National University of Singapore
School of Computing

Semester 2, AY2023-24

CS4246/CS5446 AI Planning and Decision Making

Tutorial Week 7: MDP

Guidelines

- You can discuss the content of the questions with your classmates.
- However, everyone should work on and be ready to present ALL the solutions.
- Your attendance is marked in the tutorial and participation noted to award class participation marks.

Problem 1: Online Search for Markov Decision Process

Consider an MDP where the state is described using M variables where each variable can take n values. The MDP has 2 actions and at each state each action can only lead to 2 possible next states.

a) What is the size of the state space of this MDP? Can this MDP be efficiently solvable with value iteration as M grows?

Solution:
States space size is n^M. Value iteration is not efficient as M grows as runtime will be exponential in M.

b) A search tree of depth D (number of actions from the root to any leaf is D) is constructed from an initial state s. What is the size of the search tree (the number of nodes and edges) as a function of M and D, in O-notation? Can online search be done efficiently as M grows if D is a fixed small constant?

Solution:
The search tree size is $O(2^{2D})$. If D is a small fixed constant, then online search is efficient as the size of the search tree is constant as M grows (although the computation at each node will still grow at least linearly with M for representing the state).

c) MCTS is used for solving this MDP. What is the size of the search tree if T trials of MTCS is performed up to a search depth of D, as a function of M, D and T in O-notation?

Solution:
Each trial contributes at most D nodes and edges to the search tree, so the size is $O(DT)$.

d) Consider a search tree where the reward is zero everywhere except at the leaves. When a MCTS trial goes through a node, we say that an action at the node wins if the trial ends in a leaf with reward 1. Consider an MCTS simulation where a node has been visited 16 times and has two actions, A and B. Action A has a won 2 out 4 times whereas action B has won 8 out of 12 times. Which action will the MCTS algorithm chose given the exploration parameter c is set to 1? Give the values of π_{UCT} for the node (consider log base 2 in UCT bound).

Solution:
Node A. $\pi_{UCT}(n) = \underset{a}{\text{argmax}}\Big(\hat{Q}(n,a) + c\sqrt{\frac{\log(N(n))}{N(n,a)}}\Big)$. UCT function value for action A is $\frac{2}{4} + \sqrt{\frac{\log 16}{4}} = 1.5$ and for action B is $\frac{8}{12} + \sqrt{\frac{\log 16}{12}} = 1.244$, so $\pi_{UCT}(n) = 1.5$.

Problem 2: Value Iteration

Consider the following 2 state, 2 action MDP with discount factor 0.9.

$P(s_1\|s_1,a_1)$	$P(s_2\|s_1,a_1)$	$P(s_1\|s_2,a_1)$	$P(s_2\|s_2,a_1)$
0.9	0.1	0	1

$P(s_1\|s_1,a_2)$	$P(s_2\|s_1,a_2)$	$P(s_1\|s_2,a_2)$	$P(s_2\|s_2,a_2)$
0.1	0.9	0	1

$R(s_1,a_1)$	$R(s_1,a_2)$	$R(s_2,a_1)$	$R(s_2,a_2)$
1	0	3	3

1. Assume a finite horizon problem with horizon 1 (only 1 action is to be taken). What is the utility or value function and the optimal action in each state?

   Solution:
   $U_1(s_1) = 1$, $U_1(s_2) = 3$, $a^*(s_1) = a_1$, $a^*(s_2) = a_1$ or a_2.

2. Assume a finite horizon problem with horizon 2 (2 actions to be taken). What is the utility or value function and the optimal action in each state?

   Solution:
   Use
   $$U_2(s_i) = \max_a(R(s_i,a) + \gamma\sum_{j=1}^{2}P(s_j|s_i,a)U_1(s_j)).$$
   For state 1 action 1
   $$value(utility) = 1 + 0.9(0.9 * 1 + 0.1 * 3) = 2.08.$$
   For state 1 action 2
   $$value = 0 + 0.9(0.9 * 3 + 0.1 * 1) = 2.52.$$

Taking the max, we get $V_2(s_1) = 2.52$ with action a_2. For state 2, the system will self loop regardless of action with

$$value = 3 + 0.9 * 3 = 5.7.$$

Notice that this policy is different from the policy for horizon 1. Finite horizon problems have non-stationary (time dependent) policies.

3. What is the optimal infinite horizon policy?

   Solution:
   At state s_2, the system self-loops with reward 3 regardless of the action taken, so the infinite horizon utility or value at state 2 is $\frac{3}{1-\gamma} = \frac{3}{1-0.9} = 30$.
   Once the action in state s_2 is fixed, there are two possible policies corresponding to action a_1 and a_2 in state s_1.
   If action a_1 is taken, the value of the policy must satisfy

   $$U(s_1) = 1 + 0.9(0.9U(s_1) + 0.1 * 30)$$

   giving $U(s_1) = 19.47$.
   If action a_2 is taken, the value of the policy must satisfy

   $$U(s_1) = 0 + 0.9(0.9 * 30 + 0.1U(s_1))$$

   giving $U(s_1) = 26.7$.
   Hence action a_2 should be taken in state s_1.

Problem 3: Bellman operator

[RN 17.6] Suppose that we view the Bellman update

$$U_{t+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a)U_t(s')$$

as an operator B that is applied simultaneously to update the utility of every state, that is,

$$U_{t+1} \leftarrow BU_t \ .$$

We claim that the Bellman operator B is a contraction.

1. Show that, for any function f and g,

   $$|\max_a f(a) - \max_a g(a)| \leq \max_a |f(a) - g(a)| \ .$$

2. Write out an expression for $|(BU_t - BU'_t)(s)|$ and then apply the result from part 1 to complete the proof that the Bellman operator B is a contraction.

Solution.

1. Assume, w.l.o.g., that $\max_a f(a) \geq \max_a g(a)$. Also, let $a^* = \arg\max_a f(a)$. Then,

$$\begin{aligned}
|\max_a f(a) - \max_a g(a)| &= \max_a f(a) - \max_a g(a) \\
&= f(a^*) - \max_a g(a) \\
&\leq f(a^*) - g(a^*) \\
&\leq \max_a |f(a) - g(a)| \,.
\end{aligned}$$

The first equality is by assumption. The first inequality is due to $g(a^*) \leq \max_a g(a)$. The last inequality follows from the definition of max.

2. For any s,

$$\begin{aligned}
|(BU_t - BU'_t)(s)| &= \left| R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U_t(s') - R(s) - \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U'_t(s') \right| \\
&= \gamma \left| \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U_t(s') - \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U'_t(s') \right| \\
&\leq \gamma \max_{a \in A(s)} \left| \sum_{s'} P(s'|s,a) U_t(s') - \sum_{s'} P(s'|s,a) U'_t(s') \right| \\
&= \gamma \max_{a \in A(s)} \left| \sum_{s'} P(s'|s,a) (U_t(s') - U'_t(s')) \right| \\
&= \gamma \left| \sum_{s'} P(s'|s,a^*(s)) (U_t(s') - U'_t(s')) \right|
\end{aligned}$$

The first inequality follows from part 1. Inserting the above into the expression for max norm,

$$\begin{aligned}
||(BU_t - BU'_t)|| &= \max_s |(BU_t - BU'_t)(s)| \\
&\leq \gamma \max_s \left| \sum_{s'} P(s'|s,a^*(s)) (U_t(s') - U'_t(s')) \right| \\
&\leq \gamma \max_s |U_t(s) - U'_t(s)| \\
&= \gamma \, ||U_t - U'_t||
\end{aligned}$$

Therefore, the Bellman operator B is a contraction.