CS6700: Written Assignment-1

Name: P Sai Ramana Kiran
Roll Number: AE13B064
Date: 21-1-17

Question - 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?

Answer: The main aim of the game is to win. The essence of win and lose is lost once both sides of the game is played by same type of agents. Agents aim to maximize payoff or return. In this case they will end up in a situation where both agents win and lose alternatively, such that they end up getting same payoff eventually. This is a win-win situation for the both agents.

Question - 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?

Answer: We can amend this by changing the look-up table for the agent. Advantages of this are two folds:

1. Since we are incorporating symmetries agent's state-space (or in this case search space) decreases, thereby increasing the efficiency of the algorithm.
2. This also increases rate of convergence for each state expected return. This is due to fact that value is updated to the same state more number of times compared to the algorithm which doesn't make use of symmetry.

In case the opponent is not making use of symmetry,it will be better for our algorithm to do the same. This is because the objective of the game is to win more number of times with respect to opponent. By making use of symmetry against such an opponent there is a huge possibility that it will backfire and decrease the convergence rate. This is considering the fact that we are updating the states which are not even played yet. This will give us wrong estimate in the expected return of the states which thereby, decreases the performance of the algorithm.

Symmetrically equivalent positions shouldn't necessarily have the same value. It depends on the type of opponent agent is playing with.

Question - 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 non greedy player? What problems might occur?

Answer: Initially it might play better, but after certain amount of time it will play worse than non greedy player. Entire concept of RL is to maximise the expected future reward through exploration-exploitation. If the agent always plays greedily it can't explore some plays which might increase the future reward but receives less return at current instant.

Question - 4

Learning from Exploration: 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 set of probabilities. What 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: When we learn from exploration, the set of probabilities converge to the *true expected return* as opposed to the probabilities learnt without exploration. When we don't learn from exploratory moves the situation is very similar to *frog in a well*. In other words, agent plays the game without knowing the plays which may result in better expected reward. Eventually, probabilities with exploratory moves would result in more wins, since this encapsulates all the possible moves and their corresponding fate. This is the true reinforcement learning.

Question - 5

Other Improvements: 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: Having a concept of "opening moves" will increase the efficiency of the learning. This is something similar to chess, where we have openings. Other

improvements can be like understanding the anatomy of the game,Since tic-tac-toe is a simple game, we can compute all the possibilities at each iteration and select the best possible move.