Reinforcement Learning: An Introduction (Sutton and Barto)

Jeremy Scheurer

Created in February 2018

Contents

1 Chapter 1: The reinforcement learning problem

1.1 Self-Play

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?

Well for one thing we cannot let the learning rate go to zero, because our opponent is constantly changing(learning) and thus we need to adopt aswell. Before we assumed that our opponent plays imperfectly, if this is still the case for ourself, then exploiting a certain weakness might only work once, because the opponent will learn about it. I guess it would still learn the same policies but as both sides improve in the end they will both have found out the best possible moves and thus nobody will ever win.

1.2 Symmetries

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?

We could have not a different value estimation for each possible state the board can be in, but for groups of states. These groups of states each contain states thate are symmetric to each other. By doing this we would have less states in total which would mean we have to explore less to visit each state for at least one time. That means we can have faster convergence to an optimal solution.
Well the problem is that if our oponent does not take advantage of symmetries he could be playing very differently in several states that are symmetric. He could always be playing perfectly in one state and imperfectly in the other. We would then not be able to capture this difference, because both states are

just one state in our model. So it is not necessarily true that symmetrically equivalent positions should have the same value.

1.3 Greedy Play

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?

For this question I will assume that we are not using the UCB algorithm or similar algorithms where taking the greedy choice also includes sometimes exploring states. Thus always making the greedy choice obviously neglects exploring other states. So if you are very lucky you might just make all the optimal actions by always choosing the greedy choice. But this is very unlikely, most probably you will never be able to reach the optimal solution. If we compare this with an intelligent non-greedy algorithm which will explore different actions, this greedy-algorithm clearly underperforms.