Reinforcement Learning: Homework Assignment 2 (Semester 2 - 2016/17)

Subramanian Ramamoorthy and Svetlin Penkov

3 March 2017

Instructions:

- This homework assignment is to be done *individually*, without help from your classmates or others. Plaigarism will be dealt with strictly as per University policy.
- Solve all problems and provide your **complete** solutions (with adequate reasoning behind each step, and citations where needed) in a computer-printed form.
- This assignment will be marked out of a 100 points, and will count for 10% of your *final* course mark. It is due at 4 pm on 28 March 2017.

1 Actor-Critic Architecture [25 points]

1. Describe the actor-critic architecture for temporal difference based reinforcement learning. Your task is to read about this method and write the description in your words. In addition to the basic description,
   - Give a description of one application example where this architecture has been used, and explain why the actor-critic architecture was beneficial in that application.
   - How does the SARSA algorithm relate to the actor-critic architecture.

(As a guideline, we expect your answer to this question to need not more than 1 page. A starting point for your reading is Sec 6.6 of the Sutton and Barto text book, print edition.)

2 RL with Function Approximation [75 points]

1. Consider the following linear approximation of the $Q_t(s,a)$ state-action value function at time t:

$$Q_t(s,a) = \boldsymbol{\theta}_t^T \boldsymbol{\phi}_{s,a} = \sum_{i=1}^{n} \theta_t^i \phi_{s,a}^i \tag{1}$$

   where θ_t^i and $\phi_{s,a}^i$ denote the i^{th} component of the corresponding $n - dim$ vectors. Explain how the features vector $\boldsymbol{\phi}_{s,a}$ and the parameters vector $\boldsymbol{\theta}_t$ should be constructed in order to reproduce the tabular case of the Q function. [10 points]

2. We can write an off-policy TD update rule for the linear approximation of the state-action value function in (1) as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \left(r_{t+1} + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \right) \nabla_{\boldsymbol{\theta_t}} Q_t(s_t, a_t) \tag{2}$$

   Implement a reinforcement learning agent for the Enduro game based on equations (1) and (2) where the features vector $\boldsymbol{\phi}_{s,a}$ includes at least one feature corresponding to each of the following behavioural requirements:

   - Collisions should be avoided
   - Moving faster results in passing by more cars
   - Staying in the centre of the road is preferred when possible

   [20 points: 5 points for explaining your design of each feature; 5 additional points for a functioning implementation of the learning agent.]

   Note: The solution to assignment 1 is a good starting point for your implementation. Source code to help get started with this implementation will also be made available by the Teaching Assistant on 6^{th} March 2017 in the following repository: `www.github.com/ipab-rad/rl-cw2`.

3. Based on your implementation of the learning agent,

   (a) Provide the learning curve (i.e., plot(s) of performance achieved over time) for your agent. Compare this against the learning curve of the basic Q-learning algorithm (as in assignment 1). [15 points]

   (b) Discuss the usefulness of each feature by visualising and inspecting the weights associated with it. [15 points]

(c) Report on the convergence rate of the linear function approximation model and analyse it with respect to that for the basic Q-learning algorithm. [15 points]