n-step Bootstrapping

Friday, June 19, 2020 12:00 PM

n-step bootstrapping methods lie in between Monte Carlo (MC) and Temporal Difference (TD) methods in the sense that MC methods use the entire sequence of rewards from a given state until termination to update the value of a state and one-step TD methods use only the reward of the next state to update the value of the previous state

n-Step TD Methods:

Methods in which the temporal difference extends over n steps

n-Step Return:

$$G_{t:t+n} \equiv \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} + \gamma^n V_{t+n-1}(S_{t+n}) = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

Where $n \geq 1$ and $0 \leq t \leq T - n$

Note that these are successively more accurate approximations of the full return as n increases and reach MC methods at $n = \infty$

n-Step TD:

$$V_{t+n}(S_t) \equiv V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

Note that state updates cannot be performed until n time steps after the state has been visited $(t = t + n)$

Also note that no state updates are made during the first $n - 1$ steps of each episode and the same number of extra updates are made after the epidode is over

Error Reduction Property of n-Step Returns:

$$\max_s \left| E_\pi\left[G_{t:t+n} \middle| S_t = s\right] - v_\pi(s) \right| \leq \gamma^n \max_s \left| V_{t+n-1}(s) - v_\pi(s) \right|$$

This says that the worst error of the expected -step return is guaranteed to be less than or equal to the worst error under the previous value function

n-Step Sarsa:

$$Q_{t+n}(S_t, A_t) \equiv Q_{t+n-1}(S_t, A_t) + \alpha\left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)\right]$$

Where:

$$G_{t:t+n} \equiv \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$$

$$= R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n})$$

For n-Step Expected Sarsa:

$$G_{t:t+n} \equiv \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} + \gamma^n \bar{V}_{t+n-1}(S_{t+n})$$

Where the value update process is the same as n-step Sarsa and $\bar{V}_t(s)$ is the expected approximate value

Expected Approximate Value:

$$\bar{V}_t(s) \equiv \sum_a \pi(a|s) Q_t(s, a)$$

If s is terminal this is defined to be 0

n-Step Importance Sampling Ratio:

$$\rho_{t:h} \equiv \prod_{k=t}^{\min(h, T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

This is the relative probability under the two policies of taking the n actions from A_t to A_{t+n-1}

Note that if the two policies are the same this will always be 1 and the off-policy methods below reduce to their on-policy analogs

Off-Policy n-Step TD:

$V_{t+n}(S_t) \equiv V_{t+n-1}(S_t) + \alpha\rho_{t:t+n-1}[G_{t:t+n} - V_{t+n-1}(S_t)]$

Off Policy n-Step Sarsa:

$Q_{t+n}(S_t, A_t) \equiv Q_{t+n-1}(S_t, A_t) + \alpha\rho_{t+1:t+n}[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)]$

Off Policy n-Step Expected Sarsa Return:

$Q_{t+n}(S_t, A_t) \equiv Q_{t+n-1}(S_t, A_t) + \alpha\rho_{t+1:t+n+1}[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)]$

Where as before:

$$G_{t:t+n} \equiv \sum_{k=0}^{n-1} \gamma^k R_{t+k+1} + \gamma^n \bar{V}_{t+n-1}(S_{t+n})$$

$$\bar{V}_t(s) \equiv \sum_a \pi(a|s) Q_t(s, a)$$

n-Step Tree Backup Return:

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

For $t < T - 1$ and $n \geq 2$ with the case handled by the expected Sarsa return (with $G_{T-1:t+n} \equiv R_T$)

The n-Step Tree Backup algorithm uses this return and the n-Step Sarsa value update equation