1 Making decisions (29/11/2018)

1.1 Objectives

At the end of this repetition you should be able to:

- Define a Markov Decision Process (MDP)
- Define what is the reward, the Expected Utility, an Optimal Policy.
- Define the Bellman equations of a simple MDP and apply value iteration.
- Define and apply policy iteration.
- Define a Partially Observable Markov Decision Process.
- Explain the difference between Model-based and Model-free reinforcement algorithms.

1.2 Exercises

a MDPs: Micro-Blackjack [1] (≈ 20 min)

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. Otherwise, you must Stop. When you Stop, your utility is equal to your total score (up to 5), or zero if you get a total of 6 or higher. When you Draw, you receive no utility. There is no discount ($\gamma = 1$).

1. Formalise the associated MDP.

   - The set of states: $\mathcal{S} = (s_1, s_2) = \{0, 2, 3, 4, 5, 6^+\} \times \{\text{Playing}, \text{Done}\}$. It represents the current sum of cards and if the player is still playing.
   - The set of actions: $\mathcal{A} = \{\text{Draw}, \text{Stop}\}$.
   - The transition model: Let $s = (s_1, s_2)$ and $s' = (s'_1, s'_2)$.
     If s_2 is equal to Done then:

$$P(s'|s, a) = \begin{cases} 1, & \text{if } s'_1 = s_1,\ s'_2 = \text{Done and whatever a.} \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

     .
     Else if s_2 is equal to Playing then:

$$P(s'|s, \text{Draw}) = \begin{cases} \frac{1}{3}, & \text{if } s'_1 - s_1 \in \{2, 3, 4\} \text{ and } s'_2 = \text{Playing.} \\ \frac{2}{3}, & \text{if } s_1 = 3 \text{ and } s' = 6^+ \text{ and } s'_2 = \text{Playing.} \\ 1, & \text{if } s_1 \in \{4, 5\} \text{ and } s'_1 = 6^+ \text{ and } s'_2 = \text{Playing.} \\ 0, & \text{Otherwise.} \end{cases} \tag{2}$$

$$P(s'|s, \text{Stop}) = \begin{cases} 1, & \text{If } s'_1 = s_1 \text{ and } s'_2 = \text{Done.} \\ 0, & \text{Otherwise.} \end{cases} \tag{3}$$

     .
   - The reward function:

$$R(s) = \begin{cases} s_1, & \text{if } s_1 \in \{2, 3, 4, 5\} \text{ and } s_2 = \text{Done.} \\ 0, & \text{Otherwise.} \end{cases} \tag{4}$$

2. Give the optimal policy for this MDP.
   The Policy iteration algorithm gives the sequence of policies of Table 1 if we initialize the policy by always doing the action Stop.

[1] source: `http://ai.berkeley.edu/sections/section_4_9vzQ4e7e6xqagFuQDinu0lmeSEI29Z.pdf`

s_2	Playing						Done					
s_1	0	2	3	4	5	6^+	0	2	3	4	5	6^+
$Action : V_0$	S:0	S:2	S:3	S:4	S:5	S:0	S:0	S:2	S:3	S:4	S:5	S:0
$Action : V_1$	D:3	D:3	S:3	S:4	S:5	S:0	S:0	S:2	S:3	S:4	S:5	S:0
$Action : V_2$	D:$\frac{10}{3}$	D:3	S:3	S:4	S:5	S:0	S:0	S:2	S:3	S:4	S:5	S:0

Table 1: Policy iteration applied to micro blackjack game.

b Pursuit Evasion [2] (≈ 20 min)

Pacman is trapped in the following 2 by 2 maze with a hungry ghost (the horror)! 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 receives utility -100. Otherwise, he receives a utility of 1. The ghost attempts to minimize the utility that Pacman receives. Assume the ghost makes the first move.
For example, with a discount factor of $\gamma = 1.0$, if the ghost moves down, then Pacman moves left, Pacman earns a reward of 1 after the ghost's move and -100 after his move for a total utility of -99.
Note that this game is not guaranteed to terminate.

Figure 1: Pacman vs Ghost

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 truncated game after 2 ghost moves and 2 Pacman moves? (Hint: you should not need to build the minimax tree)
   If Pacman plays optimally he will avoid the ghost and so he will exactly do the opposite move of this latter. Then the score of pacman will be equal to: $\sum_{i=0}^{3} 1\gamma^i = 1 + 0.5 + 0.5^2 + 0.5^3 = 1.875$.

2. Assume a discount factor $\gamma = 0.5$. What is the minimax value of the complete (infinite) game? (Hint: you should not need to build the minimax tree)
   $\sum_{i=0}^{\infty} 1\gamma^i = \frac{1}{1-\gamma} = 2$

3. Why is value iteration superior to minimax for solving this game ?
   Because value iteration converges to a value whereas the size of the minimax tree is infinite. Value iteration gives a policy whereas minimax only provides the next action.

4. 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 position of Pacman and the ghost), and let $A_P(s)$ be the space of actions available to Pacman in state s (and similarly let $A_G(s)$ be the space of actions available to the ghost). Let $N(s, a) = s'$ denote the successor function (given a starting state s, this function returns the state s' which results after taking action a. Finally, let $R(s)$ denote the utility received after moving to state s. Write down an expression for $P^\star(s)$, the value of the game to Pacman as a function of the current state s (analogous to the Bellman equations). Use a discount factor of $\gamma = 1.0$. Hint: your answer should include $P^\star(s)$ on the right hand side.
   $P^\star(s) = \max_{a \in A_p(s)} [R(N(s,a)) + \min_{a' \in A_G(N(s,a))} R(N(N(s,a),a')) + P^\star(N(N(s,a),a'))]$

c Crying Baby Problem (≈ 20 min)

To earn money you've decided to do babysitting. Tonight you'll keep a 9 months old child, his parents asked you to take care to food him when he is hungry. Because you're 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 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 give you a smaller salary). You will have many things to do tonight and so you decide to optimize your time by just checking the baby cries every 10 minutes. Between two intervals of time you assume that the baby has a probability of 0.2 to become hungry. You can also suppose that the baby has a uniform probability of being hungry when you start your babysitting.

[2] source: `http://ai.berkeley.edu/sections/section_4_9vzQ4e7e6xqagFuQDinu0lmeSEI29Z.pdf`

1. Draw the dynamic bayesian network (as well as the conditional probability tables) associated with your babysitting of tonight.
   Let $F_t \in \{\text{True, False}\}$ denote the fact that you food the baby or not at time t, $H_t \in \{\text{True, False}\}$ the unknown fact that the baby is hungry or not at time t, $C_t \in \{\text{True, False}\}$ the fact that the baby is crying or not at time t and $R_t \in \{0, -5, -10\}$. The dynamic Bayesian network is presented by Figure 2 and the conditional probabilities are given by Tables 2, 3 and 4. For completeness we also have to define $P(H_0 = \text{True}) = 0.5$.

F_t	True		False	
H_t	True	False	True	False
$P(H_{t+1} = \text{True} \mid H_t, F_t)$	0	0	1	0.2

Table 2: CPT of H_{t+1}

H_t	True	False
$P(C_t = \text{True} \mid H_t)$	0.8	0.1

Table 3: CPT of C_t

F_t	True		False	
H_t	True	False	True	False
R_t	-5	-5	-10	0

Table 4: Degenerated CPT of R_t

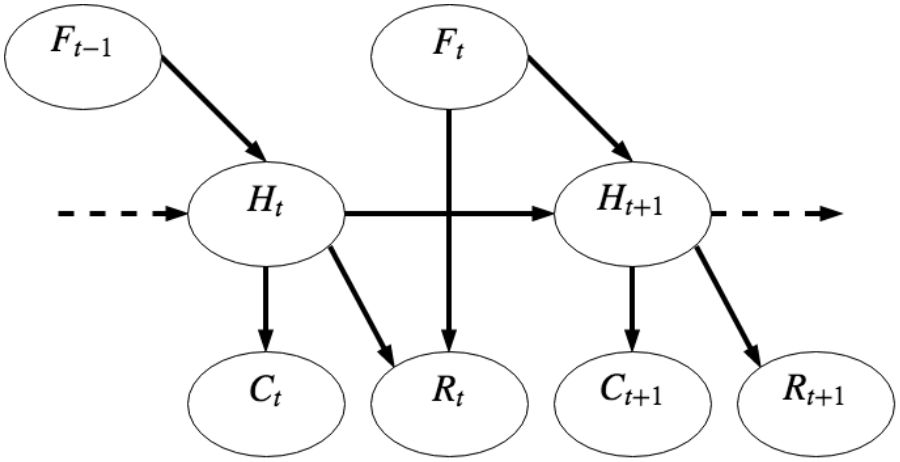

Figure 2: Dynamic Bayesian network of the crying baby problem.

2. Give the sequence of belief states if you assume a uniform prior and that the sequence of (action, obser-

vation) is: (No feed, cry), (Feed, no cry), (No feed, No cry), (No feed, No cry).

$$b_{H_0} = P(H_0 = \text{True}) \tag{5}$$
$$= 0.5 \tag{6}$$
$$b_{H_1} = P(H_1|F_0 = \text{False}, C_1 = \text{True}) \tag{7}$$
$$= \sum_{H_0} P(H_0, H_1|F_0 = \text{False}, C_1 = \text{True}) \tag{8}$$
$$= \alpha \sum_{H_0} P(H_0, H_1, F_0 = \text{False}, C_1 = \text{True}) \tag{9}$$
$$= \alpha \sum_{H_0} P(H_0)P(H_1|H_0, F_0 = \text{False})P(C_1 = \text{True}|H_1) \tag{10}$$
$$= \frac{\alpha}{2} P(C_1 = \text{True}|H_1) \sum_{H_0} P(H_1|H_0, F_0 = \text{False}) \tag{11}$$
$$= \begin{cases} P(H_1 = \text{True}|F_0 = \text{False}, C_1 = \text{True}) & = \frac{\alpha}{2} P(C_1 = \text{True}|H_1 = \text{True}) \\ & \sum_{H_0} P(H_1 = \text{True}|H_0, F_0 = \text{False}) \\ & = \frac{\alpha}{2} 0.8(1 + 0.2) = 0.48\alpha \\ P(H_1 = \text{False}|F_0 = \text{False}, C_1 = \text{True}) & = \frac{\alpha}{2} P(C_1 = \text{True}|H_1 = \text{False}) \\ & \sum_{H_0} P(H_1 = \text{False}|H_0, F_0 = \text{False}) \\ & = \frac{\alpha}{2} 0.1(0 + 0.8) = 0.04 \end{cases} \tag{12}$$
$$= \begin{cases} 0.923 \\ 0.077 \end{cases} \tag{13}$$
$$b_{H_2} = \begin{cases} 0 \\ 1 \end{cases} \tag{14}$$
$$b_{H_3} = \sum_{H_2} P(H_2, H_3|F_2 = \text{False}, C_3 = \text{False}, b_{H_3}) \tag{15}$$
$$= \alpha \sum_{H_2} P(H_2)P(H_3|H_2, F_2 = \text{False})P(C_3 = \text{False}|H_3) \tag{16}$$
$$= \alpha P(C_3 = \text{False}|H_3) \sum_{H_2} P(H_2)P(H_3|H_2, F_2 = \text{False}) \tag{17}$$
$$= \begin{cases} \alpha \times 0.2(0 \times 1 + 1 \times 0.2) \\ \alpha \times 0.9(0 \times 0 + 1 \times 0.8) \end{cases} \tag{18}$$
$$= \begin{cases} 0.052 \\ 0.948 \end{cases} \tag{19}$$
$$b_{H_4} = \begin{cases} \alpha \times 0.2(0.052 \times 1 + 0.948 \times 0.2) \\ \alpha \times 0.9(0.052 \times 0 + 0.948 \times 0.8) \end{cases} \tag{20}$$
$$= \begin{cases} 0.067 \\ 0.933 \end{cases} \tag{21}$$

1.3 Supplementary material