**Homework 8 Particle Filtering and Bayes Nets**

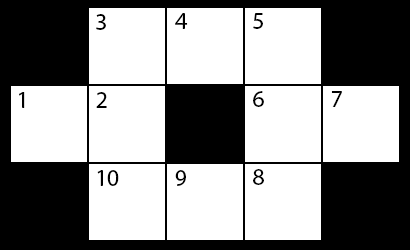
**Name *Ngoc Ha***

**NOTE: Only do the first three problems!!**

**Question 1: Particle Filtering**

8/8 points

In this question, we will use a particle filter to track the state of a robot that is lost in the small map below:



The robot's state is represented by an integer 1 ≤ Xt ≤ 10 corresponding to its location in the map at time t. We will approximate our belief over this state with N = 8 particles.

You have no control over the robot's actions. At each timestep, the robot either stays in place, or moves to any one of its neighboring locations, all with equal probability. For example, if the robot starts in state Xt = 7, it will move to state Xt+1 = 6 with probability ½ or Xt+1 = 7 with probability ½. Similarly, if the robot starts in state Xt = 2, the next state Xt+1 can be any element of {1, 2, 3, 10}, and each occurs with probability ¼.

At each time step, a sensor on the robot gives a reading Et ∈ {H, C, T, D} corresponding to the *type* of state the robot is in. The possible types are:

* **Hallway (H)** for states bordered by two parallel walls (4,9).
* **Corner (C)** for states bordered by two orthogonal walls (3,5,8,10).
* **Tee (T)** for states bordered by one wall (2,6).
* **Dead End (D)** for states bordered by three walls (1,7).

The sensor is not very reliable: it reports the correct type with probability ½, but gives erroneous readings the rest of the time, with probability 1/6 for each of the three other possible readings.

**Part 1: Sensor Model**

Fill in the sensor model below:

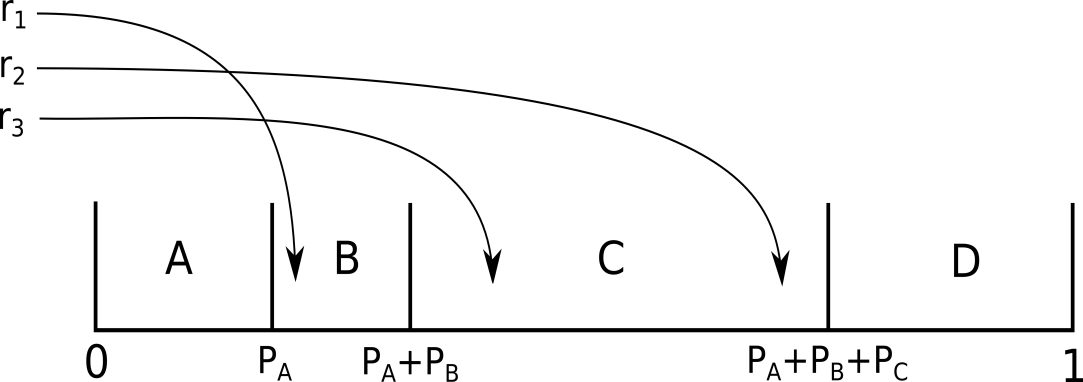
|  |  |  |
| --- | --- | --- |
| **Sensor Reading** | **State Type** | **P(Sensor | State Type)** |
| H | H |  |
| C | H |  |
| T | H |  |
| D | H |  |
| H | C |  |
| C | C |  |
| T | C |  |
| D | C |  |
| H | T |  |
| C | T |  |
| T | T |  |
| D | T |  |
| H | D |  |
| C | D |  |
| T | D |  |
| D | D |  |

**problem**

8/8 points

**Sampling Review**

Suppose that we want to sample from a set of 4 events, {A, B, C, D}, which occur with corresponding probabilities PA, PB, PC, PD. First, we form the set of cumulative weights, given by {0, PA, PA + PB, PA + PB + PC, 1}. These weights partition the [0, 1) interval into bins, as shown below. We then draw a number r uniformly at random from [0, 1) and pick A, B, C or D based on which bin r lands in. The process is illustrated in the diagram below. If r1, uniformly chosen from [0, 1), lands in the interval [PA, PA + PB], then the resulting sample would be B. Similarly, if r2 lands in [PA + PB, PA + PB + PC], the sample would be C, and r3 landing in [PA + PB, PA + PB + PC] would also be C.



**Part 2: Belief State for t=0**

Now we will sample the starting positions for our particles. For each particle pi, we have generated a random number ri sampled uniformly from [0, 1). Your job is to use these numbers to sample a starting location for each particle. As a reminder, locations are integers from the range [1, 10], as shown in the map. You should assume that the locations go in ascending order and that each location has equal probability. The random number generated for particle i, denoted by ri, is provided in the table below. Please fill in the locations of the eight particles.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Particle | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 |
| ri | 0.914 | 0.473 | 0.679 | 0.879 | 0.212 | 0.024 | 0.458 | 0.154 |
| Location: x0 | 10 | 5 | 7 | 9 | 3 | 1 | 5 | 2 |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

**problem**

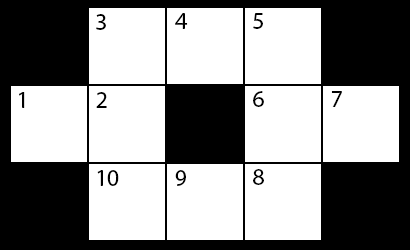
8/8 points

**Part 3: Time update from t=0 ⟶ t=1**

Now we'll perform a time update using the transition model. Stated again, the transition model is as follows: At each timestep, the robot either stays in place, or moves to any one of its neighboring locations, all with equal probability.

For each particle, take the starting position you found in Part 2, and perform the time update for that particle. You should again sample from the range [0, 1), where the bins are the possible locations **sorted in ascending numerical order**. As an example, if Xt = 2, the next state can be one of {1, 2, 3, 10}, each with equal probability, so the [0, 0.25) bin would be for Xt+1 = 1, the [0.25, 0.5) bin would be for Xt+1 = 2, the [0.5, 0.75) bin would be for Xt+1 = 3, and the [0.75, 1) bin would be for Xt+1 = 10.

The map is shown again below:



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Particle | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 |
| ri | 0.674 | 0.119 | 0.748 | 0.802 | 0.357 | 0.736 | 0.425 | 0.058 |
| Location: xi | 10 | 4 | 7 | 10 | 3 | 2 | 5 | 1 |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |
|  |

**problem**

5/5 points

**Part 4: Probability Distribution Induced by the Particles**

Recall that a particle filter just keeps track of a list of particles, but at any given time, we can compute a probability distribution from these particles. Using the current newly updated set of particles (that you found in Part 3) , give the estimated probability that the robot is in each location.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x1 | 1 | 2 | 3 | 4 | 5 |
| P(x1) | 0.125 | 0.125 | 0.125 | 0.125 | 0.125 |
| x1 | 6 | 7 | 8 | 9 | 10 |
| P(x1) | 0 | 0.125 | 0 | 0 | 0.25 |

|  |
| --- |
|  |

**problem**

12/12 points

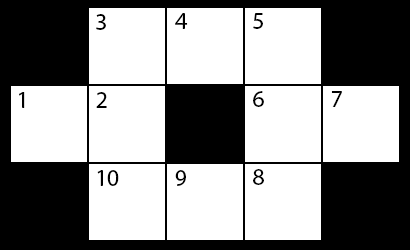
**Part 5: Incorporating Evidence at t=1**

The sensor reading at t=1 is: E1 = D:

Using the sensor model you specified in Part 1, incorporate the evidence by reweighting the particles. Also enter the normalized and cumulative weights for each particle. The normalized weight for a specific particle can be calculated by taking that particle's weight and dividing by the sum of all the particle weights. The cumulative weight keeps track of a running sum of all the weights of the particles seen so far (meaning, particle i at will have a cumulative weight equal to the sum of the weights of all particles j such that j ≤ i).

Refer back to Part 3 to get the positions of your particles.

The map is shown again below:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Particle | p1 | p2 | p3 | p4 |
| Weight | 0.167 | 0.167 | 0.5 | 0.167 |
| Normalized Weight | 0.083 | 0.083 | 0.25 | 0.083 |
| Cumulative Weight | 0.083 | 0.166 | 0.416 | 0.5 |
| Particle | p5 | p6 | p7 | p8 |
| Weight | 0.167 | 0.167 | 0.167 | 0.5 |
| Normalized Weight | 0.083 | 0.083 | 0.083 | 0.25 |
| Cumulative Weight | 0.583 | 0.666 | 0.75 | 1 |

**problem**

8/8 points

**Part 6: Resampling**

Finally, we'll resample the particles. This reallocates resources to the most relevant parts of the state space in the next time update step.

Notice that your cumulative weights effectively tell you where the bins used in resampling the particles lie. For example, for particle 1, you calculated the cumulative weight to be some value, p. Then, on a random value draw, if a value between 0 and p was chosen, you would generate a new particle where particle 1 is. Use these bounds to resample the eight particles. In the "New Particle" row, enter the particle corresponding to the bin that the random value chose. In the "New Location" row, enter the location corresponding to this new particle. You may need to look back at Part 3 to get the locations of the particles.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Particle | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 |
| ri | 0.403 | 0.218 | 0.217 | 0.826 | 0.717 | 0.460 | 0.794 | 0.016 |
| New Particle | 3 | 3 | 3 | 8 | 7 | 4 | 8 | 1 |
| New Location | 7 | 7 | 7 | 1 | 5 | 10 | 1 | 10 |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
|  |
|  |

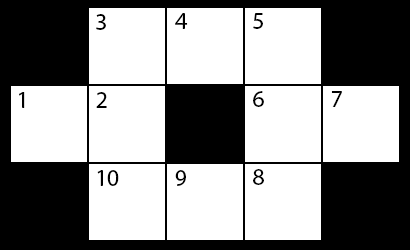
**problem**

1/1 point

**Part 7: Analysis**

The sensor provided a reading E1 = D. What fraction of the particles are now on a dead end?

The map is shown again below:





This completes everything for the first time step, t=0 ⟶ t=1. Of course, we would now continue by repeating the time update, evidence incorporation by reweighting, and resampling. We'll leave that to the computers, though.

### Question 2: Particle Filtering Implementation

0/20 points

Consider the following particle filter implementations.

#### Default Implementation: Resample after Evidence Incorporation

1. Initialize particles by sampling from initial state distribution.
2. Repeat:
   1. Perform time update
   2. Weight according to evidence
   3. Resample according to weights

#### Alternative Implementation: Resample after Time Update

1. Initialize particles by sampling from initial state distribution and assigning uniform weights.
2. Repeat:
   1. Perform time update, retaining weights
   2. Resample according to weights
   3. Weight according to evidence

For each of the following statements about the two implementations, select whether they are true or false.

The default implementation will typically provide a better estimate of the distribution than the alternate implementation.

True False

If the observation model is deterministic then the default implementation will typically provide a better estimate of the distribution than the alternate implementation.

True False

If the transition model is deterministic then the default implementation will typically provide a better estimate of the distribution than the alternate implementation.

True False

### Question 3: D-Separation

30/30 points

You are given several graphical models below, and each graphical model is associated with an independence (or conditional independence) assertion. Please specify if the assertion is true or false.

NOTE: This problem will re-randomize values on reset. If your diagrams are not the same as the ones I show, please cut and paste your correct marked answer here.

