CPSC 633-600 Homework 2 (Total 100 points)
Reinforcement Learning

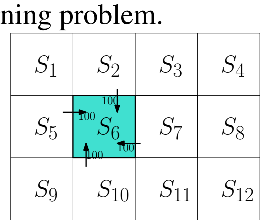
See course web page for the **due date**.
Use **ecampus.tamu.edu** to submit your assignments.

Instructor: Yoonsuck Choe

February 15, 2017

1 Deterministic Case

Consider the following reinforcement learning problem.



- There are 12 states, and the actions are $\{up, down, left, right\}$. Legal actions are those that go to the immediate neighbor, horizontally or vertically (but not diagonally). State S_7 is the goal state, and the only action here is to itself with reward 0. Treat State 7 (s_7) as having no legal action.
- The rewards for all action are 0, except for all actions that lead into s_7, which are 100.
- In all cases, assume $\gamma = 0.9$.

Problem 1 (Program: 20 pts): Program a Q-learning algorithm to learn the $Q(s, a)$ values for the above example. Use the algorithm in slide04.pdf, Mitchell slide page 18 (4-up pdf page 6). You have to find out what stopping criterion to use. You can try a fixed value for max iteration, or observe the max difference in $Q_n(s, a)$ and $Q_{n+1}(s, a)$, and stop when the difference drops below a certain level over 100 iterations or 1000 iterations, etc. Note: use a random policy to select action a given current state s (take care to check if the random action chosen is a legal one).

(1) Include your code.

(2) Show resulting Q table (12×4 matrix).

- Rows represent state and columns represent action.
- Row ordering should be $s_1, s_2, ..., s_9$.
- Column ordering should be $up, down, left, right$, from left to right. **If you use a different ordering you will get 0 points for this problem.**
- Set $Q(s, a) = -1$ to mark illegal moves. Don't use this value during your calculations.

(3) Show a plot showing $\sum_{s,a} |Q_{t+1}(s,a) - Q_t(s,a)|$ over the iterations t.

Problem 2 (Program: 20 pts): Modify the program from problem 1 so that the exploration policy is ϵ-greedy. Initialize your Q table with a very small random number to break the initial tie (rand * 0.0001).

(1) Include your code.

(2) Test $\epsilon \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Note: $\epsilon = 0.0$ is the greedy policy, and $\epsilon = 1.0$ is the random policy. (ϵ-greedy chooses the greedy move with $(1-\epsilon)$ probability.)

If `rand()` $> (1-\epsilon)$, choose random action. Otherwise, choose `[val, a] = max(Q(s,:))`.

(3) Show resulting Q tables for all 4 cases (12×4 matrix).

(4) Show plots showing $\sum_{s,a} |Q_{t+1}(s,a) - Q_t(s,a)|$ over the iterations t for all 6 cases with different ϵ values.

(5) Discuss the effect of ϵ on the quality of the learned Q-table.

2 Stochastic Case

Consider a stochastic version of the reinforcement learning problem posed in Section 1. Modify the rules so that:

- $\delta(s,a)$ is stochastic: The probability of landing in the intended direction is 0.70. The probability of landing in one of n unintended legal direction is $\frac{0.30}{n}$.
- Example 1 : If you are in s_5 and a was right, probability of landing in s_6 is 0.70, and ending up in s_1 or s_9 is 0.15 each.
- Example 2: If you are in s_1 and a was down, probability of landing in s_5 is 0.70, and ending up in s_2 is 0.30.
- Reward $r(s,a)$ depends on where you landed based on the above. All rewards are 0 unless the resulting state was the goal state s_6. For example, if you were in s_7 and the action was $a = right$, with 10% chance you will land in s_6, the goal state. In this case $r(s_7, right) = 100$ for that specific run. In a different run, if you landed in s_{11}, then $r(s_7, right) = 0$.

Problem 3 (Program: 20 pts): Repeat problem 1, with the stochastic version of the task (random policy). In addition to all the requirements, keep a running estimate of $E[r(s,a)]$ for states s_2, s_5, s_7, and s_{10} and report their **final** values. Use the learning rule in slide04.pdf, Mitchell slide page 31 (4-up pdf page 9).

Estimating $E[r(s,a)]$ throughout the learning run:

$$E[r(s,a)] = \frac{\sum_{\forall \text{ visits to } (s,a)} (\text{observed reward } r)}{visits(s,a)}$$

Problem 4 (Program: 20 pts): Repeat problem 2, with the stochastic version of the task (ϵ-greedy policy with the four different ϵ values). In addition to all the requirements, keep a running estimate of $E[r(s,a)]$

for states s_2, s_5, s_7, and s_{10} and report the values. Use the learning rule in slide04.pdf, Mitchell slide page 31 (4-up pdf page 9).

Problem 5 (Written: 10 pts): For states s_2, s_5, s_7, and s_{10} manually compute $E[r(s,a)]$ (using the exact probabilities **[note: it relates with $P(s'|s,a)$ and the reward depending on state outcome s']**) and compare those to the estimated values from problem 3 and problem 4. Are the results similar?

Problem 6 (Written: 10 pts): For states s_2, s_5, s_7, and s_{10} using the estimated $E[r(s,a)]$ and all the estimated $\hat{Q}(s,a)$ values from your simulation result in problem 3 above, see if the following holds:

$$\hat{Q}(s,a) = E[r(s,a)] + \gamma \sum_{s'} P(s'|s,a) \max_{a'} \hat{Q}(s',a')$$