8 Markov Decision Processes

Exercise 8.1

Tell if the following statements about MDPs are true or false. Motivate your answers.

1. To solve an MDP it is enough to consider the reward function for each state-action pair;
2. An action you take on an MDP might influence the future rewards you gained;
3. Problems in which an agent knows the state of the environment and the MDP do not require the use of RL;
4. Policies applied to an environment are influenced by other learning processes ongoing on the considered environment.

Exercise 8.2

State if the following applications may be modeled by means of an MDP:

1. Robotic navigation in a grid world;
2. Stock Investment;
3. Robotic soccer;
4. Playing Carcassonne (board game).

Define the possible actions and states of each MDP you considered.

Exercise 8.3

Consider the following modeling of a classification problem as sequential decision making problem:

$$
\begin{aligned}
o_i &\leftarrow x_i \\
a_i &\leftarrow \hat{y}_i \\
r_i &\leftarrow 1 - |t_i - \hat{y}_i|
\end{aligned}
$$

Does this correspondence makes sense? Comment adequately your answer.

Exercise 8.4

For each one of the following dichotomies in MDP modeling provide examples of problems with the listed characteristics:

1. Finite/infinite actions;
2. Deterministic/stochastic transitions;
3. Deterministic/stochastic rewards;
4. Finite/indefinite/infinite horizon.

Exercise 8.5

Are the following statements about the discount factor γ in a MDP correct?

- A myopic learner corresponds to have low γ values in the definition of the MDP;
- In an infinite horizon MDP we should avoid using $\gamma = 1$, while it is reasonable if the horizon is finite;
- γ is an hyper-parameter for the policy learning algorithm;
- The probability that an MDP will be played in the next round is γ.

Provide adequate motivations for your answers.

Exercise 8.6

Comment the following statements about solving MDPs. Motivate your answers.

1. In a finite state MDP we may look for Markovian, stationary and deterministic optimal policies;
2. For finite horizon MDPs we should consider non-stationary optimal policies;
3. The results of coupling a specific policy and an MDP is a Markov process;
4. Given a policy we can compute P^π and R^π on an MDP;
5. The value function $V^*(s)$ contains all the information to execute the optimal policy π^* on a given MDP;
6. The action-value function $Q^*(s, a)$ contains all the information to execute the op-

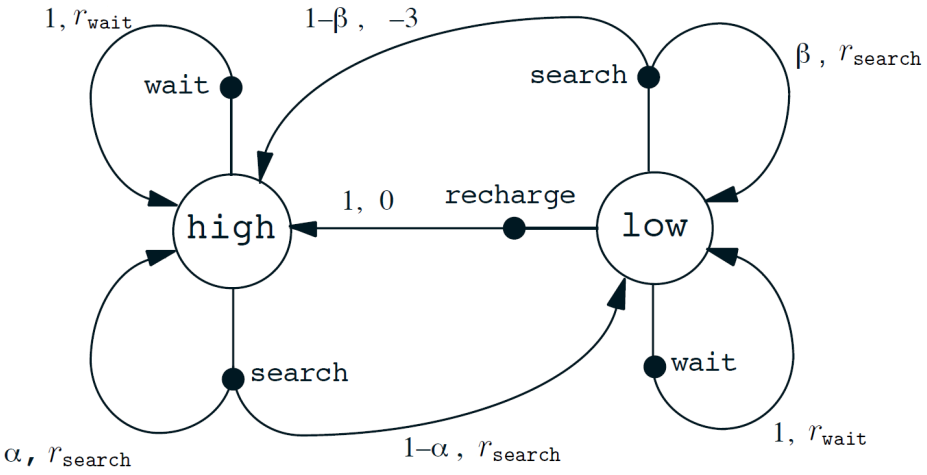


Figure 8.1: The MDP for the cleaning robot problem.

timal policy π^* on a given MDP;

7. There is a unique optimal policy in an MDP;
8. There is a unique optimal value function in an MDP.

Exercise 8.7

Consider the MDP in Figure 8.1 with $\alpha = 0.3$, $\beta = 0.5$, $\gamma = 1$, $r_{search} = 2$, $r_{wait} = 0$ and the following policy:

$$\pi(s|H) = 1,$$
$$\pi(s|L) = 0.5,$$
$$\pi(r|L) = 0.5.$$

1. Compute the Value function where the MDP stops after two steps.
2. Compute the Values function in the same setting assuming a discount factor of $\gamma = 0.5$.
3. Compute the action-value function for each action value pair in the case the MDP stops after a single step.

Exercise 8.8

Provide the formulation of the Bellman expectation for V equations for the MDP in Figure 8.1, with $\alpha = 0.2$, $\beta = 0.1$, $r_{search} = 2$, $r_{wait} = 0$, $\gamma = 0.9$ and in the case we

consider the policy:

$$\pi(H|s) = 1,$$
$$\pi(L|r) = 1.$$

Exercise 8.9

Tell if the following statements are TRUE or FALSE. Motivate your answers.

1. We are assured to converge to a fixed point solution when we apply repeatedly the Bellman expectation operator;
2. We are assured to converge to a fixed point solution when we apply repeatedly the Bellman optimality operator;
3. The Bellman solution to Bellman expectation equation is always a good choice to compute the value function for an MDP;
4. The solution to the Bellman expectation equation provided by the iterative use of the Bellman expectation operator is always less expensive than computing the exact solution using the Bellman expectation equation;
5. The application of the Bellman optimality operator 10 times applied to a generic value function V_0 guarantees that $||V^* - T^{10}V_0||_\infty \leq \gamma^{10}||V^* - V_0||_\infty$

Exercise 8.10

Which one would you chose between the use of the Bellman iterative equation vs. Bellman exact solution in the case we are considering the following problems:

1. Chess
2. Cleaning robot problem in Figure 8.1
3. Maze escape
4. Tic-tac-toe

Provide adequate motivations for your answers.

Exercise 8.11

Consider the MDP in Figure 8.2:

1. Provide the transition matrix for the policy $\pi(I|s_1) = 1, \pi(M|s_2) = 1, \pi(M|s_3) = 1$;

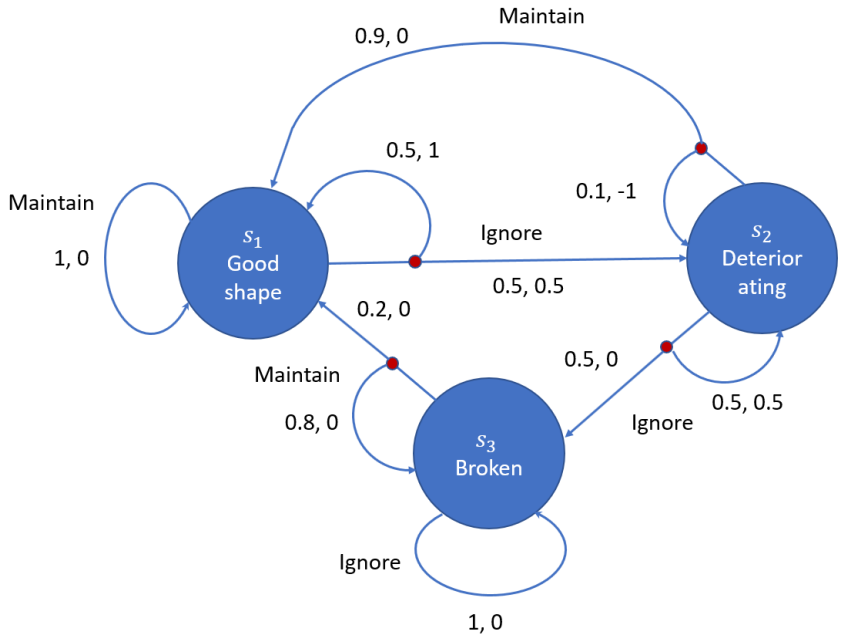


Figure 8.2: The MDP for machinery maintenance problem.

2. Provide the expected instantaneous reward for the previous policy;
3. Compute the value function for the previous policy in the case the MDP stops after two steps;
4. Compute the action-value function for each state-action pair in the case the MDP stops after a single step.

Exercise 8.12

Consider the following snippet of code:

```
1 V1 = np.linalg.inv(np.eye(nS) - gamma * pi @ P_sas) @ (pi @ R_sa)
2
3 V_old = np.zeros(nS)
4 tol = 0.0001
5 V2 = pi @ R_sa
6 while np.any(np.abs(V_old - V2) > tol):
7   V_old = V2
8   V2 = pi @ (R_sa - gamma * P_sas @ V)
```

1. Describe the purpose of the procedure of line `1` and the purpose of the procedure of lines `3-8`. Are they correct? If not, propose a correction.
2. What is the main disadvantage of the procedure of line `1` compared to the one of lines `3-8`?

3. What happens to the two procedures when `gamma`$= 1$?

Exercise 8.13

We are given an Heating, Ventilation, and Air Conditioning (HVAC) system in which the states are cold (c), medium (m), warm (w) temperature. We can perform three actions: heat (h), refrigerate (r), and do nothing (d). Assume to have the following partial episodes for the HVAC functioning.

$$(c, d, 0) \to (c, h, 1) \to (m, h, 1) \to (m, h, -1) \to (w, r, 1) \to (m, \cdot, \cdot) \to \dots$$
$$(m, r, -2) \to (c, h, -2) \to (c, h, 1) \to (m, h, 1) \to (m, h, 1) \to (w, \cdot, \cdot) \to \dots$$

where a tuple (S, A, R) correspond to the State, Action, and Reward at a specific time.

1. Model it as an MDP and draw the corresponding graphical representation, specifying the transition probabilities and rewards (estimated from the episodes) for each transition.

2. Can you tell if the reward of this process is stochastic or deterministic? And what about the transitions?

3. Assuming we want to evaluate the performance of the HVAC, tell which kind of problem we are in and suggest a technique to solve it.

Exercise 8.14

Tell whether the following statements about MDP and DP are true or false. Motivate your answers.

1. After the policy improvement step, the policy always changes.

2. Some sequential decision problems cannot be modeled as an MDP.

3. It is always better to perform Policy Evaluation using the iterative application of the Bellman expectation operator instead of solving a linear system.

4. There are some tasks in which we may not need to discount rewards.

5. The best approach to solve exactly a large MDP, exploiting the knowledge of the one-step dynamics, is Dynamic Programming.

6. The only optimal policy of an MDP can be non-Markovian.

7. We can solve Bellman Optimality Equation with a linear system.

8. With the optimal value function, I can compute the optimal policy, even without knowing one-step dynamics.

9. In Policy Evaluation, we can converge to the correct value function, only if we properly initialize its values.

10. In an MDP, given the current state and action, the past actions you performed might influence the future rewards you will gain.

11. Given a value function, there is only one policy that corresponds to it.

12. Value Iteration may not converge to the optimal value function.

Exercise 8.15

Tell if the following statements about solving MDPs are true or false. Motivate your answers.

1. In a finite state MDP there only exist optimal policies which are Markovian, stationary and deterministic;

2. If we observe a finite episode on an MDP, we should consider the undiscounted ($\gamma = 1$) cumulative reward as performance metric;

3. The Bellman expectation operator can be used to compute the optimal policy of an MDP;

4. There is a unique optimal policy in an MDP (if true provide a theoretical result, if false a counterexample).

Exercise 8.16

Tell if the following statements are TRUE or FALSE. Motivate your answers.

1. Starting from any value function, we are not assured to converge to a solution when we apply repeatedly the Bellman optimality operator;

2. The closed form solution to the Bellman equation is always a good choice to compute the value function of an MDP;

3. The application of the Bellman optimality operator T for 5 times to a generic value function V_0 guarantees that $||V^* - T^5 V_0||_\infty \leq \gamma^3 ||V^* - V_0||_\infty$;

4. Starting from any value function, we are assured to converge to a solution when we apply repeatedly the Bellman expectation operator.

Exercise 8.17

Provide an MDP modeling, specifying all its defining elements, of the following processes:

1. Car manifacturing;
2. Chess.

Exercise 8.18

Consider an MDP with three states $\{s_1, s_2, s_3\}$ and actions $\{d, so, em\}$. We have the following optimal action-value function $Q^*(s, a)$ for the three states and the three actions when we consider different discount factors γ:

	$\gamma = 0.9$			$\gamma = 0.95$			$\gamma = 0.99$		
	d	so	cm	d	so	cm	d	so	cm
s_1	35	25	0	95	90	0	780	785	0
s_2	55	0	45	120	0	125	810	0	825
s_3	165	0	0	240	0	0	940	0	0

1. Provide the optimal policy π^* for each discount factor γ.
2. What is the expected reward for π^* if the initial state distribution is $(0.4, 0.4, 0.2)$.
3. Which γ would you choose for this specific problem?

Exercise 8.19

Consider the MDP in Figure 8.1 with transition probabilities $\alpha = 0$ and $\beta = 0.5$, discount factor $\gamma = 1$, instantaneous rewards $r_{search} = 4$ and $r_{wait} = 1$ and the following policy:

$$\pi(high, search) = 0.5,$$
$$\pi(high, wait) = 0.5,$$
$$\pi(low, search) = 1.$$

1. Compute the Value function in the case the MDP stops after two steps.
2. Compute the action-value function for each action value pair in the case the MDP stops after a single step.

Solutions

Answer of exercise 8.1

1. FALSE The dynamic of the MDP can be inferred only if we relate the state/action pairs with the next states.
2. TRUE The action might determine the sequence of states you will visit in the future, thus the reward you will collect.
3. FALSE RL could be used for computational reasons, e.g., if the number of states is too large.
4. TRUE In the case a learner, other than the agent, is operating in the considered environment we could have a non-stationary environment as a result.

Answer of exercise 8.2

An MDP is fully defined if you specify:

- $\mathcal{S}$ a set of states
- $\mathcal{A}$ a set of actions
- P a state transition probability matrix
- R a reward function, $R(s;a) = \mathbb{E}[r|s;a]$
- γ a discount factor,
- μ_i^0 a set of initial probabilities

Robotic navigation in a grid world:

- $\mathcal{S}$ each one of the locations of the grid
- $\mathcal{A}$ usually is go left, go right, go up and go down, no action in the goal state
- P a $\{0,1\}$ matrix where you have 1 if you reach s' starting from s with action a, and 0 otherwise
- R 0 if you are still away from the goal point and 1 if you reached the goal
- $\gamma = 1$
- μ_i^0 1 in the initial location, 0 otherwise

Stock Investment decision:

- $\mathcal{S}$ amount of money in the bank account, amount of stocks owned
- $\mathcal{A}$ amount of stocks to buy or sell at the next time instant
- P deterministic from a state to another
- R difference in the portfolio value between two time instants
- $\gamma < 1$
- μ_i^0 1 in the state corresponding to the amount of money one has in the bank account and the number of stocks she owns, 0 in all other states

Robotic soccer (let us consider the opponent team as a stochastic event):

- $\mathcal{S}$ position of each one of the robots of our team on the field, position of the ball
- $\mathcal{A}$ movement of each one of the robots of the team in the field
- P stochastic transition given from the not deterministic behaviour of the ball and the actions of the environment (opponent soccer robots)
- R 1 if a goal has been scored, 0 otherwise
- $\gamma = 1$ since the game ends after a finite amount of time
- μ_i^0 1 in the state corresponding to the initial disposition of the robots on the field and the initial possession of the ball.

Complete rules to play Carcassonne can be found at: `http://riograndegames.com/getFile.php?id=670`. Playing Carcassonne (given the fixed and deterministic strategies of the other players):

- $\mathcal{S}$ all the possible disposition of the tiles on the table and of the meeples of each player on the tiles in a valid position and a score for each player
- $\mathcal{A}$ place a tile adjacent to a valid tile
- P deterministic state transition from a state to another
- R points scored in the turn or points scored in the final turn
- $\gamma = 1$ since the number of tiles is finite
- μ_i^0 1 on the initial tile on the board, 0 otherwise

Answer of exercise 8.3

The correspondence makes sense, since it is a specific case of an MDP having a single state. The use of traditional techniques to solve MDPs does not make sense, since they are unnecessarily complex for the problem they are trying to solve, i.e, there does not exist a temporal dependence over the predictions.

Answer of exercise 8.4

1.
   - Finite: robotic navigation (up, down, left and right)
   - Infinite actions: pole balancing with continuous space applied force
2.
   - Deterministic transitions: chess (given a fixed and deterministic opponent strategy)
   - Stochastic transitions: blackjack (playing against the dealer)
3.
   - Deterministic rewards: robotic navigation (0 everywhere, 1 in the exit point)
   - Stochastic rewards: ad banner allocation (depending on clicks)
4.
   - Finite horizon: Carcassonne (finite number of tiles)
   - Indefinite horizon: robotic navigation
   - Infinite horizon: stock exchange

Answer of exercise 8.5

- TRUE Low values of gamma means that we are not considering valuable the revenues gained in the far future. Conversely, in the case we are far–sighted learners, we should use a high value for γ;
- TRUE If we consider $\gamma = 1$ we are not discounting rewards gained in the future, which may lead to produce infinite cumulative rewards. On the contrary, it is reasonable not to discount rewards if we know the horizon is finite;
- FALSE γ is a parameter of the MDP we are considering and it is conceptually related to the problem we are facing and not to the learner. Different values for γ may correspond to different optimal policies for the MDP;
- TRUE By considering a specific $\gamma < 1$ we are somehow stating the fact that the process might end with a given probability at the next time step.

Answer of exercise 8.6

1. TRUE We have theoretical guarantee that at least one Markovian, deterministic and stationary optimal policy exists. This does not imply that it is only one or that all the optimal policies satisfies the aforementioned properties;

2. TRUE The fact that we know that the process will end implies that the optimal action might be influenced also by the number of rounds remaining;

3. TRUE Once you fix the policy there is no need of deciding anything else and the succession of the states becomes a Markovian process;

4. TRUE Once we fix the policy, we can compute the transition P^π and the reward R^π in each state. This is needed if we want to solve the Bellman expectation equation;

5. FALSE The knowledge of the optimal value in each state $V^*(s)$ does not imply that we know the optimal action to perform in each state to get it;

6. TRUE The state-value function $Q^*(s,a)$ specifies the value obtained at each state for each action. Thus the optimal policy is just $\pi^*(s) = \arg\max_a Q^*(s,a)$;

7. FALSE There might be multiple policies getting the same value in each state;

8. TRUE V^* is the unique fixed point for the Bellman optimality equation.

Answer of exercise 8.7

1. Since $V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(s|a)\left(R(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|a,s)V^\pi(s)\right)$ thus:

$$\begin{aligned}
V^\pi(H) &= 1[0.3(2 + V^\pi(H)) + 0.7(2 + V^\pi(L))] \\
&= 0.6 + 0.3(0.3 \cdot 2 + 0.7 \cdot 2) + 1.4 + 0.7[0.5 \cdot 0 + 0.5(0.5 \cdot (-3) + 0.5 \cdot 2)] \\
&= 0.6 + 0.6 + 1.4 - 0.175 = 2.425 \\
V^\pi(L) &= 0.5(0 + V^\pi(H)) + 0.5[0.5(-3 + V^\pi(H)) + 0.5(2 + V^\pi(L))] \\
&= 0 + 0.5(0.3 \cdot 2 + 0.7 \cdot 2) - 0.25 + 0.25(0.3 \cdot 2 + 0.7 \cdot 2) + \\
&+ 0.25[0.5 \cdot 0 + 0.5(0.5 \cdot (-3) + 0.5 \cdot 2)] \\
&= 0 + 1 - 0.25 + 0.5 - 0.0625 = 1.1875
\end{aligned}$$

2. In the case we have a discount factor $\gamma = 0.5$ we have:

$$\begin{aligned}
V^\pi(H) &= 1[0.3 \cdot 2 + 0.5 \cdot 0.3V^\pi(H)) + 0.7 \cdot 2 + 0.5 \cdot 0.7V^\pi(L)] \\
&= 0.6 + 0.15[0.3 \cdot 2 + 0.7 \cdot 2] + 1.4 + 0.35[0.5 \cdot 0 + 0.5(0.5 \cdot (-3) + 0.5 \cdot 2)]; \\
&= 0.6 + 0.3 + 1.4 - 0.0875 = 2.2125 \\
V^\pi(L) &= 0.5(0 + 0.5V^\pi(H)) + 0.5[0.5(-3 + 0.5V^\pi(H)) + 0.5(2 + 0.5V^\pi(L))] \\
&= 0 + 0.5 \cdot 0.5(0.3 \cdot 2 + 0.7 \cdot 2) - 0.75 + 0.125(0.3 \cdot 2 + 0.7 \cdot 2) + \\
&+ 0.5 + 0.25[0.5 \cdot 0 + 0.5(0.5 \cdot (-3) + 0.5 \cdot 2)] \\
&= 0.5 - 0.75 + 0.25 + 0.5 - 0.0625 = 0.4375
\end{aligned}$$

   Clearly the discounted value for the states is lower than the not discounted one, in the case we consider an MDP with finite time horizon.

3. The action-value function in our case correspond to the values of the expected instantaneous reward, thus:

$$\begin{aligned}
&Q(L, w) = 1 \cdot 0 = 0 \\
&Q(L, s) = 0.5 \cdot 2 + 0.5 \cdot (-3) = -0.5 \\
&Q(L, r) = 1 \cdot 0 = 0 \\
&Q(H, w) = 1 \cdot 0 = 0 \\
&Q(H, s) = 0.3 \cdot 2 + 0.7 \cdot 2 = 2
\end{aligned}$$

Answer of exercise 8.8

$$\begin{aligned}
&V(H) = 2 + 0.9[0.2V(H) + 0.8V(L)], \\
&V(L) = 0 + 0.9V(H),
\end{aligned}$$

or in matrix form:

$$V = \begin{bmatrix} 2 \\ 0 \end{bmatrix} + 0.9 \begin{bmatrix} 0.2 & 0.8 \\ 1 & 0 \end{bmatrix} V$$

Answer of exercise 8.9

1. TRUE Since it is possible to show that it is a contraction operator, it is assured to have a single optimal point;

2. TRUE As its expectation counterpart it is a contraction, thus converges to an optimal point;

3. FALSE In the case we have computational constraints (or equivalently a huge state space) we might resort to the recursive solution;

4. FALSE If we want to find the exact solution we might need more computational power than solving the linear system corresponding to the Bellman expectation equation;

5. TRUE Since the fact that the operator is a contraction will guarantee to shrink the distance from the optimal solution at each step of a factor γ.

Answer of exercise 8.10

1. ITERATIVE The state space is too large

2. EXACT The state space is small enough that the exact solution is feasible

3. NONE We do not have information about the state we are in

4. EXACT Again we have a small enough state space

Answer of exercise 8.11

1.
$$P = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.9 & 0.1 & 0 \\ 0.2 & 0 & 0.8 \end{bmatrix}$$

2.
$$R = \begin{bmatrix} 0.75 \\ -0.1 \\ 0 \end{bmatrix}$$

3.
$$\begin{aligned} V(s_1) &= R(s_1) + 0.5V(s_1) + 0.5V(s_2) \\ &= R(s_1) + 0.5R(s_1) + 0.5R(s_2) = 0.75 - 0.1 + 0.75 = 1.4 \\ V(s_2) &= R(s_2) + 0.9V(s_1) + 0.1V(s_2) \\ &= R(s_2) + 0.9R(s_1) + 0.1R(s_2) = -0.1 + 0.675 - 0.01 = 0.664 \\ V(s_3) &= R(s_3) + 0.8V(s_1) + 0.2V(s_3) \\ &= R(s_3) + 0.8R(s_1) + 0.2R(s_3) = 0 + 0.15 + 0 = 0.15 \end{aligned}$$

4.

$$
\begin{aligned}
&Q(M, s_1) = 0 \\
&Q(I, s_1) = 0.5 \cdot 0.5 + 1 \cdot 0.5 = 0.75 \\
&Q(M, s_2) = -1 \cdot 0.1 + 0 \cdot 0.9 = -0.1 \\
&Q(I, s_2) = 0 \cdot 0.5 + 0.5 \cdot 0.5 = 0.25 \\
&Q(M, s_3) = 0 \cdot 0.8 + 0 \cdot 0.2 = 0 \\
&Q(I, s_3) = 0
\end{aligned}
$$

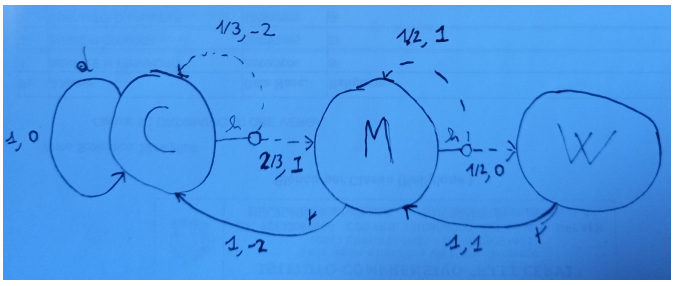
Answer of exercise 8.12

1. The procedure of line `1` computes the closed-form solution of the state value function V^{π} of policy `pi` in the MDP with transition model `P_sas`, reward `R_sa` and discount factor `gamma`. The procedure of lines `3-8` performs the iterative application of the Bellman expectation operator to compute the same value function V^{π}. The iterative procedure is stopped when a given threshold `tol` between consecutive approximation is reached. However, line `8` does contain a mistake and should be corrected as follows:
   `V2 = pi @ (R_sa + gamma * P_sas @ V).`

2. The main disadvantage of the procedure of line `1` compared to the one of lines `3-8` is a computational one, i.e., the computation of the closed-form solution might be infeasible when the number of states/actions is large.

3. When `gamma`$= 1$ the procedure of line `1` might lead to a singular matrix (attempting to invert it), whereas the procedure of lines `3-8` might never reach the requested tolerance `tol`.

Answer of exercise 8.13

1. 

2. The transition are stochastic since some of the actions are leading to two different new states, e.g., the action heat (h) in the state cold (c). The reward is stochastic as well, since heating in the medium state provided once the reward 1 and once -1.

3. This setting suggests it is an MDP prediction problem, either using directly the original episodes (using MC or TD) or one might use the estimated model and use DP techniques to solve it in an exact way, due to the limited dimension of the problem.

Answer of exercise 8.14

1. FALSE, if the algorithm finds the optimal policy, then it remains the same after the policy improvement step.

2. TRUE, to model a problem as an MDP, we have to assume that the environment state is fully observable and Markovian, hence the current state of the environment should be completely determined by the current observation made by the agent. In some cases, we can augment the state of the environment to make it Markovian.

3. FALSE, when the state space is small we can use the closed form solution. However, even if the linear system approach offers the exact solution, for very large problems it can be impractical to solve it, hence, we can resort to using the approximated solution offered by the iterative application of the Bellman expectation operator.

4. TRUE, for example in an episodic task, we may set the discount factor equal to 1.

5. TRUE, an MDP can be solved with a brute force approach, using Dynamic Programming (DP) approaches (as Policy Iteration or Value iteration), or with Linear Programming (LP). Since the number of states is often very large (e.g., often growing exponentially with the number of state variables) it is impractical to use brute force (which scales exponentially in the number of states) or LP, which typically does not scale well on large problems. On the other hand, DP approaches have a polynomial dependence on the number of states and actions, hence, they can scale better on larger instances. Instead approximated solutions can be derived by using Reinforcement Learning.

6. FALSE, we are guaranteed that a stationary deterministic Markovian optimal policy exists, but it might also exist a non-Markovian optimal policy.

7. FALSE, since the equation involves the computation of a maximum, which is not linear.

8. FALSE, the optimal policy is the greedy policy w.r.t. the optimal Q-function. If I have the optimal value function, but not the one-step dynamics, I cannot recover the optimal Q-function.

9. FALSE, it can be proved that the convergence is assured for any initialization of

the value function (the Bellman Expectation Operator is a contraction).

10. FALSE, past actions influence past rewards and the dynamics of the process that brought you to a certain state. However, rewards depend only on the current (Markovian) state and action, not on past actions.

11. FALSE, a policy is mapped on a single value function, but the opposite is not true: different policies can obtain the same values in each state, but in different ways.

12. FALSE, Value Iteration always converges to the optimal in the limit on infinite iterations.

Answer of exercise 8.15

1. FALSE: we are assured that Markovian, stationary and deterministic optimal policy, but there might be others which are still optimal but do not have the aforementioned properties;

2. FALSE: it might be an indefinite horizon problem, in which some of the episodes terminates and some other not. In this case, using the undiscounted cumulative reward as a performance metric on the former kind of episodes might lead to infinite values;

3. TRUE: for instance it is used in Policy iteration coupled with a greedy update to find the optimal policy. Another possible solution, when the policy space is limited, is to use them to solve the MDP by a brute force approach;

4. FALSE: for instance if in a state two different actions a_1 and a_2 provide the same reward, has the same transition probabilities, and one of them a_1 is included in the optimal policy, then substituting a_1 with a_2 in the optimal policy would provide an optimal policy too.

Answer of exercise 8.16

1. FALSE: the Bellman equation has a single fixed point and its iterative formulation is a contraction, thus no matter where we start we are assured to converge if we apply the operator enough times;

2. FALSE: if the MDP is too large it requires to invert a large matrix, which could be not feasible;

3. TRUE: we are assured that $||V^* - T^5 V_0||_\infty \leq \gamma^5 ||V^* - V_0||_\infty \leq \gamma^3 ||V^* - V_0||_\infty$ since $\gamma \leq 1$;

4. TRUE: the same reasoning for the optimality operator holds for the expectation operator.

Answer of exercise 8.17

To fully describe an MDP we need to define each element in the tuple $\langle S, A, R, P, \gamma, \mu_0 \rangle$.

1.
   - S: a vector describing the state of the car and the construction phase it is in;
   - A: the possible processes we might apply to finish the car building process (paint the external part, mount parts);
   - R: 1 if it is completed, 0 otherwise;
   - P: stochastic since some of the process might not succeed;
   - $\gamma = 1$: since the process has a finite time horizon;
   - μ_0: initial building state of the car.
2. Assuming the opponent is fixed and is playing a deterministic policy we are in an MDP, otherwise not
   - S: a vector describing the state of chess board at each time;
   - A: the possible moves for the player;
   - R: 1 for winning the game, 0 otherwise;
   - P: deterministic since a specific move leads you to a given board state;
   - $\gamma < 1$: the game might go on for an indefinite amount of time;
   - μ_0: the starting board state and the starting turn of the white player.

Answer of exercise 8.18

1. Since we have the optimal q-values of the MDP, the greedy policy w.r.t. it is optimal. Therefore:

$$\pi^*(0.9) = (d,\ d,\ d),$$
$$\pi^*(0.95) = (d,\ cm,\ d),$$
$$\pi^*(0.99) = (so,\ cm,\ d).$$

2. The expected reward $R(\gamma)$ given an initial distribution $\rho = (0.4, 0.4, 0.2)$ is given

by $\rho^T V^*(\gamma)$, therefore:

$$R(0.9) = (0.4, 0.4, 0.2)^T (35,\ 55,\ 165) = 14 + 22 + 33 = 69,$$
$$R(0.95) = (0.4, 0.4, 0.2)^T (95,\ 125,\ 240) = 38 + 50 + 48 = 136,$$
$$R(0.99) = (0.4, 0.4, 0.2)^T (785,\ 825,\ 940) = 314 + 330 + 188 = 832.$$

3. The third question does not make sense, since the discount factor γ is not a parameter that should be chosen by the learner, but a characteristic of the MDP. The use of different values of γ depends on the fact that the problem requires to be more far-sighted or myopic.

Answer of exercise 8.19

1.

$$\begin{aligned} V^\pi(s) &= \sum_a \pi(s|a) \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s') \right] \\ &= \sum_a \pi(s|a) \left[R(s,a) + \sum_{s'} P(s'|s,a) \sum_{a'} R(s',a') \right] \end{aligned}$$

$$\begin{aligned} V(high) &= 0.5 \left[4 + 1 \left(0 \sum_{a'} R(high, a') + 1 \sum_{a'} R(low, a') \right) \right] + 0.5 \left(1 + \sum_{a'} R(high, a') \right) \\ &= 0.5 \left(4 + 0.5 \cdot (-3) + 0.5 \cdot 4\right) + 0.5(1 + 0.5 \cdot 1 + 0.5 \cdot 4) = 0.5 \cdot 4.5 + 0.5 \cdot 3.5 = 4 \end{aligned}$$

$$\begin{aligned} V(low) &= 1 \left[0.5 + 1 \left(0.5 \sum_{a'} R(high, a') + 0.5 \sum_{a'} R(low, a') \right) \right] \\ &= 0.5 + 0.5 \cdot 2.5 + 0.5 \cdot 0.5 = 2 \end{aligned}$$

2.

$$\begin{aligned} &Q(high, search) = 0.5 \cdot 4 + 0.5 \cdot 4 = 4 \\ &Q(high, wait) = 1 \\ &Q(low, search) = 0.5 \cdot (-3) + 0.5 \cdot 4 = 0.5 \\ &Q(low, wait) = 1 \\ &Q(low, recharge) = 0 \end{aligned}$$