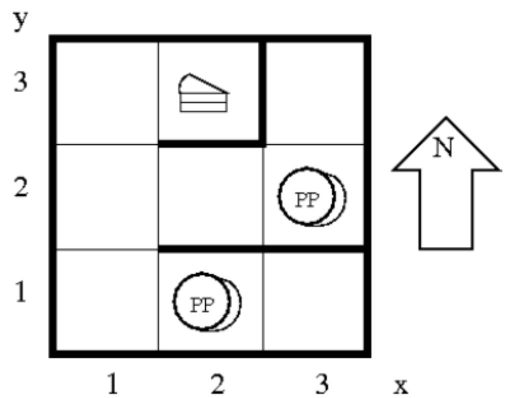
CS 6300 **HW10: Particle Filters and POMDPs** **Due April 27, 2017**

Please use the LaTeX template to produce your writeups. See the Homework Assignments page on the class website for details. Hand in at: `https://webhandin.eng.utah.edu/index.php`.

1 Particle Filtering

PacBot is lost! It was exploring a maze when a solar storm occurred, erasing its memory and leaving it with no information about where it is. Luckily, the map of the maze was stored on its hard drive, and is still available for PacBot to use.



PacBot decides to use particle filtering to figure out where it is. It can make noise-free observations of the number of walls $W \in \{1, 2, 3\}$, and noisy observations of food $F \in \{pie, power-pellet, no-food\}$ at its current location.

If there is food, it detects it correctly with probability $\frac{3}{4}$, and detects nothing with probability $\frac{1}{4}$. An example observation is (power-pellet, 2), which could occur in locations (2,1) or (3,2).

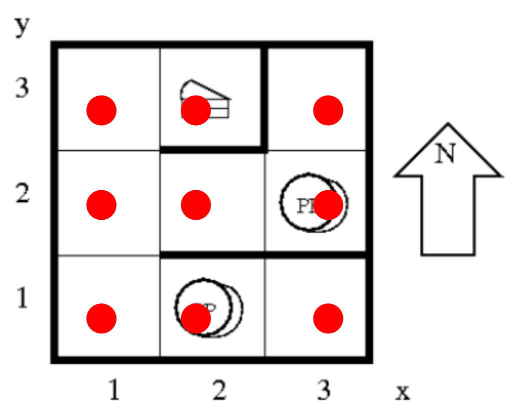
It chooses actions with the following probabilities: N: $\frac{1}{2}$, S: $\frac{1}{6}$, E: $\frac{1}{6}$, W: $\frac{1}{6}$. In other words, if you roll a standard 6 sided die and value is 1, 2, or 3 the particle would move N, it would move S if the value was 4, E if the value was 5, and W if the value was 6. If it tries to make a move and bumps into a wall, it stays where it is, otherwise it moves with no noise. *Important:* When moving the particles start at the top left and iterate row-by-row.

a) Specify (i.e. give numbers for) the emission probabilities $P(E|X)$ associated with the HMM for this problem.

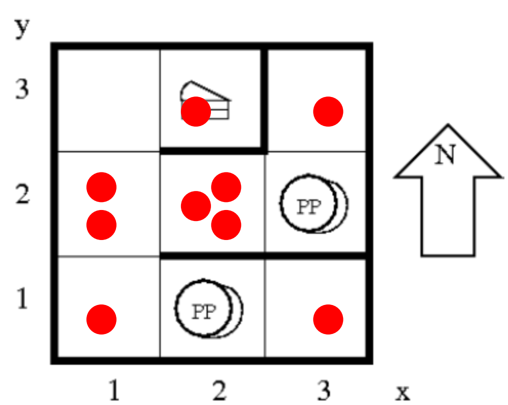
X	$P(E\|X)$
$(1,1)$	$P(\text{no food}, 2\|X) = 1$
$(2,1)$	$P(\text{no food}, 2\|X) = 1/4$, $P(\text{power}, 2\|X) = 3/4$
$(3,1)$	$P(\text{no food}, 3\|X) = 1$
$(1,2)$	$P(\text{no food}, 1\|X) = 1$
$(2,2)$	$P(\text{no food}, 2\|X) = 1$
$(3,2)$	$P(\text{no food}, 2\|X) = 1/4$, $P(\text{power}, 2\|X) = 3/4$
$(1,3)$	$P(\text{no food}, 2\|X) = 1$
$(2,3)$	$P(\text{no food}, 3\|X) = 1/4$, $P(\text{pie}, 3\|X) = 3/4$
$(3,3)$	$P(\text{no food}, 3\|X) = 1$

b) PacBot starts doing particle filtering with 9 particles, one in each location. Perform a single *time* step of particle filtering and indicate where particles moved, using the following random numbers in order: (4, 1, 5, 5, 2, 6, 1, 6, 2). For example, particle 1 is in $(x, y) = (1, 3)$ and will move (or not) to some other square which you'll indicate with a table.

Initially we start off with the following

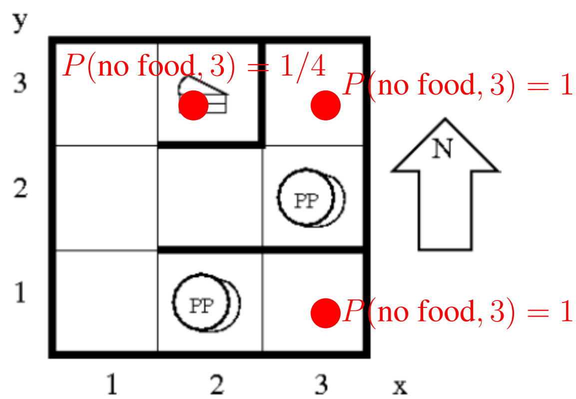


However, after applying the first time step it becomes



c) PacBot now makes the observation (no-food, 3). Complete the evidence step of particle filtering. *Re-weight* your particles from step (b), and normalize them based on the sum of all weights * 9.

In reweighting the particles only based on the evidence we get

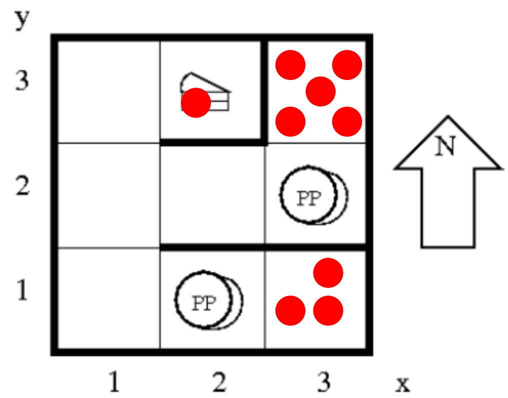


The rest of the weights are 0

d) Now re-sample the particles with 9 new, equally weighted (1.0) particles. To do the re-sampling, use the following pseudo-code:

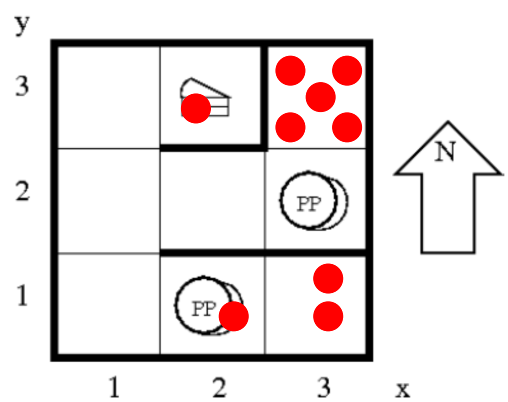
```
for i = 1 → 9 do
  rand ← rand[0, 9]
  sum ← 0
  for all weighted (normalized) particles j do
    sum ← sum + weight of particle j
    if sum ≥ rand then
      place a new particle onto new map at this particle's location
      break
    end if
  end for
end for
```

To help you with the sampling, here are 9 randomly generated numbers between 1 and 9: (2, 9 , 1, 4, 9, 5, 6, 3, 4). Assign values from this probability distribution in ascending order and assign them to cells from left-to-right, top-to-bottom.



e) What do the particles look like after another *time* step? Here are 9 uniformly, randomly chosen numbers between 1 and 6: (4, 3, 1, 5, 3, 6, 3, 6, 5)

Based on the movements, we get S, N, N, E, N, W, N, W, E



2 POMDP

An agent is in one of the two cells s_1, s_2. There are two actions $a \in \{go, stay\}$: the agent can either stay in the cell, or attempt to go to the other cell. The transition probabilities $T(s_i, a, s_j)$ (take action a from state s_i and arrive in state s_j) are:

$$T(s_i, stay, s_j) = \begin{cases} 0 & \text{for} \quad i \neq j \\ 1 & \text{for} \quad i = j \end{cases}$$

$$T(s_i, go, s_j) = \begin{cases} 0.25 & \text{for} \quad i \neq j \\ 0.75 & \text{for} \quad i = j \end{cases}$$

The reward function has the simplified form $R(s_i, a, s_j) = R(s_j)$, i.e., it depends only on the state you end up in. There is a reward for transitioning to state s_2, but none to state s_1:

$$R(s_2) = 1, \quad R(s_1) = 0$$

The agent has an ultrasound sensor which helps to distinguish which cell it's in. There are two possible readings z_1 or z_2 corresponding to an estimation of being in cell s_1 or s_2 respectively, but the sensor is noisy and sometimes gives the wrong reading. Its conditional probability is given by:

$$P(z_i|s_j) = \begin{cases} 0.2 & \text{for} \quad i \neq j \\ 0.8 & \text{for} \quad i = j \end{cases}$$

The agent maintains and updates a belief function $b(s_i)$ based upon combinations of actions and associated sensor readings. For brevity, define $p_1 = b(s_1)$. Hence $b(s_2) = 1 - p_1$.

1. For the first action and without receiving any sensor readings yet, derive the one-time-step utilities $V^{stay}(s_i)$ and $V^{go}(s_i)$, $i = 1, 2$, for actions $stay$ and go.

$$V_{p_i}(s) = r(s, a(p_i)) = \sum_{s'} T(s, a(p_i), s') R(s, a(p_i), s')$$

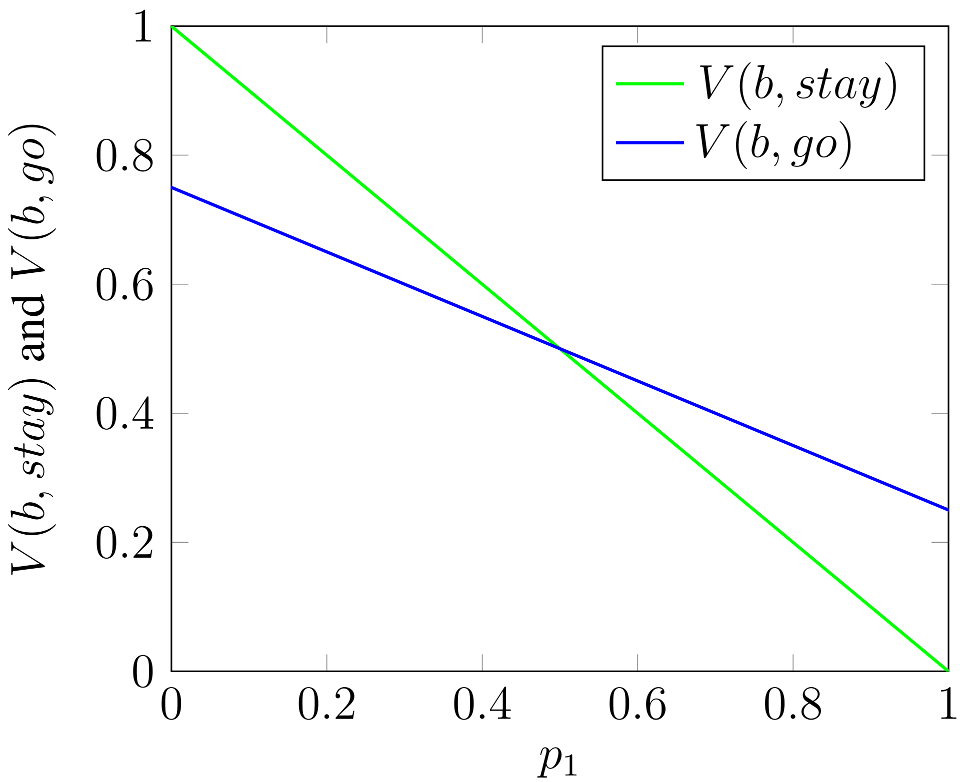
$$\begin{aligned} V^{stay}(s_1) &= 1 \cdot 0 + 0 \cdot 1 = 0 \\ V^{stay}(s_2) &= 1 \cdot 0 + 1 \cdot 1 = 1 \\ V^{go}(s_1) &= 0.75 \cdot 0 + 0.25 \cdot 1 = 0.25 \\ V^{go}(s_2) &= 0.25 \cdot 0 + 0.75 \cdot 1 = 0.75 \end{aligned}$$

2. You don't actually know which state you're in, and you have to use your belief function $b(s_i)$ to combine the results above. Find the expected reward $V(b, go)$ for action go, and $V(b, stay)$ for action $stay$.

$$
\begin{aligned}
V_{p_i}(b) &= \sum_s b(s) V_{p_i}(s) \\
V^{stay}(b, stay) &= V^{stay}(s_1)p_1 + V^{stay}(s_2)(1-p_1) \\
&= 0 \cdot p_1 + 1 \cdot (1-p_1) \\
&= 1 - p_1 \\
V^{go}(b, go) &= V^{go}(s_1)p_1 + V^{go}(s_2)(1-p_1) \\
&= 0.25 \cdot p_1 + 0.75(1-p_1) \\
&= 0.75 - 0.5p_1
\end{aligned}
$$

The expected reward can be found by calculating $E[V(b, go)]$

3. Plot both expected reward functions on the same plot with p_1 on the x-axis. Identify the optimal strategy based on your plot.



From this, we can say that we chose to stay if $p_1 \leq 1/2$ and to go if $p_1 > 1/2$.

4. Suppose you are able to get a sensor reading before taking an action, and you observe z_1. Update your belief to find $p(s_1|z_1)$ and $p(s_2|z_1)$.

   Before doing this we need $p(z_1)$, giving

$$
\begin{aligned}
p(z_1) &= \sum_i p(z_1|s_i)p(s_i) \\
&= p(z_1|s_1)p(s_1) + p(z_1|s_2)p(s_2) \\
&= 0.8p_1 + 0.2(1-p_1)
\end{aligned}
$$

Now, we can solve for $p(s_i|z_1)$, giving

$$
\begin{aligned}
p(s_i|s_1) &= \frac{p(z_1|s_i)p(s_i)}{p(z_1)} \\
p(s_1|z_1) &= \frac{p(z_1|s_1)p(s_1)}{p(z_1|s_1)p(s_1) + p(z_1|s_2)p(s_2)} \\
&= \frac{0.8p_1}{0.8p_1 + 0.2(1-p_1)} \\
p(s_2|z_1) &= \frac{p(z_1|s_2)p(s_2)}{p(z_1|s_1)p(s_1) + p(z_1|s_2)p(s_2)} \\
&= \frac{0.2(1-p_1)}{0.8p_1 + 0.2(1-p_1)}
\end{aligned}
$$

5. Solve for the new value functions given b'.

$$
\begin{aligned}
V(b', stay) &= p(s_1|z_1)V^{stay}(s_1, stay) + p(s_2|z_1)V^{stay}(s_2, stay) \\
&= \frac{0.8p_1}{0.8p_1 + 0.2(1-p_1)}V^{stay}(s_1, stay) + \frac{0.2(1-p_1)}{0.8p_1 + 0.2(1-p_1)}V^{stay}(s_2, stay) \\
&= \frac{0.8p_1}{0.8p_1 + 0.2(1-p_1)} \cdot 0 + \frac{0.2(1-p_1)}{0.8p_1 + 0.2(1-p_1)} \cdot 1 \\
V(b', go) &= p(s_1|z_1)V^{go}(s_1, go) + p(s_2|z_1)V^{go}(s_2, go) \\
&= \frac{0.8p_1}{0.8p_1 + 0.2(1-p_1)}V^{go}(s_1, go) + \frac{0.2(1-p_1)}{0.8p_1 + 0.2(1-p_1)}V^{go}(s_2, go) \\
&= \frac{0.8p_1}{0.8p_1 + 0.2(1-p_1)} \cdot 0.25 + \frac{0.2(1-p_1)}{0.8p_1 + 0.2(1-p_1)} \cdot 0.75
\end{aligned}
$$