CS181 Assignment 5—Markov Decision Processes and Reinforcement Learning

Lucas Freitas and Angela Li

April 27, 2013

1. (a) We have a probability distribution $P(\text{points} \mid \text{target})$ and a utility function $U(\text{points, score})$. The expected utility of aiming for a target t is:

$$\sum_{k \in K} P(\text{k}|\text{t})\, U(\text{k, S})$$

Where K is the set of possible points scored after one throw, and S is the current score. To determine the optimal action, we use the maximum expected utility principle:

$$\operatorname*{argmax}_{t \in T} \sum_{k \in K} P(\text{k}|\text{t})\, U(\text{k, S})$$

Where T is the set of possible targets we can aim for.

(b) This utility function works well for paring down the score in as few dart throws as possible, but only until it reaches the point where it is possible to win in one throw. At that point, it makes decisions poorly in that it values a non-winning throw as nearly as good as a winning throw.

For example, if the current score was 20, a throw resulting in a 20-point gain would end the game (and thus should be valued extremely highly). However, a throw resulting in a 19-point gain (which requires, at the very least, one more throw to win the game) would be valued at only 5% less utility than a winning throw. So the proposed utility function is not conducive to good decision-making in states where it is possible to win in one move—in those cases, the winning move should be valued significantly more highly.

2. (a) In our MDP model, the states are the possible scores in the game (so every integer in the range [0, `START_SCORE`] is a state), and the actions are the possible areas (ring and wedge) that the dart player can aim for in any given turn.

Therefore `get_states()` should return `range(throw.START_SCORE + 1)`.

(b) The reward function should not depend on the action a - it should only take into account if the user has won ($s == 0$) or not yet. Thus, we can write the function as `return 1 if s == 0 else 0`. The problem about this reward function is that it doesn't show our preference of winning earlier than late.

That is why the discount factor is important - we should use it to show that bias for earlier wins. If the discount factor was 1, then we would still not be able to add that preference (we would still not care if we won early or late), and if the discount factor was 0, that would mean that we don't care about winning later at all, which is not good, since we will definitely need future actions in order to win eventually. Thus, we should have a discount factor between 0 and 1, closer to 0 in order to show the preference for winning early.

(c)

(d) There is no guarantee that the darts game can be won in a certain number of steps, so it doesn't make sense to impose an arbitrary finite horizon on the game—after all, the optimal policy should place primary importance on being able to get to a score of 0, and secondary importance on accomplishing that score in as few steps as possible.

(e)

(f)

3. (a)

   (b)

   (c)

4. (a)

   (b)

   (c)