Decision making, Markov decision processes

Solved tasks
Collected by: Jiří Kléma, klema@fel.cvut.cz

Spring 2017

The main goal:

The text presents solved tasks to support labs in the A4B33ZUI course.

1 Simple decisions, Bayesian decision making

Example 1. *(AIMA, 16.10): A used car buyer can decide to carry out various tests with various costs (e.g., kick the tires, take the car to a qualified mechanic) and then, depending on the outcome of the tests, decide which car to buy. We will assume that the buyer is deciding whether to buy car a_1 and that there is time to carry out at most one test t_1 which costs 1,000 Kč and which can help to figure out the quality of the car. A car can be in good shape (s_+) or in bad shape (s_-), and the test might help to indicate what shape the car is in. There are only two outcomes for the test: pass (t_{1+}) or fail (t_{1-}). Car a_1 costs 30,000 Kč, and its market value is 40,000 Kč if it is in good shape; if not, 14000 Kč in repairs will be needed to make it in good shape. The buyers estimate is that a_1 has 70% chance of being in good shape. The test is uncertain: $Pr(t_{1+}(a_1)|a_{1+}) = 0.8$ a $Pr(t_{1+}(a_1)|a_{1-}) = 0.35$.*

Calculate the expected net gain from buying car a_1, given no test.

$EU(buy+|\{\}) = \sum_{s\in\{+,-\}} U(s)Pr(s|buy+) = 40,000 - (0.7 \times 30,000 + 0.3 \times 44,000) = 40,000 - 34,200 = 5,800$ Kč

An analogy in classic decision making:
$d^*(t) = \underset{buy+,buy-}{\operatorname{argmin}} \sum_{s\in\{+,-\}} l(d,s)Pr(s|t) = \underset{buy+,buy-}{\operatorname{argmin}} \sum_{s\in\{+,-\}} l(d,s)Pr(s) =$
$= \underset{buy+,buy-}{\operatorname{argmax}} (10000 \times 0.7 - 4000 \times 0.3, 0) = \operatorname{argmax}(5800, 0) = buy+$

Conclusion 1: It pays-off to buy the car without a test.

Use Bayes' theorem to calculate the probability that the car will pass or fail its test and hence the probability that it is in good or bad shape.

$Pr(a_{1+}|t_{1+}(a1)) = \frac{Pr(t_{1+}(a1)|a1+)\times Pr(a_{1+})}{Pr(t_{1+}(a1))} = \frac{0.8\times 0.7}{0.8\times 0.7+0.35\times 0.3} = \frac{0.56}{0.665} = 0.842$
$Pr(a_{1+}|t_{1-}(a1)) = \frac{Pr(t_{1-}(a1)|a_{1+})\times Pr(a_{1+})}{Pr(t_{1-}(a1))} = \frac{0.2\times 0.7}{0.2\times 0.7+0.65\times 0.3} = \frac{0.14}{0.335} = 0.418$

Calculate the optimal decisions given either a pass or a fail, and their expected utilities.

$EU(\alpha_{t_1}|t_{1+}(a1)) = 40,000 - (0.842\times 30,000 + 0.158\times 44,000) = 40,000 - 32,240 = 7,788$ Kč
$EU(\alpha_{t_1}|t_{1-}(a1)) = 40,000 - (0.418\times 30,000 + 0.582\times 44,000) = 40,000 - 38,120 = 1,852$ Kč

An analogy in classic decision making:
$d^*(t_{1+}(a1)) = \underset{buy+,buy-}{\operatorname{argmin}} \sum l(d,s)Pr(s|t) = \underset{buy+,buy-}{\operatorname{argmin}} (10,000\times 0.842 - 4,000\times 0.158, 0) = \underset{buy+,buy-}{\operatorname{argmin}} (7788, 0) = buy+$
$d^*(t_{1-}(a1)) = \underset{buy+,buy-}{\operatorname{argmin}} \sum l(d,s)Pr(s|t) = \underset{buy+,buy-}{\operatorname{argmin}} (10,000\times 0.418 - 4,000\times 0.582, 0) = \underset{buy+,buy-}{\operatorname{argmin}} (1852, 0) = buy+$

Conclusion 2: It pays-off to buy the car without for both the test outcomes. This immediately suggests zero VPI of the test – the test has no potential to change buyer's decision.

Calculate the value of (perfect) information of the test. Should the buyer pay for t_1?

$EU(\alpha|\{\}) = \max(5800, 0) = 5800$ Kč
$EU(\alpha_{t_1}|t_{1+}(a1)) = \max(7788, 0) = 7788$ Kč
$EU(\alpha_{t_1}|t_{1-}(a1)) = \max(1852, 0) = 1852$ Kč

$VPI(t_1(a_1)) = (Pr(t_{1+}(a1))\times 7788 + Pr(t_{1-}(a1))\times 1852) - 5800 = (0.665\times 7788 + 0.335\times 1852) - 5800 = 5800 - 5800 = 0$ Kč
It is "hard" zero, can be confirmed as follows:
$Pr(t_{1+}(a1))\times(10000\times Pr(a_{1+}|t_{1+}(a1)) - 4000\times Pr(a_{1-}|t_{1+}(a1))) + Pr(t_{1-}(a1))\times(10000\times Pr(a_{1+}|t_{1-}(a1)) - 4000\times Pr(a_{1-}|t_{1-}(a1))) = 10000\times(Pr(a_{1+},t_{1+}(a1)) + Pr(a_{1+},t_{1-}(a1))) - 4000\times(Pr(a_{1-},t_{1+}(a1)) + Pr(a_{1-},t_{1-}(a1))) = 10000\times Pr(a_{1+}) - 4000\times Pr(a_{1-}) = 5800$ Kč
$VPI(t1(a1)) - Cost(t1(a1)) = -1000 < 0$

Conclusion 3: A logical resolution. The test cannot change decision, it has zero value, when considering its cost it brings negative outcome. The best strategy is to buy the car without the test. The test would need better sensitivity to pay-off. Accuracy of the test (note that the trivial "good state" classifier shows accuracy 0.7):
$Pr(t_{1+}(a1), a_{1+}) + Pr(t_{1-}(a1), a_{1-}) = 0.8 \times 0.7 + 0.65 \times 0.3 = 0.755$

Example 2. *You are going on a trip from San Francisco to Oakland. You have two options to get to Oakland, you want to get there as soon as possible. You can drive your car across the Bay Bridge or go by train through the tunnel under the bay. Bay Bridge is often jammed (on the given part of the day it is in about 40 % of cases). During normal operation, it takes 30 minutes drive. If there is traffic congestion, it takes 1 hour. The train journey always takes 40 minutes.*

When having no traffic information, does it pay off to drive or take a train?

$EU(train|\{\}) = 40$ min
$EU(car|\{\}) = \sum_{z \in \{+,-\}} U(z)Pr(z|car) = 0.4 \times 60 + 0.6 \times 30 = 42$ min

Conclusion 1: The train journey is faster.

Let us assume, that the traffic information for Bay Bridge is available on web, you can get it in 5 minutes. You know, that for congested bridge, the web page says the same with 90% probability. For normal traffic, the page indicates a traffic jam in 20% cases.

What is the congestion probability when having the traffic information?

We employ Bayes theorem (z ... congestion, real ... real situation, pred ... traffic information prediction):

$P(z_{real}|z_{pred}) = \frac{P(z_{pred}|z_{real})P(z_{real})}{P(z_{pred})} = \frac{0.9 \times 0.4}{P(z_{pred})} = \frac{0.36}{P(z_{pred})}$
$P(\neg z_{real}|z_{pred}) = \frac{P(z_{pred}|\neg z_{real})P(\neg z_{real})}{P(z_{pred})} = \frac{0.2 \times 0.6}{P(z_{pred})} = \frac{0.12}{P(z_{pred})}$
$P(z_{real}|z_{pred}) = \frac{0.36}{0.36+0.12} = 0.75$
$P(\neg z_{real}|z_{pred}) = \frac{0.12}{0.36+0.12} = 0.25$

Note: $P(z_{pred}) = 0.48$, and $P(\neg z_{pred}) = 0.52$.

$P(z_{real}|\neg z_{pred}) = \frac{P(\neg z_{pred}|z_{real})P(z_{real})}{P(\neg z_{pred})} = \frac{0.1 \times 0.4}{0.52} = 0.07$
$P(\neg z_{real}|\neg z_{pred}) = \frac{P(\neg z_{pred}|\neg z_{real})P(\neg z_{real})}{P(\neg z_{pred})} = \frac{0.8 \times 0.6}{0.52} = 0.93$

What should we do if the traffic information predicts normal operation / congestion?

$EU(car|z_{pred}) = P(z_{real}|z_{pred}) \times 60 + P(\neg z_{real}|z_{pred}) \times 30 = 0.75 \times 60 + 0.25 \times 30 =$ 52.5 min
$EU(car|\neg z_{pred}) = P(z_{real}|\neg z_{pred}) \times 60 + P(\neg z_{real}|\neg z_{pred}) \times 30 = 0.07 \times 60 + 0.93 \times 30 = 32$ min

Conclusion 2: If congestion is predicted, we will take train, otherwise we will go by car.

Is it efficient to spend 5 minutes by finding out the traffic information or is it better to simply set out?

If congestion is predicted, we will take train and vice versa. The train journey takes 40 min, the drive through the free bridge takes 32 minutes. These times must be weighted by their probability given by the probability of both the states of traffic information and add 5 min to both for the time needed to find out the prediction. On average, we would reach Oakland in:

$U(z_{pred}) = 5 + P(z_{pred}) \times 40 + P(\neg z_{pred}) \times 32 = 5 + 0.48 \times 40 + 0.52 \times 32 = 40.8$ min

This travel time needs to be compared with the default option without any information. This option is the train journey in 40 minutes.

Conclusion 3: Traffic information helps to increase quality of our decision. However, the 5 minute time for its acquisition is too large. The best option is to simply set out by train.

2 Markov decision processes

Example 3. *Concern an episodal process with three states* $(1, 2, 3)$. *The rewards in individual states are* $R(1) = -1$ $R(2) = -2$, *and* $R(3) = 0$, *the process terminates by reaching state* 3. *In the states* 1 *and* 2, *actions a and b can be applied. Action a keeps the current state with 20% probability, with 80% probability it leads to transition from* 1 *to* 2 *resp. from* 2 *to* 1. *Action b keeps the current state with 90% probability, with 10% probability it leads to state* 3.

Try to guess the best policy qualitatively for states 1 and 2.

We maximize our reward, the rewards are not positive, the process should be terminated as soon as possible, i.e., state 3 should be reached. At the same time, the transition to 3 by b is relatively improbable. The expected number of b trials to terminate the process is 10 (for $p = 0.1$: $E = p + 2p(1-p) + 3p(1-p)^2 + \cdots = 1/p = 10$). It pays off to switch to state 1 first and employ a to terminate the process by transition to state 3.

Conclusion 1: The best policy seems to be $\pi^* = (b, a, NULL)$.

Formalize as MDP. Apply policy iteration. Start with the policy $\pi_0 = (b, b, NULL)$ and illustrate its convergence to the optimal policy in detail.

Init: $\pi_0 = (b, b, NULL)$, $U_0 = (0, 0, 0)$.
Iteration 1:
Evaluation: $U_1(1) = -1 + 0.9 \times U_1(1)$, $U_1(2) = -2 + 0.9 \times U_1(2)$, $U_1(3) = 0$,
$U_1(1) = -10$, $U_1(2) = -20$,
(can be solved by DP, an analytical solution is available too).
Improvement: $Q_1(a, 1) = -1 + 0.2 \times U_1(1) + 0.8 \times U_1(2) = -19$,
$Q_1(b, 1) = -1 + 0.9 \times U_1(1) + 0.1 \times U_1(3) = -10$,
$Q_1(a, 1) < Q_1(b, 1) \to$ for state 1 we pick b,
$Q_1(a, 2) = -2 + 0.8 \times U_1(1) + 0.2 \times U_1(2) = -14$,
$Q_1(b, 2) = -2 + 0.9 \times U_1(2) + 0.1 \times U_1(3) = -20$,
$Q_1(a, 2) > Q_1(b, 2) \to$ for state 2 we pick a,
$\pi_1 = (b, a, NULL)$.
Iteration 2:
Evaluation: $U_2(1) = -1 + 0.9 \times U_2(1)$, $U_2(2) = -2 + 0.8 \times U_2(2) + 0.2 \times U_2(1)$,
$U_2(1) = -10$, $U_2(2) = -12.5$,
(analytical solution again).
Improvement: $Q_2(a, 1) = -1 + 0.2 \times U_2(1) + 0.8 \times U_2(2) = -13$,
$Q_2(b, 1) = -1 + 0.9 \times U_2(1) + 0.1 \times U_2(3) = -10$,
$Q_2(a, 1) < Q_2(b, 1) \to$ for state 1 we pick b,
$Q_2(a, 2) = -2 + 0.8 \times U_2(1) + 0.2 \times U_2(2) = -12.5$
$Q_2(b, 2) = -2 + 0.9 \times U_2(2) + 0.1 \times U_2(3) = -13.25$
$Q_2(a, 2) > Q_2(b, 2) \to$ for state 2 we pick a,
$\pi_2 = (b, a, NULL)$, no policy change, STOP.

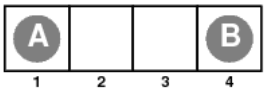
Conclusion 2: We confirmed our qualitative guess, the optimal policy is $\pi^* = (b, a, NULL)$.

Reapply policy iteration. Start with $\pi_0 = (a, a, NULL)$. What happens? What is the solution?

Init: $\pi_0 = (a, a, NULL)$, $U_0 = (0, 0, 0)$.
Iteration 1:
Evaluation: $U_1(1) = -1 + 0.2 \times U_1(1) + 0.8 \times U_1(2)$,
$U_1(2) = -2 + 0.8 \times U_1(1) + 0.2 \times U_1(2)$,
(this system of equations has no solution)
(DP solution diverges, the state values grow towards ∞).

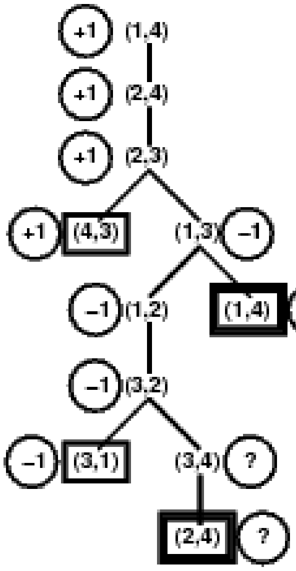
Conclusion 3: The solution is to introduce a discount factor γ. The system of linear equations will not be singular any longer. But, a bit different task gets solved, the best policy can be different, especially for small discount factors. With a small γ, immediate reward gets preferred, one can find b as the best option even in state 2.

Example 4. *Consider a two-player game on a four-field board. Each player has one stone, the goal is to move its stone to the opposite side of the board (A player moves from field 1 to field 4, B player from field 4 to field 1). The player that first reaches its goal field wins. The players may move one filed left or right, they cannot skip their move nor move out of the board. If a neighbor field is occupied by the opponent's stone, the stone can be jumped. (example: if A is in the position 3 and B in the position 2 and A moves left, it ends up in position 1).*



Which player wins? Demonstrate the classic solution based on state space search first.

The state of the game can be represented by the position of both the stones, it can be written down as an ordered pair (s_A, s_B). There are 11 reachable states (state $(4, 1)$ is not reachable). The standard solution is by MiniMax procedure. The game tree is in figure below (the evaluation is for A player, who is a maximization player).
The only bottleneck lies in the fact that the game contains cycles and the standard depth-first MiniMax would fall into an infinite loop. For this reason, we will put the expanded states on a stack. As soon as a cycle is detected, the value of the state is denoted as "?" and the current branch is terminated. When propagating the evaluation we assume that max(1,?)=1 and min(-1,?)=-1. This improvement is sufficient for the given game which does not distinguish beyond wins and losses.



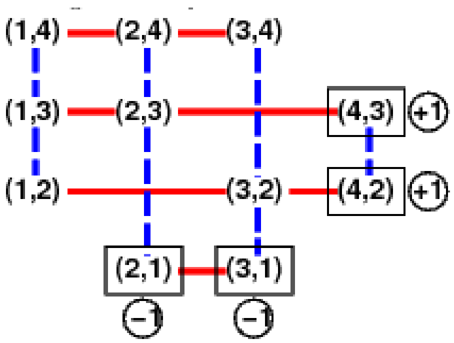
Conclusion 1: Two-player games can be solved by MiniMax. If keeping the optimal game strategy, A player wins (the player who moves first).

Can we formalize this game as MDP? Is it a good choice?

Conclusion 2: Every search problem can be formalized as MDP. The transformation is routine: states and actions do not change, the goal states map to terminal MDP states, the transition matrix is deterministic and the reward is inverted evaluation. However, MDP is not a good choice in the case of deterministic actions. Its formalism is too heavy and time demanding. It is a good choice for stochastic problems.

Formalize this game as MDP. Let $V_A(s)$ be the state s value if A player is on move, $V_B(s)$ be the state s value if B player is on move. Let $R(s)$ be the reward in state s, for the terminal states where wins A it is 1, for the terminal states where wins B it is -1. Draw a state space diagram. Put down Bellman equations for both the players and apply these equations in terms of value iteration. Formulate the iteration termination condition.

The state space diagram is in figure below. The moves of A player are in solid red, the moves of B player in dashed blue.



Bellman equations stem from the MiniMax principal:
$V_A(s) = R(s) + \max_a P^a_{ss'} V_B(s')$
$V_B(s) = R(s) + \min_a P^a_{ss'} V_A(s')$

$R(s)$ will only be used in terminal states, the value of the rest of the states is given solely by its descendants. A player maximizes evaluation, B player does the opposite. As the actions are deterministic, each action has the unit probability for one of the descendant states and zero probability for remaining states.

The players take moves in turns, we apply the individual Bellman equations in turns too. In the beginning, the terminal states start with their $R(s)$, the rest of the states has zero value. The values gradually propagate, the state space diagram is used, see the table below:

s	(1,4)	(2,4)	(3,4)	(1,3)	(2,3)	(4,3)	(1,2)	(3,2)	(4,2)	(2,1)	(3,1)
V_A	0	0	0	0	0	+1	0	0	+1	-1	-1
V_B	0	0	0	0	-1	+1	0	-1	+1	-1	-1
V_A	0	0	0	-1	+1	+1	-1	+1	+1	-1	-1
V_B	-1	+1	+1	-1	-1	+1	-1	-1	+1	-1	-1
V_A	+1	+1	+1	-1	+1	+1	-1	+1	+1	-1	-1
V_B	-1	+1	+1	-1	-1	+1	-1	-1	+1	-1	-1

The termination condition is no change in value vector for one of the players (i.e., the match between the current $V_A(s)$ vector and the $V_A(s)$ vector generated two moves before, or the same match for $V_B(s)$). In the table above, we observe the match in two last $V_B(s)$ vector instances. Obviously, no change may appear for the next $V_A(s)$ too as it will be derived from the identical $V_B(s)$.
Note that $V_A(s)$ and $V_B(s)$ vectors do not have to match. $V_A(s)$ assumes that A player is on move and vice versa (e.g., (3,2) state switches its value in principle as the player on move always wins whoever it is).

Conclusion 3: MDP solves the problem concurrently for both the players taking the first move. The value of the terminal states is given apriori. In states (2,4) and (3,4), A player wins disregarding turns. In states (1,3) and (1,2), B player wins disregarding turns. In states (1,4), (2,3) and (3,2), the player on move wins.
MiniMax tree shown earlier employs different state values at different tree levels, the tree de facto combines $V_A(s)$ and $V_B(s)$ according to the tree depth.