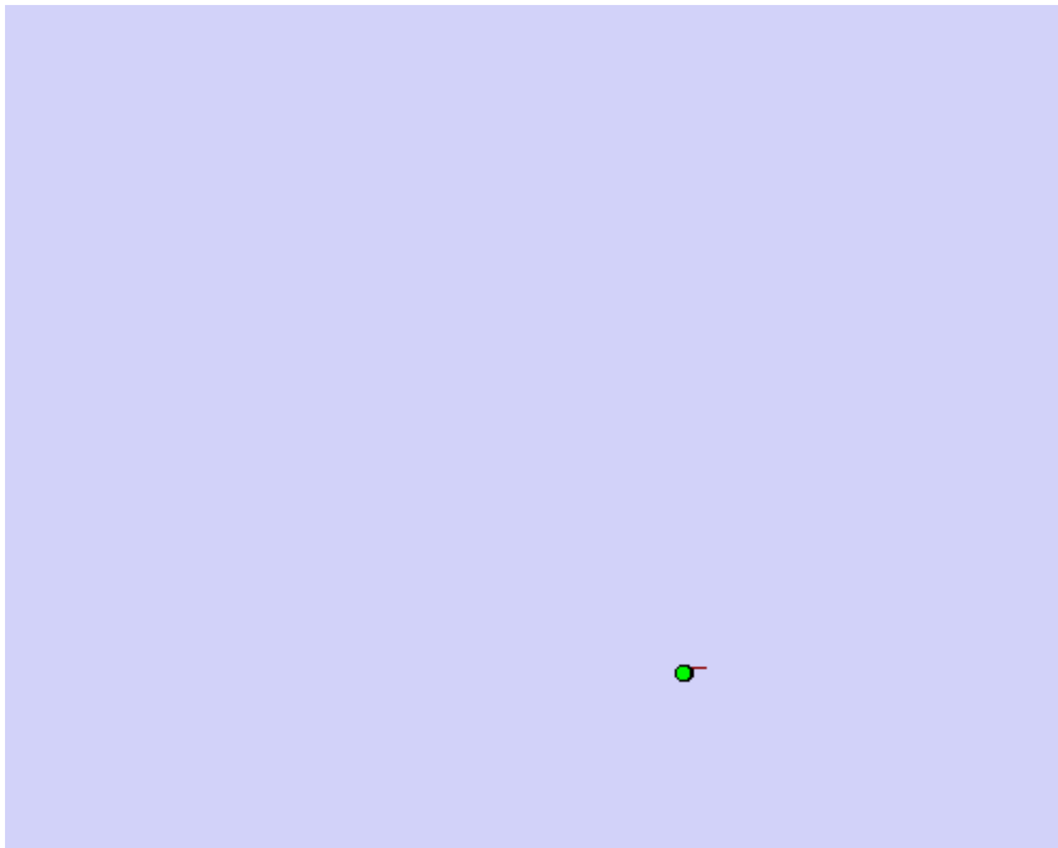
Particle Filter SLAM – Video 1



[Sebastian Thrun, et al.]

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

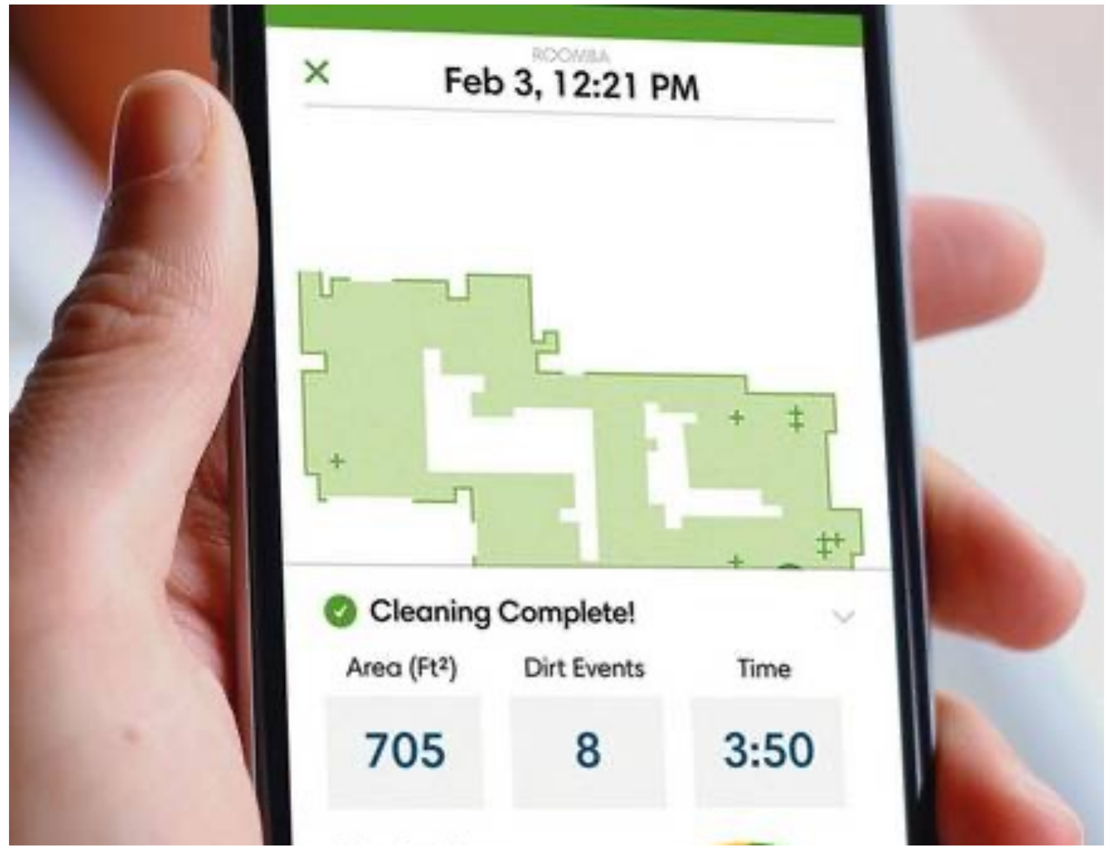
Particle Filter SLAM – Video 2



[Dirk Haehnel, et al.]

[Demo: PARTICLES-SLAM-fastslam.avi]

SLAM



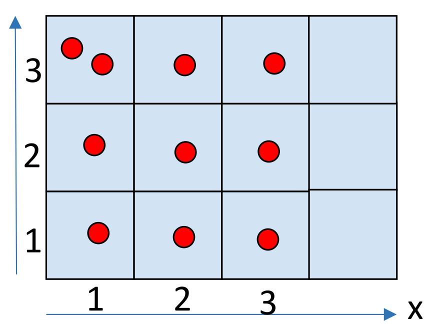
https://www.irobot.com/

Use Python random.random() or Google to sample

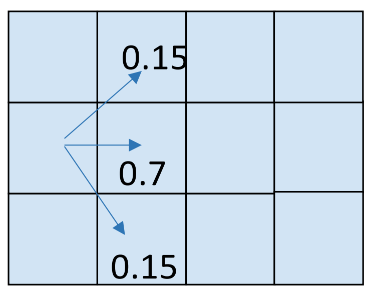
In Class Activity

Given the following starting particles, transition model, and e_1 and e_2 observed at time 1 and time 2, what is the approximate belief state at time 2?

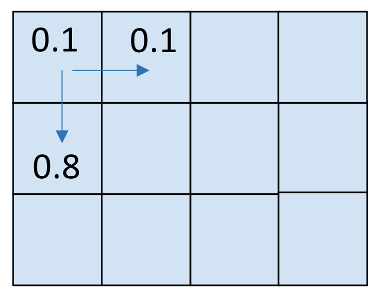
T=0



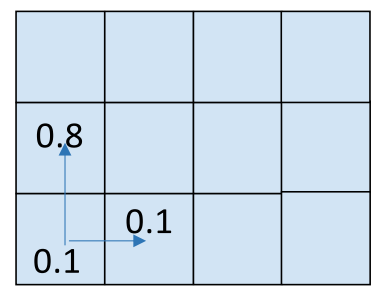
$P(M_{t+1}|M_t$ in middle row)



$P(M_{t+1}|M_t$ in top row)



$P(M_{t+1}|M_t$ in bottom row)



$P(e_1|m_1)$

.3	.5		
.5	.5		
.2	.5		

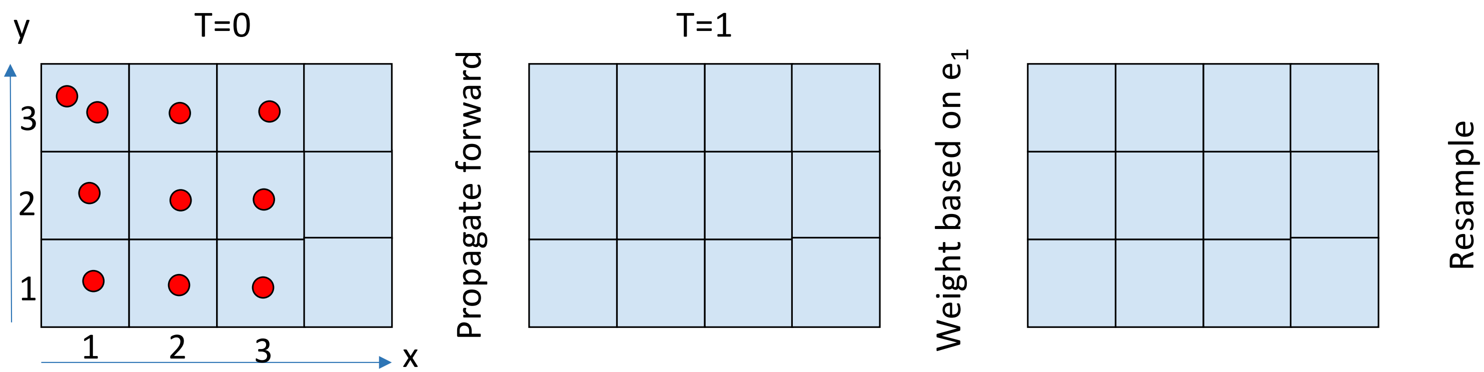
$P(e_2|m_2)$

	.05	.4	
	.3	.5	
	.05	.2	

Use random.random() or Google to sample

In Class Activity

Given the following starting particles, transition model, and e_1 observed at time 1, what is the approximate belief state at time 1?



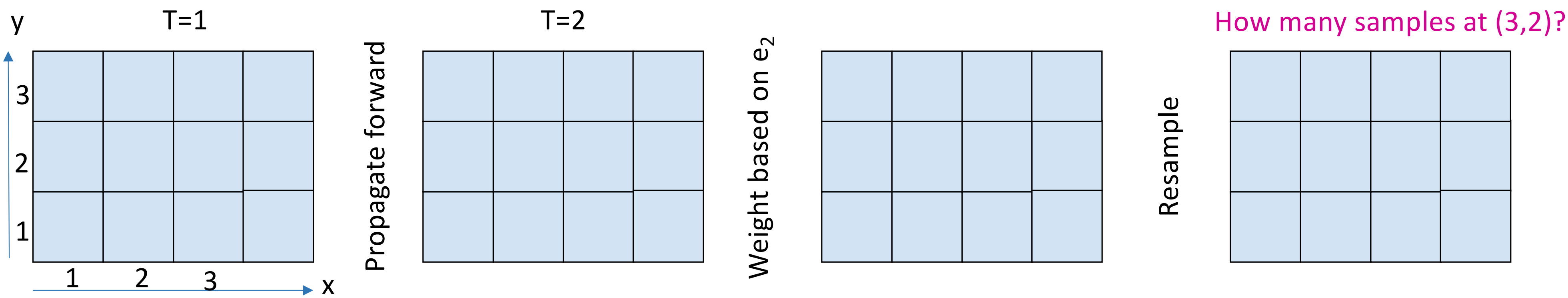
$P(e_1|m_1)$

.3	.5		
.5	.5		
.2	.5		

Use random.random() or Google to sample

In Class Activity – Poll 2

Given the particles at T=1, transition model, and e_2 observed at time 2, what is the approximate belief state at time 2?



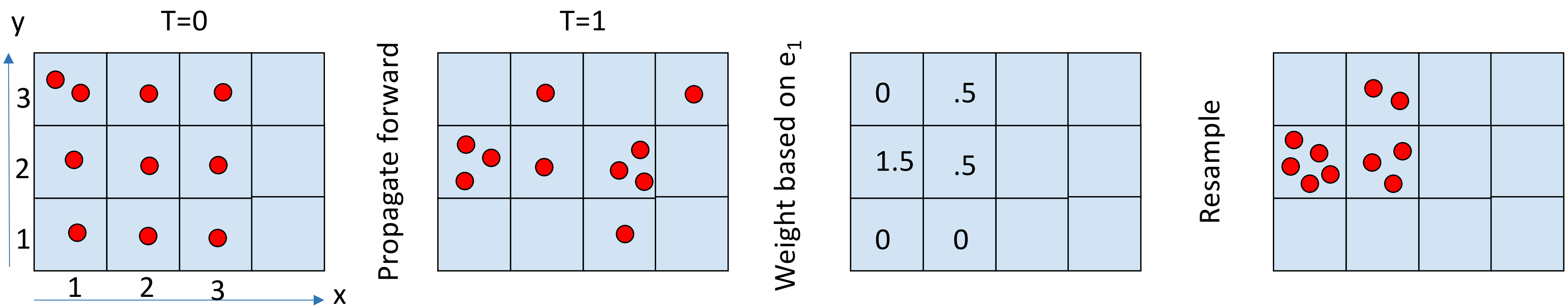
$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	

Use random.random() to sample

In Class Activity – Example Solution

Given the following starting particles, transition model, and e_1 observed at time 1, what is the approximate belief state at time 1?



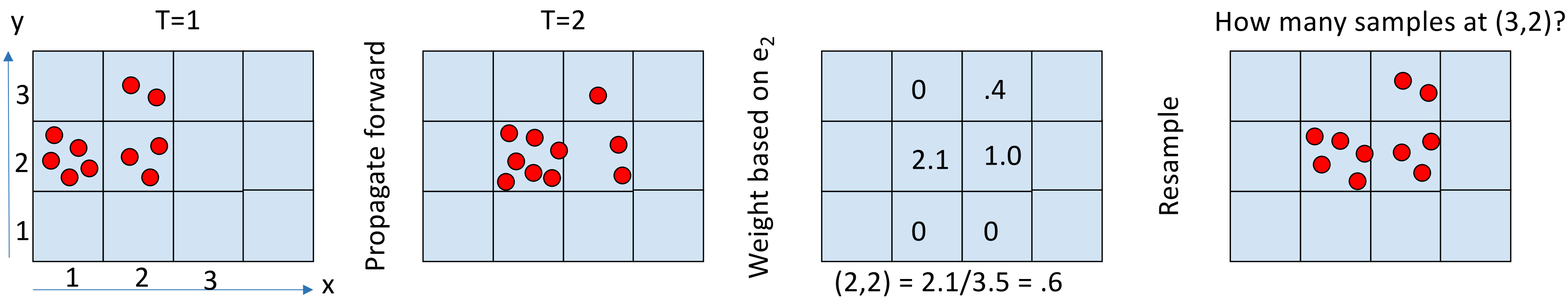
P(e_1|m)

.3	.5		
.5	.5		
.2	.5		

Use random.random() to sample

In Class Activity – Example Solution

Given the T=1 particles, transition model, and e_2 observed at time 2, what is the approximate belief state at time 2?



(2,2) = 2.1/3.5 = .6

(3,2) = 1.0/3.5 = .29

(3,3) = .4/3.5 = .11

$P(e_2|x_2)$

	.05	.4	
	.3	.5	
	.05	.2	