EdX and its Members use cookies and other tracking technologies for performance, analytics, and marketing purposes. By using this website, you accept this use. Learn more about these technologies in the <u>Privacy Policy</u>.





<u>Course</u> > <u>Week 10</u> > <u>Project 4: Ghostbusters</u> > p4_tracking_q4_approximate_inference_observation

p4_tracking_q4_approximate_inference_observation Question 4 (3 points): Approximate Inference Observation

Approximate inference is very trendy among ghost hunters this season. Next, you will implement a particle filtering algorithm for tracking a single ghost.

Implement the functions initializeUniformly, getBeliefDistribution, and observe for the ParticleFilter class in inference.py. A correct implementation should also handle two special cases. (1) When all your particles receive zero weight based on the evidence, you should resample all particles from the prior to recover. (2) When a ghost is eaten, you should update all particles to place that ghost in its prison cell, as described in the comments of observe. When complete, you should be able to track ghosts nearly as effectively as with exact inference.

To run the autograder for this question and visualize the output:

python autograder.py -q q4

Hints:

- A particle (sample) is a ghost position in this inference problem.
- The belief cloud generated by a particle filter will look noisy compared to the one for exact inference.
- util.sample or util.nSample will help you obtain samples from a distribution. If you use util.sample and your implementation is timing out, try using util.nSample.

© All Rights Reserved

1 of 1 12/23/18, 9:20 AM