Chapter 7 - Multi-step Bootstrapping

Atul Balaji - EE16B002

February 26, 2019

- MC methods use an exact sample of return (till episode termination) whereas TD methods use an estimate using a one-step look ahead. n-step methods generalize both methods.
- In TD updates are performed for each time step by bootstrapping on the estimate of the next-state's value. But, bootstrapping works better if the estimates differ over a large temporal range. This is because successive states might be correlated and as a result, there will be only a small difference between successive state's value functions. Therefore, the increments will be less in magnitude and learning will be slower.
- By contrast, n-step returns consider bootstrapping with a state that occurs n time steps in the future. Thus, there would be lesser correlation between the states S_t and S_{t+n}. This results in better learning.

1 n-step TD

- Given state s, the target for n-step backup is the n-step return $G_{t:t+n}$.

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1},\ if\ n \geq 1, 0 \leq t < T - n. \tag{1}$$

$$G_{t:t+n} = G_t\ if\ t + n \geq T. \tag{2}$$

- The target depends on the rewards obtained over n-steps, so, no update will happen during the first (n-1) steps in the episode. To make up for this, the same number of updates are done once the episode terminates.
- As $n \to \infty$, the n-step TD becomes the Monte Carlo update. MC method converges to an estimate which minimizes error on the training data. This means n-step TD gets closer to MC as n $\uparrow$ and hence error on training data for n-step TD is less than one-step TD. This is referred to as the error-reduction property which bounds the max-error in the estimate of the value function.

2 n-step SARSA

- Here, n-step methods are combined with SARSA to produce on-policy TD control. The target is approximated using n rewards and is bootstrapped with an estimate of $Q(S_{t+n}, A_{t+n})$.

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A + t + n),\ n \geq 1, 0 \leq t < T - n. \tag{3}$$

$$G_{t:t+n} = G_t\ if\ t + n \geq T \tag{4}$$

- After n steps, update is done for $Q(S_t, A_t)$. For $s \neq S_t$ and $a \neq A_t$, Q(s,a) remain unchanged.
- For Expected SARSA, instead of using $Q(S_{t+n}, A_{t+n})$, we use expectation over $Q(S_{t+n}, a)$ for all actions a.

3 n-step Off-policy Learning

- Say we are using a behaviour policy b which could be some ϵ-greedy policy and we estimate the target policy π. This method uses importance sampling.
- An off policy version of n-step TD can be made by weighing the samples with the relative probability of taking that action according to policies π and b. The weights here are the **importance sampling ratios** $\rho_{t:t+n-1}$ Here, since n steps are considered, the weight depends on relative probability of those actions A_t to A_{t+n-1}.
- Similarly, off-policy versions of n-step SARSA, n-step expected SARSA can be derived.

4 n-step Tree Backup algorithm

- Consider a particular (S_t, A_t). In this method, the target consists of a backup using $Q(S_{t+1}, x)$ where $x \neq A_{t+1}$ (not selected). It uses the sample (S_{t+1}, A_{t+1}) to bootstrap in order to calculate the **n-step return** $G_{t:t+n}$. Each first level action a contributes with a weight of $\pi(a|S_{t+1})$. The probabilities get multiplied in the subsequent levels.

- A state-action tree is formed, with each node (actions not selected) contributing to the target, its probability of occurring under policy π.

- The n-step return can be found using a recursive formula:

$$G_{t:t+n} = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+n-1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+n}, \tag{5}$$

  for $t < T - 1$, $n \geq 2$.

- The action value update is done only for the state-action pair (S_t, A_t) and the equation can be written as:

$$Q_{t+n}(S_t, A_t) = Q_{t+n-1}(S_t, A_t) + \alpha[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)] \ for \ 0 \leq t < T \tag{6}$$

- The n-step tree backup estimate has lower variance compared to importance sampling based methods.