CS6700 : Reinforcement Learning
Written Assignment #1

Intro to RL and Bandits — Deadline: 15 Feb 2019, 11:55 pm

- This is an individual assignment. Collaborations and discussions are strictly prohibited.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- **Please start early.**

1. Many tic-tac-toe positions appear different but are really the same because of symmetries. How might we amend the learning process described above to take advantage of this? In what ways would this change improve the learning process?
   Now think again. Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?

2. Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself, with both sides learning. What do you think would happen in this case? Would it learn a different policy for selecting moves?

3. Suppose the reinforcement learning player was greedy, that is, it always played the move that brought it to the position that it rated the best. Might it learn to play better, or worse, than a non-greedy player? What problems might occur?

4. The results shown in Figure 2.3 (of course text book uploaded in moodle) should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

5. Define a bandit set up as follows. At each time instant for each arm of the bandit we sample a reward from some unknown distribution. Now the agent picks an arm. The environment then reveals all the rewards that were chosen. Regret is now defined as the difference between the best arm at that instant and the one chosen summed over all times steps. Would the existing algorithms for bandit problems work well in this setting? Can we do better by taking advantage of the fact that all rewards are revealed? For e.g., exploration is not an issue now, since all arms are revealed at each time step.

6. Consider a bandit problem in which you know the set of expected payoffs for pulling various arms, but you do not know which arm maps to which expected payoff. For example, consider a 5 arm bandit problem and you know that the arms 1 through 5

have payoffs 3.1, 2.3, 4.6, 1.2, 0.9, but not necessarily in that order. Can you design a regret minimizing algorithm that will achieve better bounds than UCB? What makes you believe that it is possible? What parts of the analysis of UCB will you modify to achieve better bounds? Note that I am not asking you for a complete algorithm or analysis, only the intuition.

7. Consider a bandit problem in which the parameters on which the policy depends are the preferences of the actions and the action selection probabilities are determined by the softmax relationship as $\pi_t(a) = \frac{e^{\rho_t}}{\sum_{b=1}^{n} e^{\rho_t(a)}}$, where n is the total number of actions and $\rho_t(a)$ is the preference value of action a at time t. Derive the parameter update conditions according to the REINFORCE procedure considering the above described parameters and where the baseline is the reference reward defined as $\overline{r_{t+1}} = \overline{r_t} + \beta[r_t - \overline{r_t}]$, where r_t is the reward received at time t and β is the step size parameter.

8. Repeat the above problem for the case where the parameters are the mean and variance of the Normal distribution according to which the actions are selected and the baseline is zero.