Computer Science 181

Joshua Lee

Homework 5

1 1a

Let $P(s'|s,a)$ be the probability of transitioning from state s to state s' by taking action a. Let $U(p,s)$ be the utility of gaining p points in state s. We can find the optimal policy $\pi : S \to A$ using the following equations:

$$Q(s,a) = \sum_{s' \in S} P(s'|s,a) \cdot U(s - s', s)$$

$$\pi(s) = \arg\max_{a \in A} Q(s,a)$$

Note: Normally our calculation of $Q(s,a)$ depends on some function $R(s,a)$ that represents the reward of taking action a in state s. However, we note for the darts scenario, the reward you get only depends on the state s' (where you actually end up). Thus we only used the expected utility to calculate $Q(s,a)$.

2 1b

This utility function aims to maximize the number of points obtained each throw. It correctly takes into account the fact that throws that yield p points with $p > s$ where s is your current state do not change your state. However, it does not take into account the fact that you want to end the game. For example, it "easier" (o.e. there exists higher probability) of reaching state 0 from state 6 than reaching state 0 from state 1. This is because on a dart board, there are multiple ways to get a score of 6 with one throw (single 6, double 3, triple 2) but only one way to get a score of 1 (single 1).

So, this utility function does well at the beginning of the game, when you are simply trying to get your score as low as possible to put you into a position to finish the game. But this utility function does poorly at the end of the game in that it does not acknowledge the fact that minimizing your score is not an optimal strategy to end the game.

3 2b

We write a reward function $R(s,a)$ that represents the expected number of points you receive by taking action a in state s. We use expected number of points because the number of points (i.e. reward) you get depends on the state s' that you actually end up in.

The discount factor plays two roles. First it represents how much we value future reward. The closer γ is to one, the more we value future reward. Second, it determines how fast our value iteration algorithm will converge. The closer γ is to zero, the faster it will converge.

4 2d

We have finite states and finite actions, so infinite horizon is guaranteed to have a unique solution which converges to the Bellman equations.