Jimmy Liang
haoming3
1430396

Both Monte Carlo (MC) and one-step Temporal Difference (TD(0)) cannot optimally solve all problems, but their ideas are still applicable. Generalizing both methods, an n-step TD algorithm is introduced. TD(0) uses the return value in the next time step as the current estimate. In n-step TD, the return value of next n time step is computed before an estimate of the current time step is determined. Then in the end of episode, additional updates are made after termination and before starting the next episode to make up the incomplete computations. By using n-step TD, each estimation is more converging to the final return, therefore accuracy towards the correct answer is more guaranteed than just TD(0).

Sarsa can also be upgraded into n-step version for control problems, simply by changing the TD(n) state value prediction into action value and then using e-greedy policy. One-step Sarsa only updates the last action leading to the reward, compared with n-step Sarsa updates n actions towards the final reward, n-step Sarsa learns more from a single episode.

For off-policy learning, the value function $v_\pi$ is learned by following another policy b. The policy b denotes the exploration policy, and could be e-greedy. The practicality of the result from policy b is determined by the difference between $\pi$ and b, so in n-step off-policy learning, importance sampling of n actions are done, and the importance sampling ratio are taken into account in the value function. The importance sampling ratio is defined as the probability of these n actions. If an action are never taken under a policy ($\pi = 0$), the ratio would be 0 and therefore ignored. If the ratio is 1 ($\pi = b$), this makes the learning on-policy, and the algorithm would be exactly same as n-step TD value estimating. Therefore, the algorithms of n-step TD and n-step Sarsa can be replaced by the new version of using importance sampling ratio.