Reinforcement Learning S&B

Introduction

Exercises for chapter 5 : An extended example: Tic-Tac-Toe

1. 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?

   Answer proposal. Both agents will learn to beat each other probably resulting in some equilibrium. For that they would of course learn something different than against a random opponent. ■

2. 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?

   Answer proposal. Since positions are symmetrics, one could simply take one state to represent any of its symmetrics since the probability of winning from any of these states would be the same against an optimal player.
   Reducing the number of state would make the learning phase faster by reducing the combinatorics of the simulation. However if the opponent does not take into account the symmetry, hence does not play optimaly, we can't exploit his bad moves if they happen non symmetrically, let's say in one angle of the board for example. Indeed we would notice a non optimal play happen sometime if the representative state that we have taken but we could not grasp a systematic mistake on an angle.
   From this observation, we can deduce that giving the same value to all symmetric states would be a mistake against a non optimal opponent if our goal is to win the most often time possible. ■

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 nongreedy player? What problems might occur?

   Answer proposal. For the example of tic tac toe, I doubt it would change anything as the possible states would be quickly exhausted. However in a more complex case, if an agent is trained to be exclusively greedy, it would probably miss a lot of the possible states and would not generalize well to a real envrionment (a different type of player for example). ■

4. Suppose learning updates occurred after all moves, including exploratory moves. If the step-size parameter is appropriately reduced over time (but not the tendency to explore), then the state values would converge to a different set of probabilities. What (conceptually) are the two sets of probabilities computed when we do, and when we do not, learn from exploratory moves? Assuming that we do continue to make exploratory moves, which set of probabilities might be better to learn? Which would result in more wins?

   Answer proposal. If the agent learn from exploratory moves, it's learning the probability of winning from each state making any move. Oppositely, if it's not learning from exploratory moves, it's learning the probability of winning from each state playing what it deems to be the best moves. Of course learning the probabilityof winning from a state doing the best move looks better than learning the probability of winning from a state doing any move. ■

5. Can you think of other ways to improve the reinforcement learning player? Can you think of any better way to solve the tic-tac-toe problem as posed?

   Answer proposal. Decaying the probability of exploration over time would probably an improvement in the long run as the state space is small in tic tac toe and the probability to miss an optimal strategy against our opponent is low. Another improvement might be to not explore states randomly by accordingly to our knowledge of the state. Frankly, if the opposite player is suboptimal as stated, I can't think of any better strategy, if not brute force is easy here. ■