Reinforcement learning assignment

In a nutshell

We want you to implement a reinforcement learning (RL) method for an episodic stochastic environment with a discrete state-action space.

Clarified

We have an environment and an agent. We are looking for an optimal policy for this environment. We assume that the state-action space is discrete. In other words, the agent knows that it can choose from a certain number of actions. It also knows that it can experience a limited number of discrete states. However, the agent knows neither the dynamic of this environment nor the meaning of a state. Furthermore, it is a stochastic environment, so the agent's action may result in an unintended consequence. In other word, the transition from one state to another state is stochastic. In addition, the environment is episodic, which means the agent will end up in a terminal state after performing some actions. In this case, the sequence of states experienced and actions performed would make up one episode.

We assume that the agent can perform four actions at each state. An action is represented by a positive integer number $a \in \{1,2,3,4\}$. A state is also represented by a positive integer number $1 \le s \le 16$. `next_state(s, a)` will be your interface to the dynamic of the environment.

```
function [ terminal, snext, reward ] = next_state( s,a )
```

- `s` denotes the current state
- `a` denotes the action; `a` belongs to {1, 2, 3, 4}
- `snext` is the next state after performing action `a` and being in `s`
- `reward` is the immediate transition reward
- `terminal` is a boolean
    - if `terminal==0`, `snext` is a non_terminal state otherwise it is a terminal state indicating the end of an episode

Problems

1. You are given a starting state, a discount factor, and the `next_state( s,a )`function. We want you to pick up an RL method and return the learned policy. We want a deterministic policy. There are 16 states, so the policy will be a row vector of size 16. The first value of this vector should correspond to the best action for the first state. The other elements of this vector should be assigned values in the same way. In other words, the indexes of this vector indicate the states, and each value of the vector relates to the best deterministic action for the corresponding state. Return zero for the states which your algorithm did not or could not explore. We have provided you some Matlab functions. You need to implement the `learnpolicy.m` function. Check the "assignment3.zip" folder. There is enough comment explaining the functions. **Important note**:

Your code should return a policy in less than 7 minutes. Check `question_1.m` file. This Matlab script helps you check your returned policy.

2. Which RL method have you selected and explain very briefly why?
3. Extract the policy from 10, 100, 1000, and 10000 episodes. Evaluate these four policies using 10000 episodes. Keep the starting state of both the evaluation and the learning cases the same. Check `question_3.m` file. This Matlab script helps you evaluate your policies by calculating the average reward of a policy over 10000 policies. Report the average rewards for the aforementioned four policies. In other words, we want you to report 4 numbers. For example, the first number indicates the average reward if you are limited to learn a policy only from 10 episodes.
4. How would you compare the speed of learning in value iteration versus the RL method you have selected for an environment with a known dynamic model?

Rules

- Your solution should be submitted through MyCourses portal.
- The deadline for the assignment is November 2.
- You need to submit two files, a zip file containing the `learnpolicy.m` file and assignment3.pdf including your answers to questions 2 to 4.
- By submitting your solution, you affirm that you have developed the solution yourself.
- You are allowed to discuss the task with other students but you are not allowed to co-operate beyond discussion.

You can contact Murtaza Hazara (Murtaza.Hazara@aalto.fi) for clarifying the assignment or if you need support.

Grading

Maximum 20 points

- +10 points for the `learnpolicy.m` function
- +2 points for question 2
- +4 for question 3
- +4 for question 4
- -1 point per each day the deadline is exceeded