Homework 1 – Solutions

CSE 3521 – Summer 2018

Instructor: Andrew R. Plummer

Department of Computer Science and Engineering
The Ohio State University

June 12th, 2018

Concept check

1. (1 points) For each of the following search algorithms, state which kind of data structure models their collection of fringe/frontier nodes: breadth-first search, depth-first search, A^*-search.

   (1/3 point each) BFS: queue (First In First Out); DFS: stack (Last In First Out); A^*-search: priority queue.

2. (1 points) Provide an admissible and consistent heuristic for the state space graph shown in Figure 1.

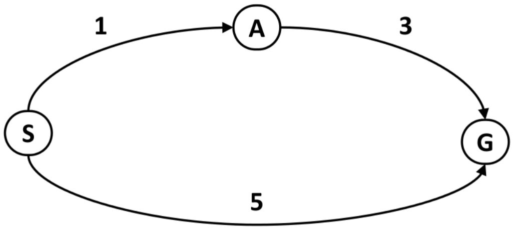
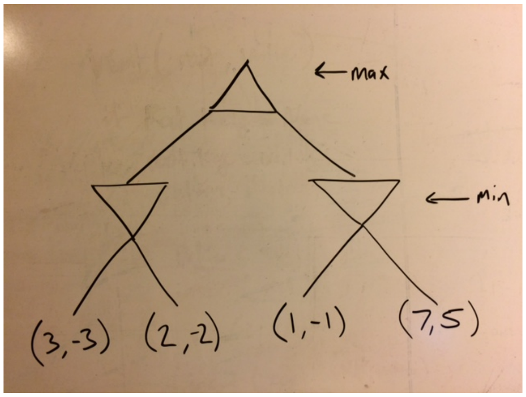


Figure 1: A simple state space graph (from Berkeley AI slides).

   (all or nothing point-wise) One example is: $h(S) = 2$, $h(A) = 1$, $h(G) = 0$. Let's verify that h is admissible. We indeed have $h(G) = 0$, which is necessarily the case. Next, $h(S) = 2$, so we need to verify that this heuristic value is less than or equal to the true cost to a nearest goal. The path $S \rightarrow G$ has cost 5 and the path $S \rightarrow A \rightarrow G$ has cost 4, so we're okay with $h(S) = 2$. Finally, we have $h(A) = 1$ which is okay since the path $A \rightarrow G$ has true cost of 3. Now let's verify that the heuristic is consistent. In this case, we only need to show that $h(S) - h(A)$ is less than or equal to the true cost of $S \rightarrow A$. Indeed, $h(S) - h(A) = 1$, which is less than or equal to the cost of the path $S \rightarrow A$, which is 1.

3. (2 points) In general, given a non-zero-sum game setting, can we still make use of alpha-beta pruning? In either case, provide an argument and example illustrating your answer. To get started, try re-expressing the terminal values in the zero-sum game trees we've looked at using vectors where each component represents a value to each respective player.



(1 point for correct answer, 1 point for arguing either way) In the game tree above, the vectors at the base are of the form $(maxUtil, minUtil)$. Note that the vector $(7, 5)$ cannot be pruned under the assumption that min is trying to maximize their own utility, but could be pruned under the assumption that min is trying to minimize max's utility. In short, the tree illustrates that, in general alpha-beta pruning is not possible in non-zero sum games.

4. (2 points) In your own words, explain the relationship between how the expectimax algorithm works and how Q-states work in Markov Decision Processes (MDPs). Try to be as explicit as possible about the parallel between the two concepts.

Just recap slide 13 from the mdp1 slides. In words, the optimal action that an agent takes at a state is selected from expected value computations at each state-action pair, or Q-state, available to the agent. Thus the Q-states correspond to the game play of an exp agent in an expectimax game.

5. (2 points) In your own words, explain the role of the "Markov assumption" in our formulation of MDPs, i.e., why are we making this assumption about our decision processes?

   Just recap slide 7 from the mdp1 slides.

6. (2 points) In your own words, explain the key differences between value iteration and policy iteration for solving MDPs.

   Just recap slide 39 from the mdp2 slides.

Coding

1. (4 points) [From RN, page 116] The "missionaries and cannibals" problem is usually stated as follows. Three missionaries and three cannibals are on one side of a river [let's assume the left side], along with a boat that can hold one or two people [and must hold at least one]. Find a way to get everyone to the [right] side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place. This problem is famous in AI because it was the subject of the first paper that approached problem formulation from an analytical viewpoint (Amarel, 1968).

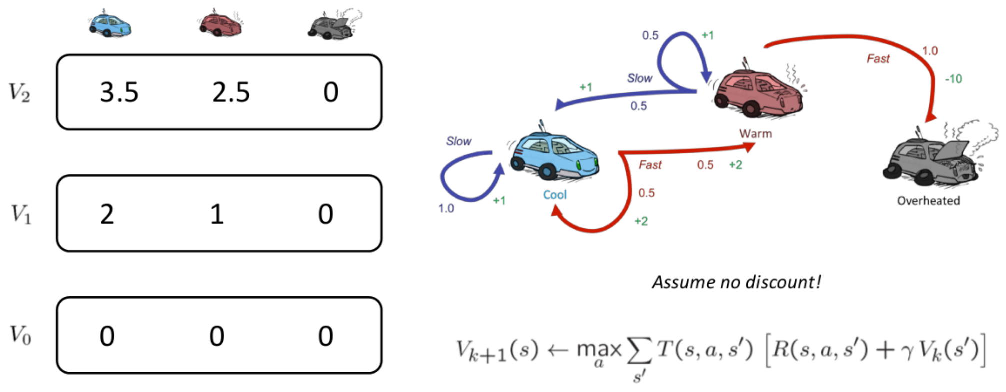
   Write a python script that implements and solves the problem optimally using an appropriate search algorithm. Make sure your script produces an optimal path from the initial state to the goal state.

   1 point for representing the states, 1 point for representing the actions, 1 point for implementing search algorithm, 1 point for an optimal path from initial state to goal state.

2. (4 points) Implementing A-star search. Using the pacman project resources from the Berkeley AI Materials site (`http://ai.berkeley.edu/home.html`), do Question 4 from the search project (`http://ai.berkeley.edu/search.html#Q4`). When you submit your answer, make sure you submit all the scripts needed to run your implementation.

   1 point for correct use of priority queue, 2 points for proper updating of the frontier during expansion, i.e., use of heuristic and cost-to-go, 1 point for getting correct path to goal.

3. (4 points) Implementing value iteration. Implement question 1 from the pacman project on reinforcement learning (`http://ai.berkeley.edu/`



V_2	3.5	2.5	0
V_1	2	1	0
V_0	0	0	0

Assume no discount!

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

Figure 2: MDP for keeping cars happy (from Berkeley AI slides).

reinforcement.html#Q1). Ignore the autograder instructions since we're not going to use an autograder to grade the submissions. Again, when you submit your answer, make sure you submit all the scripts needed to run your implementation.

2 points for the value iteration driver code, 1 point for implementing q-value computation, 1 point for implementing the action from values.

4. (4 points) Write a script to compute V_{18} and V_{19} for the cool and warm states in the MDP shown in Figure 2.

   1 point for representing the states and the actions, 2 points for implementing value iteration, 1 point for correct values.

Fun with proofs

1. (2 points) Prove that every consistent heuristic is admissible. Show by example that there exist admissible heuristics that are not consistent.

   (1 point for the proof, 1 point for the example, but basically full credit for a solid attempt since this one was a little challenging) Given a state space graph G, assume we have a consistent heuristic h on G. The proof is by induction on the shortest path distance of a node in the state space from the closest goal node. Let n be a node with a goal state g as a successor state, and let $c(n, g)$ be the true cost of moving from n to g. Since h is consistent

we have that $h(n) \leq c(n, g) + h(g) = c(n, g) + 0 \leq c(n, g) = h^*(n)$. Now, let n' be on the shortest path k steps away from a goal g, and assume that $h(n') \leq h^*$. If n' is a successor of a node n, then since h is consistent, we have:

$$h(n) \leq c(n, n') + h(n') \leq c(n, n') + h^*(n') \leq h^*(n).$$

See the example at on slide 61 of the informed-search slides for a heuristic that is admissible but not consistent.

2. (2 points) Assuming a discount γ where $0 < \gamma < 1$, prove that the value iteration values V_k converge for all states of an MDP as k increases toward infinity.

   All you had to do was express the proof from the slides in your own words.