Introduction to Artificial Intelligence (INFO8006)

Exercises 6 – Making decisions

December 1, 2021

Learning outcomes

At the end of this session you should be able to

- formulate a Markov decision process (MDP);
- define the expected utility of a policy;
- define the value of a state and an optimal policy;
- define and apply the value iteration and policy iteration algorithms;
- formulate a partially observable Markov decision process (POMDP).

Exercise 1 Micro-Blackjack (UC Berkeley CS188, Spring 2014)

In Micro-Blackjack, you repeatedly draw a card (with replacement) that is likely to be a 2, 3 or 4. At each step, if the total score of the cards is higher or equal than 6, you can either "draw" (d) or "cash" (c). Otherwise, you can only cash. When you cash, the game stops and your utility is equal to your total score (up to 5), or zero if you get a total of 6 or higher. Until you cash and after it, you receive no utility. There is no discount ($\gamma = 1$).

1. Formalize Micro-Blackjack as an MDP.

    A Markov decision process (MDP) is a tuple (S, A, P, R), where S is a set of reachable states, $A(s)$ is a set of available actions, $P(s'|s, a)$ is a (stationary) transition model and $R(s)$ is a reward function.

    In our case,

    $$S \subset \mathcal{N} \times \{0, 1\}$$

    $$A(s = (s_1, s_2)) = \begin{cases} \{d, c\} & \text{if } s_1 < 6 \text{ and } s_2 = 0 \\ \{c\} & \text{otherwise} \end{cases}$$

    $$P(s' = (s_1', s_2')|s = (s_1, 0), a) = \begin{cases} 1 & \text{if } s_1' = s_1 \text{ and } s_2' = 1 \text{ and } a = c \\ \frac{1}{3} & \text{if } s_1' - s_1 \in \{2, 3, 4\} \text{ and } s_2' = 0 \text{ and } a = d \\ 0 & \text{otherwise} \end{cases}$$

    $$R(s = (s_1, s_2)) = \begin{cases} s_1 & \text{if } s_1 < 6 \text{ and } s_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

2. Derive the optimal policy for this MDP.

    Let $(s_0, s_1, \ldots, s_l)$ be a sequence of states, *i.e.* a path. Its utility is

    $$\sum_{t=0}^{l} \gamma^t R(s_t),$$

    where $\gamma \in (0, 1]$ is the discount factor. For a policy $\pi : S \mapsto A$ to be optimal, it should

maximize its *expected* utility

$$V^\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t R(s_t)\right]$$
$$= \max_\pi R(s_0) + \gamma \, \mathbb{E}[V^\pi(s_1)]$$
$$= R(s_0) + \gamma \sum_{s_1} P(s_1|s_0, \pi(s_0))V^\pi(s_1),$$

where the states $s_{t+1} \sim P(s|s_t, \pi(s_t))$. Therefore, the value $V(s)$ of a state s is its maximum expected utility, *i.e.*

$$V(s) = V^{\pi^*}(s)$$
$$= \max_\pi V^\pi(s)$$
$$= \max_\pi R(s) + \gamma \sum_{s'} P(s'|s, \pi(s))V^\pi(s')$$
$$= R(s) + \gamma \max_\pi \sum_{s'} P(s'|s, \pi(s))V^\pi(s')$$
$$= R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a) \max_\pi V^\pi(s')$$
$$= R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a)V(s'),$$

where π^* is the optimal policy

$$\pi^*(s) = \arg\max_{a \in A(s)} \sum_{s'} P(s'|s, a)V(s').$$

To find this optimal policy, we can apply either the *value iteration* or *policy iteration* algorithms.

- In the former, we estimate the value $V(s)$ by $V_i(s)$ and update simultaneously all the estimates with

$$V_{i+1}(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a)V_i(s')$$

until convergence. An iteration of this algorithm takes $\mathcal{O}(|S|^2|A|)$ operations.

- In the latter, we estimate the optimal policy π^* by π_i and update the policy with

$$\pi_{i+1}(s) = \arg\max_{a \in A(s)} \sum_{s'} P(s'|s, a)V^{\pi_i}(s'),$$

where V^{π_i} is such that

$$V^{\pi_i}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s))V^{\pi_i}(s').$$

Finding V^{π_i} exactly requires $\mathcal{O}(|S|^3)$ operations, but $\mathcal{O}(k|S|^2)$ approximations are sufficient. Overall an iteration of the (modified) policy iteration costs $\mathcal{O}(|S||A|+k|S|^2)$ operations.

For our task, it is easier to perform the policy iteration algorithm. Let be the first policy to always cash ($\pi_0(s) = c$). It should be noted that

$$V^{\pi_i}(s = (s_1, 1)) = R(s = (s_1, 1)) + \gamma \times 0,$$

meaning that we don't need to consider the states $s = (s_1, 1)$ in our algorithm.

s_1	0	2	3	4	5	6+
$\pi_0(s), V^{\pi_0}(s)$	$c, 0$	$c, 2$	$c, 3$	$c, 4$	$c, 5$	$c, 0$
$\pi_1(s), V^{\pi_1}(s)$	$d, 3$	$d, 3$	$c, 3$	$c, 4$	$c, 5$	$c, 0$
$\pi_2(s), V^{\pi_2}(s)$	$d, \frac{10}{3}$	$d, 3$	$c, 3$	$c, 4$	$c, 5$	$c, 0$

Exercise 2   Pursuit Evasion (UC Berkeley CS188, Spring 2014)

Pacman is trapped in the following 2 by 2 maze with a hungry ghost. When it is his turn to move, Pacman must move one step horizontally or vertically to a neighboring square. When it is the ghost's turn, he must also move one step horizontally or vertically. The ghost and Pacman alternate moves. After every move (by either the ghost or Pacman), if Pacman and the ghost occupy the same square, Pacman is eaten and receive utility -100. Otherwise, he receives utility of 1. The ghost attempts to minimize the utility that Pacman receives. Pacman makes the first move. The game is not guaranteed to terminate.

For example, with a discount factor of $\gamma = 1$, if Pacman moves right, ghost moves down, then Pacman moves left, Pacman earns a reward of 1 initially ($R(s_0)$) and after the two first moves and -100 after the last move, for a total utility of -97.

1. Assume a discount factor $\gamma = 0.5$, where the discount factor is applied once every time either Pacman or the ghost moves. What is the Minimax value of the complete (infinite) game?

   If we assume the game starts from the displayed state s_0, Pacman has always an action to get out of reach of the ghost. Hence, if Pacman is optimal, which is true by definition in Minimax, he will never get eaten. Therefore, we can compute the initial state value

   $$V(s_0) = \sum_{t=0}^{\infty} \gamma^t \underbrace{R(s_t)}_{1} = \frac{1}{1 - \gamma} = 2,$$

   without actually building the (infinite) Minimax tree.

2. Why are the value/policy iteration algorithms superior to Minimax for solving this game?

   If $\gamma < 1$, we know that the value/policy iteration algorithms converge toward a fixed point, even if the game does not terminate. Conversely, Minimax is not guaranteed to terminate due to its recursivity.

This game is similar to an MDP because rewards are earned at every timestep. However, it is also an adversarial game involving decisions by two agents. Let s be the state (*e.g.* the positions of Pacman and the ghost and who is playing) and let $A(s)$ be the set of actions available to the player in state s. Let $s' = T(s, a)$ denote the successor of a state s resulting from the action a. Finally, let $R(s)$ denote the utility received after moving to state s.

4. Write down an expression (analogous to Bellman equation) for $V(s)$, the value of the current state if it is Pacman's turn.

   $$V(s) = R(s) + \gamma \max_{a \in A(s)} \left[R(T(s, a)) + \gamma \min_{a' \in A(T(s,a))} V(T(T(s, a), a')) \right]$$

Exercise 3   Crying Baby Problem

To earn some money you have decided to do babysitting. Tonight you will take care of a 9 months old baby. His parents asked you to feed him when he is hungry. Because you are a super-babysitter you know from your experience that the probability of the baby crying when he is hungry is equal to 0.8 and to cry when he is not hungry is equal to 0.1. The cost of feeding the baby is equal to 5 $ whereas if you don't feed him while he is hungry it costs you 10 $ (because the parent will be upset and will reduce your pay). You will have many things to do tonight and so you decide to optimize your time by just checking every hour whether the baby is crying. Between two intervals of time you assume that the baby has a probability of 0.2 to get hungry. You can also assume that the baby has a uniform probability of being hungry when you start your babysitting.

1. Draw the dynamic Bayesian network as well as the conditional probability tables associated with your babysitting of tonight.

   Let $H_t \in \{0, 1\}$ indicate whether the baby is hungry at time t, $C_t \in \{0, 1\}$ the baby is crying at time t and $F_t \in \{0, 1\}$ the fact that you fed the baby at time t. Your cost is $R_t \in \{0, -5, -10\}$.
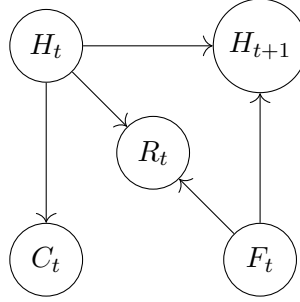
H_t	F_t	$P(C_t = 1 \| H_t)$	$P(H_{t+1} = 1 \| H_t, F_t)$	$R_t(H_t, F_t)$
0	0	0.1	0.2	0
1	0	0.8	1	-10
0	1	-	0	-5
1	1	-	0	-5

   For completeness we also have to define the prior $P(H_0 = 1) = 0.5$. Note that R_t is a deterministic function of H_t and F_t instead of a distribution.

2. Give the sequence of belief states if you assume the sequence of observation-action pairs $(0, 0)$, $(1, 1)$, $(0, 0)$, $(0, 0)$.

   In this problem, the environment is only *partially* observable (POMDP). You, the agent, does not know the current state $s = H_t$. Instead, you collect an observation $e = C_t$ through a sensor model $P(e|s)$ which allows to reason about the unknown state s and to take an informed action $a = F_t$.

   If b is the current belief state (a distribution over the state space S) and the agent does an action a and observes e', then

   $$b'(s') = \sum_s P(s'|s, a, e')b(s)$$
   $$= \alpha P(e'|s') \sum_s P(s'|s, a)b(s)$$

is the next belief state. Within the formulation of our problem, this gives

$$b_{t+1}(h_{t+1}) = \alpha P(c_{t+1}|h_{t+1}) \sum_{h_t} P(h_{t+1}|h_t, f_t) b_t(h_t),$$

with the initial belief

$$b_0 = P(H_0|c_0)$$
$$= \alpha P(c_0|H_0) P(H_0)$$

Then, knowing from the statement that $c_{0:3} = f_{0:3} = (0, 1, 0, 0)$, we have

$$b_0 = \alpha_0 \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \approx \begin{pmatrix} 0.8181 \\ 0.1819 \end{pmatrix}$$

$$b_1 = \alpha_1 \begin{pmatrix} 0.1 & 0 \\ 0 & 0.8 \end{pmatrix} \begin{pmatrix} 0.8 & 0 \\ 0.2 & 1 \end{pmatrix} b_0 \approx \begin{pmatrix} 0.1915 \\ 0.8085 \end{pmatrix}$$

$$b_2 = \alpha_2 \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} b_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$b_3 = \alpha_3 \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.8 & 0 \\ 0.2 & 1 \end{pmatrix} b_2 \approx \begin{pmatrix} 0.9474 \\ 0.0526 \end{pmatrix}.$$

3. From the sequence of belief states, determine the sequence of expected rewards.

The expected reward of a belief state b is defined as

$$\rho(b) = \sum_s b(s) R(s).$$

In our case, the expected reward also depend on the action, that is

$$r_t = \sum_{h_t} R_t(h_t, f_t) b(h_t) = R_t(H_t, f_t)^T b_t.$$

Then, by substitution,

$$r_0 = \begin{pmatrix} 0 & -10 \end{pmatrix} b_0 \approx -1.819$$
$$r_1 = \begin{pmatrix} -5 & -5 \end{pmatrix} b_1 = -5$$
$$r_2 = \begin{pmatrix} 0 & -10 \end{pmatrix} b_2 = 0$$
$$r_3 = \begin{pmatrix} 0 & -10 \end{pmatrix} b_3 \approx -0.526.$$

Exercise 4 (AIMA, Ex 16.15)

Consider a student, Sam, who has the choice to buy or not buy a textbook for a course. We'll model this as a decision problem with one Boolean decision node, B, indicating whether the agent chooses to buy the book, and two Boolean chance nodes, M, indicating whether the student has mastered the material of the course, and F, indicating whether the student fails the course. There is also a utility node, U. Sam has an additive utility function: 0 for not buying the book and $-100\,\$$ for buying it; and 0 for failing the course and $2000\,\$$ for passing it. Sam's conditional probability estimates are as follows:

B	M	$P(M\vert B)$	$P(F=1\vert B, M)$
0	0	0.3	0.7
1	0	0.1	0.5
0	1	0.7	0.2
1	1	0.9	0.1

1. Draw the decision network for this problem.

2. Compute the expected utility of buying the book and of not buying it.

3. What should Sam do?

Exercise 5   (AIMA, Ex 17.6)

The slides of the course states that the Bellman operator

$$(BV)(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a)V(s')$$

is a contraction.

1. Show that, for any two functions $f : \mathbb{R} \mapsto \mathbb{R}$ and $g : \mathbb{R} \mapsto \mathbb{R}$,

$$|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|.$$

2. Apply the previous result to prove that the Bellman operator is a contraction of factor γ, that is

$$\|(BV) - (BV')\|_\infty \leq \gamma \|V - V'\|_\infty.$$

The infinity norm of a function $f : \mathbb{R} \mapsto \mathbb{R}$ is defined as

$$\|f\|_\infty = \sup_x |f(x)|.$$

Exercise 6   (AIMA, Ex 17.8)

Consider the 3×3 world shown in Figure 1. The transition model is the same as in the 4×3 2: $80\,\%$ of the time the agent goes in the direction it selects; the rest of the time it moves at right angles to the intended direction.
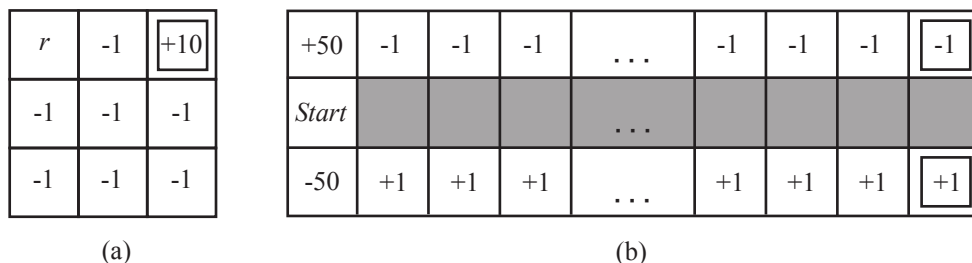
r	-1	+10
-1	-1	-1
-1	-1	-1

(a)

+50	-1	-1	-1	. . .	-1	-1	-1	-1
Start				. . .				
-50	+1	+1	+1	. . .	+1	+1	+1	+1

(b)

Figure 1. (a) 3×3 world for Exercise 6. The reward for each state is indicated. The upper right square is a terminal state. (b) 101×3 world for Exercise 7. The start state has reward 0.
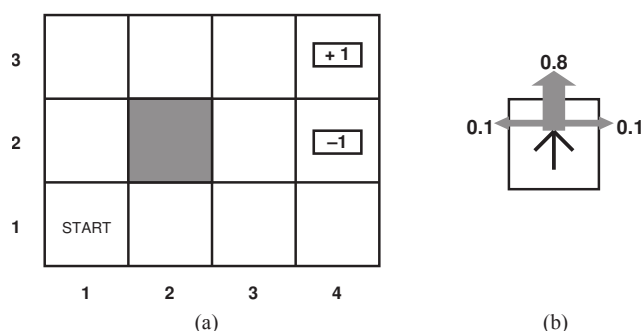

(a)

(b)

Figure 2. (a) A simple 4×3 environment that presents the agent with a sequential decision problem. The two terminal states have reward $+1$ and -1, respectively, and all other states have a reward of -0.04. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement.

Implement value iteration for this world for each value of r below. Use discounted rewards with a discount factor $\gamma = 0.99$. Show the policy obtained in each case. Explain intuitively why the value of r leads to each policy.

1. $r = -100$

2. $r = -3$

3. $r = 0$

4. $r = +3$

Exercise 7 (AIMA, Ex 17.9)

Consider the 101×3 world shown in 1. In the start state the agent has a choice of two deterministic actions, up or down, but in the other states the agent has one deterministic action, right. Assuming a discounted reward function, compute the utility of each action as a function of γ. For what values of the discount factor γ should the agent choose up and for which down?

This simple example actually reflects many real-world situations in which one must weigh the value of an immediate action versus the potential long-term consequences, such as choosing to dump pollutants into a lake.

Exercise 8 (AIMA, Ex 17.10)

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:

1. What can be determined *qualitatively* about the optimal policy in states 1 and 2?

2. 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.

3. 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?

Exercise 9 (AIMA, Ex 17.19)

A Dutch auction is similar to an English auction, but rather than starting the bidding at a low price and increasing, the seller starts at a high price and gradually lowers the price until some buyer is willing to accept that price. If multiple bidders accept the price, one is arbitrarily chosen as the winner. More formally, the seller begins with a price p and gradually lowers p by increments of d until at least one buyer accepts the price.

Assuming all bidders act rationally, is it true that for arbitrarily small d, a Dutch auction will always result in the bidder with the highest value for the item obtaining the item? If so, show mathematically why. If not, explain how it may be possible for the bidder with highest value for the item not to obtain it.

Exercise 10   (AIMA, Ex 17.21)

Teams in the National Hockey League historically received 2 points for winning a game and 0 for losing. If the game is tied, an overtime period is played; if nobody wins in overtime, the game is a tie and each team gets 1 point. But league officials felt that teams were playing too conservatively in overtime (to avoid a loss), and it would be more exciting if overtime produced a winner. So in 1999 the officials experimented in mechanism design: the rules were changed, giving a team that loses in overtime 1 point, not 0. It is still 2 points for a win and 1 for a tie.

1. Was hockey a zero-sum game before the rule change? After?

2. Suppose that at a certain time t in a game, the home team has probability p of winning in regulation time, probability $0.78 - p$ of losing, and probability 0.22 of going into overtime, where they have probability q of winning, $0.9 - q$ of losing, and 0.1 of tying. Give equations for the expected value for the home and visiting teams.

3. Imagine that it were legal and ethical for the two teams to enter into a pact where they agree that they will skate to a tie in regulation time, and then both try in earnest to win in overtime. Under what conditions, in terms of p and q, would it be rational for both teams to agree to this pact?

4. Some experts report that since the rule change, the percentage of games with a winner in overtime went up $18.2\,\%$, as desired, but the percentage of overtime games also went up $3.6\,\%$. What does that suggest about possible collusion or conservative play after the rule change?

Supplementary materials

- Uniform norm

- Dutch auction

- Chapters 16 and 17 of the reference textbook.