Machine Learning - Assignment 4

Due Thursday, November 22, 2001

1. [25 points] Consider playing Tic-Tac-Toe against an opponent that plays randomly. In particular, assume that the opponent chooses with uniform probability any open space, unless there is a forced move (in which case it makes the obvious correct move).

   (a) [10 points] Formulate the problem of learning an optimal Tic-Tac-Toe strategy against this opponent as a reinforcement learning task. What are the states, actions, rewards and transition probabilities ? How would you represent the Q-function?

   (b) [15 points] Formulate the problem as for a genetic algorithm. What is the fitness function? What is the representation of a player? What types of crossover would you allow, and how would you determine who survives in the next generation of the population?

2. [10 points] Suppose that you have an MDP M_1 with discount factor $0 \leq \gamma < 1$. An arbitrary policy π has value V_1^π in this MDP. Construct an MDP M_2 with discount factor 1, such that $V_2^\pi = V_1^\pi$.

3. [15 points] Prove the error reduction property for N-step returns (discussed in class, lecture 20).

4. [10 points] State aggregation is a special case of generalizing function approximation, in which states are grouped together and one table entry (value estimate) is used for the whole group. When a state in the group is encountered, the group's entry is used to determine the value. When a state is updated, the value for the whole group is also updated.

   (a) Show that state aggregation is a special case of linear gradient descent

   (b) Are tile codings (CMACs) creating just a state aggregation, or not? Explain the similarities and differences.

5. [40 points] In this problem, you will have to implement and compare reinforcement learning algorithms on a linear MDP task. Consider an MDP with 19 states and two actions, for moving left and right. Each action has a probability p of moving in the designated direction. With probability 1-p, the agent stays in the same state. The agent always starts in state 9 (middle of the line). There is a +1 reward for entering the rightmost state, and a -1 reward upon entering the leftmost state. The discount factor is $\gamma = 1$.

   Implement Q-learning and Sarsa for this environment. In both cases, use an ϵ-greedy policy, with $\epsilon = 0.1$. Set the learning rate $\alpha = 0.1$.

   (a) Consider $p = 1$. Start with the value function being 0 everywhere. Run 100 learning trials for each algorithm. After every 10 learning trials, do a test trial, in which the agent uses the greedy policy based on its current values, in order to pick actions. There is no learning during the test trial. For each training and test trial, output the return received during that trial (+1 or -1).

   Repeat this experiment 5 times. Plot the averages of the returns over these 5 independent runs, for the two learning algorithms. Explain your results.

   (b) Repeat the experiment for $p = 0.75$. Before the experiment, what changes did you expect to see compared to the previous graph? Have your expectations been met? Explain any other aspects of the graph that you find interesting.

   (c) Extra credit: Implement Sarsa(λ) for this task as well. Design and perform an experiment to compare its performance with that of Sarsa(0).