Probabilistic Foundations of Artificial Intelligence
Solutions to Problem Set 5

Dec 8, 2017

1. Particle filter

Suppose that you have a robot, which is moving randomly through an 1-dimensional environment. You want to track the robot's position, x, which is discretized to integer values, $x \in \mathbb{Z}$. The robot's movement is modeled as a random walk,

$$x_{t+1} = x_t + \epsilon_t, \tag{1}$$

where ϵ_t is uniformly distributed and can take integer values in $[-3, 3]$. To track the robot, a sensor that measures the distance to the robot has been placed at the origin. The measurement model is

$$y_t = (x_t + \eta_t)^2, \tag{2}$$

where η_t is distributed according to

$$P(\eta_t) = \begin{cases} 0.6 & \text{if } \eta_t = 0 \\ 0.2 & \text{if } |\eta_t| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

You want to use a particle filter with six particles to track the robot's position. At initial time, the robot is at the origin, $x_0 = 0$. Hence, the particles are initialized to $x_i = 0$, $i \in \{0, 1, 2, 3, 4, 5\}$.

(i) You draw samples from the distribution of ϵ_0 and obtain $(-1, -1, 0, 1, 2, 3)$. What is the position of the particles after the prediction update?

(ii) You obtain a measurement, $y_1 = 1$. What are the weights of the individual particles?

(iii) Are five particles enough to accurately estimate the state? Why/Why not?

(iv) Why would a Kalman filter not work reliably in this case?

Solution

(i) Using the movement model,

$$x_1 = x_0 + \epsilon_0, \tag{4}$$

we obtain $\mathbf{x}' = (-1, -1, 0, 1, 2, 3)$.

(ii) From the measurement model we obtain $\eta_t = \pm\sqrt{y_t} - x_t$, and consequently the measurement probability distribution

$$P(y_{t+1} \mid x'_i) = P(\eta_{t+1} = \pm\sqrt{y_{t+1}} - x'_i) \tag{5}$$

The particle weights are computed as $w_i = \frac{1}{Z} P(y_{t+1} \mid x'_i)$

$$n = 0, \quad P(y_1 = 1 \mid x'_0 = -1) = P(\eta_1 = 0) \quad + P(\eta_1 = 2) \quad = 0.6 + 0.0 = 0.6 \tag{6}$$
$$n = 1, \quad P(y_1 = 1 \mid x'_1 = -1) = P(\eta_1 = 0) \quad + P(\eta_1 = 2) \quad = 0.6 + 0.0 = 0.6 \tag{7}$$
$$n = 2, \quad P(y_1 = 1 \mid x'_2 = 0) \quad = P(\eta_1 = -1) + P(\eta_1 = 1) \quad = 0.2 + 0.2 = 0.4 \tag{8}$$
$$n = 3, \quad P(y_1 = 1 \mid x'_3 = 1) \quad = P(\eta_1 = 0) \quad + P(\eta_1 = -2) = 0.6 + 0.0 = 0.6 \tag{9}$$
$$n = 4, \quad P(y_1 = 1 \mid x'_4 = 2) \quad = P(\eta_1 = -1) + P(\eta_1 = -3) = 0.2 + 0.0 = 0.2 \tag{10}$$
$$n = 5, \quad P(y_1 = 1 \mid x'_5 = 3) \quad = P(\eta_1 = -2) + P(\eta_1 = -4) = 0.0 + 0.0 = 0.0 \tag{11}$$

$$Z = \sum_{i=0}^{N} P(y_1 = 1 \mid x'_i) = 0.6 + 0.6 + 0.4 + 0.6 + 0.2 + 0.0 = \frac{24}{10} \tag{12}$$

Consequently, we can calculate the weights with $w_i = \frac{1}{Z} P(y_1 = 1 \mid x'_i)$

$$w_0 = \frac{6}{24},\ w_1 = \frac{6}{24},\ w_2 = \frac{4}{24}.\ w_3 = \frac{6}{24},\ w_4 = \frac{2}{24},\ w_5 = 0 \tag{13}$$

(iii) No, because we cannot even capture the probability distribution of the movement prediction accurately (uniform distribution). We need more samples to accurately estimate the state.

(iv) A Kalman filter can only describe Gaussian distributions (unimodal). Here, the noise is not Gaussian and the measurements are nonlinear. Furthermore, the distance measurements cannot break the symmetry in the problem, so that the posterior state distribution after one step is bimodal.

2. Hero in the maze

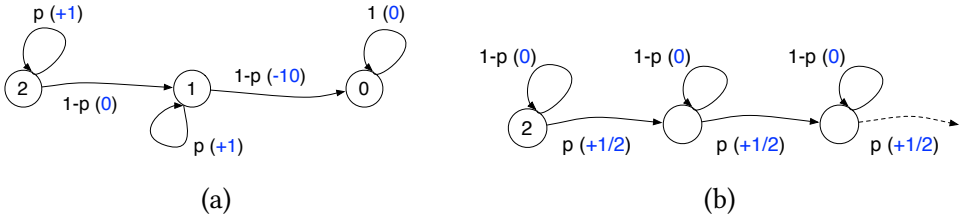
Consider the following problem related to probabilistic planning. You are a hero H who is being chased by a ghost G in a maze.

(i) Suppose the maze is a simple (infinite) chain of nodes, each node labeled with a number (from $-\infty$ to ∞): H starts at 0, G starts at -2. H always tries to move away from G, but only succeeds with probability p, and with probability $1 - p$ gets stuck (i.e., with probability p, H moves 1 step to the right, from node i to $i + 1$, and gets 1 unit of reward; with probability $1 - p$, H doesn't change its location and gets 0 units of reward). G always chases after H and never gets stuck. If G catches H, H incurs -10 reward in the timestep in which it got caught (and 0 reward in all subsequent time steps). Both G and

H move simultaneously. Write down the state space with the transition probabilities. For a discount factor γ, what is the expected long term future reward as a function p and γ? Calculate its value for $p = .9$ and $\gamma = .95$. *Hint: You may want to consider the relative positions of H and G instead of their absolute positions when choosing your state representation.*

(ii) Now, suppose the maze is a "T", i.e, an (infinitely large) tree, where only one node, the starting node of H, has degree 3, all other nodes have degree 2. In the first round, H has the choice of either moving "right" and being chased (the same as above); or moving "down" and not being chased. If H moves "down", it will also get stuck with probability $1 - p$ like above, but only incur reward $1/2$ for each step moved (which happens with probability p. In all subsequent actions, H continues to (attempt to) move in the same direction as in the first round (i.e., once it decides to move right, it has to continue to move right etc.) What is the expected long term future reward in this case, as a function of p and γ? Calculate its value for $p = .9$ and $\gamma = .95$. For these values of p and γ, which initial action should H take? For a value of $\gamma = .95$, give an explicit rule on how H should choose its initial action as a function of p. Compute the critical (decision-relevant) values of p (you may have to do this numerically).

Solution



(a) (b)

Figure 1: (a) Hero moving right; (b) hero moving down

(i) Both Ghost and Hero can only move right (hero gets stuck randomly).

State space is the set of all possible relative positions of H to G: $X = \{2, 1, 0\}$. There is only one action that the hero will move forward (although sometimes he may get stuck). Transition probabilities are labelled as Figure 1 (a):

$$
\begin{aligned}
&P(Next\ state = 2 | state = 2) = p \\
&P(Next\ state = 1 | state = 2) = 1 - p \\
&P(Next\ state = 1 | state = 1) = p \\
&P(Next\ state = 0 | state = 1) = 1 - p \\
&P(Next\ state = 0 | state = 0) = 1
\end{aligned}
$$

For the optimal policy π^* it holds (Bellman equation)

$$V^*(x) = \max_{a \in A(x)} \sum_{x'} P(x'|x, a) \Big(r(a, x, x') + \gamma V^*(x') \Big)$$

The value functions for (ending up at) the three states $2,\ 1,\ 0$ will converge to some constant values $V(2),\ V(1),\ V(0)$. Therefore,

$$V(0) = 0 \tag{14}$$
$$V(1) = p(1 + \gamma V(1)) + (1 - p)(-10 + \gamma V(0))) \tag{15}$$
$$V(2) = p(1 + \gamma V(2)) + (1 - p)(0 + \gamma V(1)) \tag{16}$$

Solve the equations, so that the long-term future reward $V(1) = -0.6897, V(2) = 5.7551$.

(ii) Hero can choose to move down in the first round.

As Figure 1 (b) shows, when hero moves down, all following states has exactly the same transition probabilities. Thus if the value function converges, it should be a constant value V^* for all states when $t \to \infty$. Using Bellman equation, we have:

$$V^* = \frac{p}{2} + \gamma\{(1 - p)V^* + pV^*\} \Longrightarrow V^* = \frac{1/2 * p}{1 - \gamma} = \frac{0.5 * 0.9}{1 - 0.95} = 9.$$

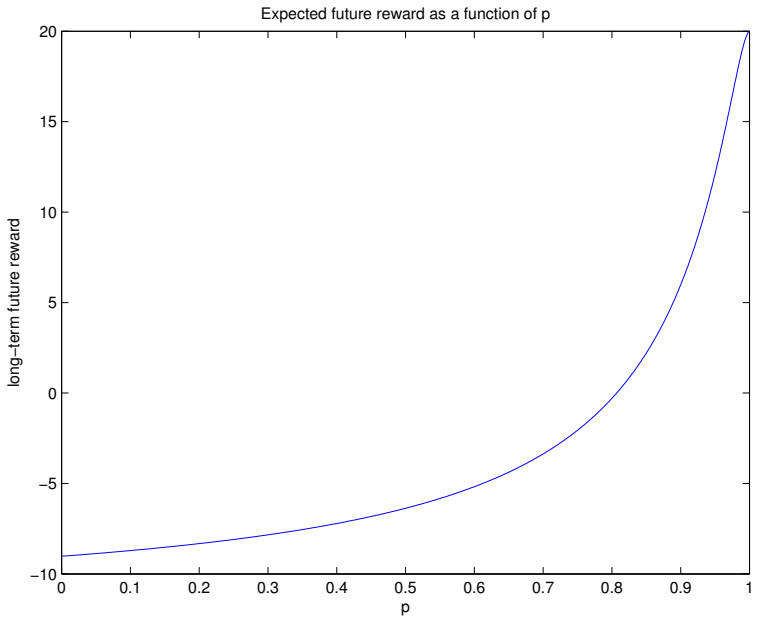


Figure 2: Expected long-term reward as a function of p. $\gamma = .95$

Figure 2 shows the expected rewards for moving right and moving down. When $p \geq 0.9277$, hero chooses to move right; otherwise, hero moves down.

3. Policy iteration

Consider an undiscounted MDP having three states, (1, 2, 3), with rewards -1, -2, 0, respectively. State 3 is a terminal state. In states 1 and 2 there are two possible actions: a and b. The transition model is as follows:

- In state 1, action a moves the agent to state 2 with probability 0.8 and makes the agent stay put with probability 0.2.
- In state 2, action a moves the agent to state 1 with probability 0.8 and makes the agent stay put with probability 0.2.
- In either state 1 or state 2, action b moves the agent to state 3 with probability 0.1 and makes the agent stay put with probability 0.9.

Answer the following questions:

(i) Draw the MDP described above. What can be determined *qualitatively* about the optimal policy in states 1 and 2?

(ii) Apply policy iteration, showing each step in full, to determine the optimal policy and the values of states 1 and 2. Assume that the initial policy has action b in both states.

(iii) What happens to policy iteration if the initial policy has action a in both states? Does discounting help? Does the optimal policy depend on the discount factor?

Solution

(i) Intuitively, the agent wants to get to state 3 as soon as possible, because it will pay a cost for each time step it spends in state 1 and state 2. However, the only action that reaches state 3 (action b) succeeds with low probability, so the agent should minimize the cost it incurs while trying to reach the terminal state. This suggests that the agent should definitely try action b in state 1; in state 2, it might be better to try action a to get to state 1 (which is the better place to wait for admission to state 3), rather than aiming directly for state 3. The decision in state 2 involves a numerical tradeoff.

(ii) The application of policy iteration precedes in alternating steps of value determination and policy update.

- Initialization: $U \leftarrow \langle -1, -2, 0 \rangle$, $P \leftarrow \langle b, b \rangle$.
- Value determination: Write out the equations in terms of the values (rewards and transition probabilities are known for a fixed policy $\pi(x)$)

$$u(x) = r(x) + \sum_{x'} P(x'|x, \pi(x))u(x')$$

$$\begin{aligned} u_1 &= -1 + 0.1u_3 + 0.9u_1 \\ u_2 &= -2 + 0.1u_3 + 0.9u_2 \\ u_3 &= 0 \end{aligned}$$

Which have the solution, $u_1 = -10$ and $u_2 = -20$.
Policy update:
The reward is not dependent on the action, so it does not affect the maximization problem and is neglected in the following. In state 1,

$$\sum_j T(1, a, j)u_j = 0.8 \times -20 + 0.2 \times -10 = -18$$

while

$$\sum_j T(1, b, j)u_j = 0.1 \times 0 + 0.9 \times -10 = -9$$

so action b is preferred for state 1.
In state 2,

$$\sum_j T(1, a, j)u_j = 0.8 \times -10 + 0.2 \times -20 = -12$$

while

$$\sum_j T(1, a, j)u_j = 0.1 \times 0 + 0.9 \times -20 = -18$$

so action a is preferred for state 2. We set $unchanged? \leftarrow$ false and proceed.

- Value determination:

$$\begin{aligned} u_1 &= -1 + 0.1u_3 + 0.9u_1 \\ u_2 &= -2 + 0.8u_1 + 0.2u_2 \\ u_3 &= 0 \end{aligned}$$

once more, $u_1 = -10$; now, $u_2 = -12.5$.
Policy update:
In state 1,

$$\sum_j T(1, a, j)u_j = 0.8 \times -15 + 0.2 \times -10 = -14$$

while

$$\sum_j T(1, b, j)u_j = 0.1 \times 0 + 0.9 \times -10 = -9$$

so action b is still preferred for state 1.
In state 2,

$$\sum_j T(1, a, j)u_j = 0.8 \times -10 + 0.2 \times -12.5 = -10.5$$

while

$$\sum_j T(1, a, j)u_j = 0.1 \times 0 + 0.9 \times -12.5 = -11.25$$

so action a is still preferred for state 2. $unchanged?$ remains true, and we terminate.

Note that the resulting policy matches our intuition: when in state 2, try to move to state 1, and when in state 1, try to move to state 3.

(iii) An initial policy with action a in both states leads to an unsolvable problem. The initial value determination problem has the form

$$\begin{aligned} u_1 &= -1 + 0.2u_1 + 0.8u_2 \\ u_2 &= -2 + 0.8u_1 + 0.2u_2 \\ u_3 &= 0 \end{aligned}$$

and the first two equations are inconsistent. If we were to try to solve them iteratively, we would find the values tending to $-\infty$.

Discounting leads to well-defined solutions by bounding the penalty (expected discounted cost) an agent can incur at either state. However, the choice of disc out factor will affect the policy that results. For γ small, the cost incurred in the distant future plays a negligible role in the value computation, because γ^n is near 0. As a result, an agent could choose action b in state 2 because the discounted short-term cost of remaining in the non-terminal states (states 1 and 2) outweighs the discounted long-term cost of action b failing repeatedly and leaving the agent in state 2 (as an additional exercise, you can decide the value of γ at which the agent is indifferent between the two choices).

4. Value iteration

In finite MDPs, the value function can be expressed as a vector that has as many entries as states in the state space, X. Given a value vector V, we defined the Bellman update operator, $\mathcal{B}(\cdot)$, for every element of V as follows:

$$\mathcal{B}(V(x)) = \max_a \big(r(x,a) + \gamma \sum_{x'} P(x'|x,a)V(x')\big). \tag{17}$$

Show that the Bellman operator is a contraction with respect to $\|\cdot\|_\infty$, that is to say that, for any V, V', holds that:

$$\max_{x\in X} |\mathcal{B}(V(x)) - \mathcal{B}(V'(x))| = \|\mathcal{B}V - \mathcal{B}V'\|_\infty \leq \gamma \|V - V'\|_\infty. \tag{18}$$

Solution

By considering a generic $x \in X$ we can write $|\mathcal{B}(V(x)) - \mathcal{B}(V'(x))|$ as:

$$\begin{aligned} &|\max_a \big(r(x,a) + \gamma \sum_{x'} P(x'|x,a)V(x')\big) - \max_a \big(r(x,a) + \gamma \sum_{x'} P(x'|x,a)V'(x')\big)|, \\ &\leq |\max_a \big(r(x,a) + \gamma \sum_{x'} P(x'|x,a)V(x') - r(x,a) + \gamma \sum_{x'} P(x'|x,a)V'(x')\big)|, \\ &= \gamma |\max_a \big(\sum_{x'} P(x'|x,a)(V(x') - V'(x'))\big)|, \\ &= \gamma |\sum_{x'} P(x'|x,a^*)(V(x') - V'(x'))|, \end{aligned}$$

where a^* is the action that attains the maximum. At this point, it is important to notice that $V(x') - V'(x') <= \|V - V'\|_\infty$ by definition. Furthermore, remember that $P(x'|x,a) \geq 0$ for every $x' \in X$ and that $\sum_{x'} P(x'|x,a) = 1$. These statements allow us to say the following:

$$\begin{aligned}
|\mathcal{B}(V(x)) - \mathcal{B}(V'(x))| &\leq \gamma |\sum_{x'} P(x'|x,a^*)(V(x') - V'(x'))|, \\
&\leq \gamma \|V - V'\|_\infty |\sum_{x'} P(x'|x,a^*)|, \\
&= \gamma \|V - V'\|_\infty.
\end{aligned}$$

We proved that this inequality holds for a generic $x \in X$. This means that it must hold in particular for the value that attains the maximum of the left hand side of the inequality:

$$|\|\mathcal{B}V - \mathcal{B}V'\|_\infty = \max_{x \in X} |\mathcal{B}(V(x)) - \mathcal{B}(V'(x))| \leq \gamma \|V - V'\|_\infty. \tag{19}$$