Model Free Prediction : Additional Topics

Easwar Subramanian

TCS Innovation Labs, Hyderabad

Email : easwar.subramanian@tcs.com / cs5500.2020@iith.ac.in

September 27, 2021

Overview

1. Review

2. Certainity Equivalence Estimate

3. Off Policy Learning

Review

Multi-step TD

- One-step TD

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \left[r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s \right] \\ V(s_t) &\leftarrow V(s_t) + \alpha_t [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \end{aligned}$$

- Two-step TD

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \left[r_{t+1} + \gamma r_{t+2} + \gamma^2 V^\pi(s_{t+2}) | s_t = s \right] \\ V(s_t) &\leftarrow V(s_t) + \alpha_t [r_{t+1} + \gamma r_{t+2} + \gamma^2 V(s_{t+2}) - V(s_t)] \end{aligned}$$

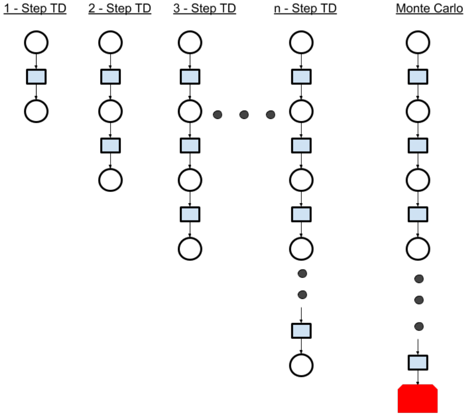
- More generally, define the n-step return

$$G_t^{(n)} \stackrel{\text{def}}{=} r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n})$$

- n-step TD

$$V(s_t) \leftarrow V(s_t) + \alpha_t [G_t^{(n)} - V(s_t)]$$

Why Multi-Step TD ?



- Multi-step TD methods tend to have better bias-variance tradeoff compared to 1-step TD method

λ-Return

- What if we average some or all the n-step returns?

Example : Target could be

$$\frac{1}{2}G_t^{(2)} + \frac{1}{2}G_t^{(4)}$$

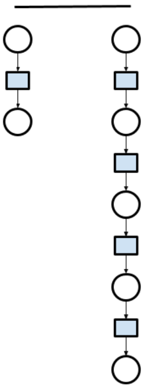


 Figure Source: Sutton and Barto

λ-Return

- Any set of returns can be averaged, even an infinite set, as long as the weights on the component returns are positive and sum to 1
- Choose $\lambda \in [0, 1]$, and define the λ-return at time t as

$$G_t^\lambda \stackrel{\text{def}}{=} (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

- λ-return algorithm: use λ-return as the target

$$V(s_t) \leftarrow V(s_t) + \alpha_t [G_t^\lambda - V(s_t)]$$

- If the episode ends at $T > t$, define $G_t^{(n)}$ to be G_t for all $n > T - t$. Then

$$G_t^\lambda = (1-\lambda) \sum_{n=1}^{T-t} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t} G_t$$

- $\lambda = 0$ gives 1-step TD, $\lambda = 1$ gives MC update

λ-Return

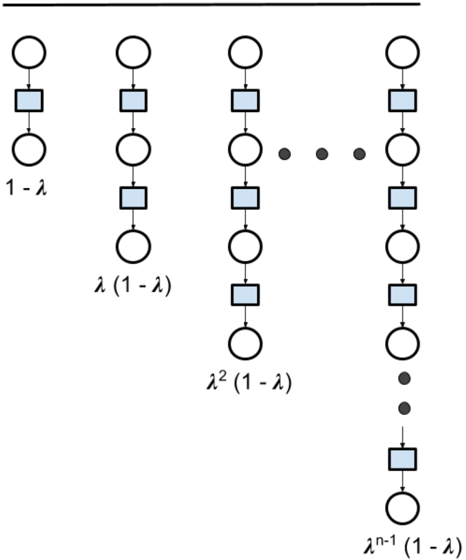
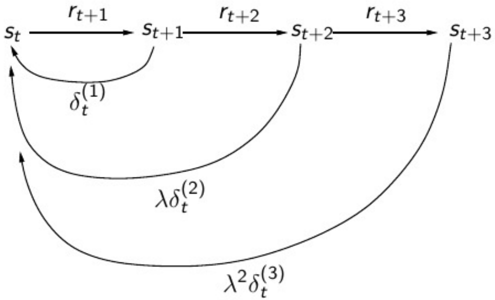


Figure Source: Sutton and Barto

Forward View

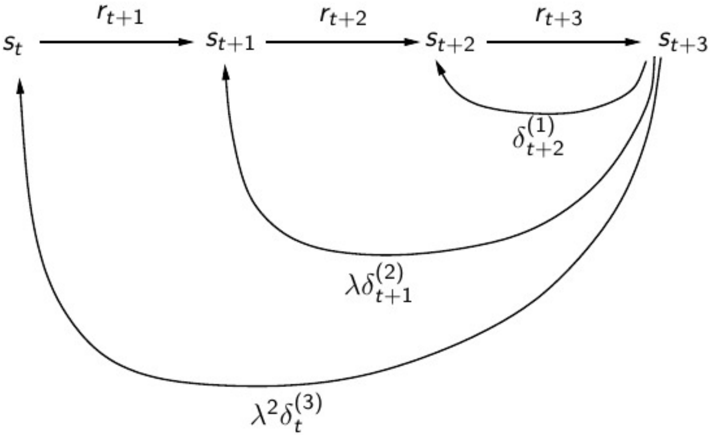
$$G_t^{\lambda} - V(s_t) = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}[G_t^{(n)} - V(s_t)] = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\delta_t^{(n)}$$



- Not suitable for online implementation;

Figure Source: Sutton and Barto

A Possible Online Implementation



- Requires storing all rewards from the episode

Figure Source: Sutton and Barto

A Rearrangement of n-step TD Errors

$$
\begin{aligned}
\delta_t^{(n)} &\overset{\text{def}}{=} r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}) - V(s_t) \\
&= \gamma^0 [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \\
&\quad + \gamma^1 [r_{t+2} + \gamma V(s_{t+2}) - V(s_{t+1})] \\
&\vdots \\
&\quad + \gamma^{n-1} [r_{t+n} + \gamma V(s_{t+n}) - V(s_{t+n-1})] \\
&= \sum_{i=t}^{t+n-1} \gamma^{i-t} \delta_i^{(1)}
\end{aligned}
$$

$$
\begin{aligned}
G_t^\lambda - V(s_t) &= (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} \delta_t^{(n)} = (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} \sum_{i=t}^{t+n-1} \gamma^{i-t} \delta_i^{(1)} \\
&= \sum_{i=0}^{\infty} (\lambda\gamma)^i \delta_{t+i}^{(1)}
\end{aligned}
$$

Online Implementation

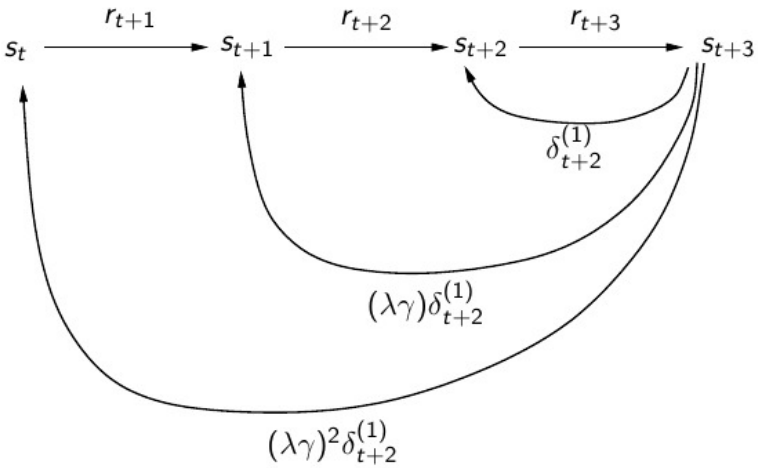


Figure Source: Sutton and Barto

Eligibility Traces

- The eligibility trace of a state $s \in \mathcal{S}$ at time t is defined recursively by

$$
\begin{aligned}
e_0(s) &= 0 \\
e_t(s) &= \begin{cases} (\lambda\gamma)e_{t-1}(s), & s_t \neq s \\ (\lambda\gamma)e_{t-1}(s) + 1, & s_t = s \end{cases}
\end{aligned}
$$

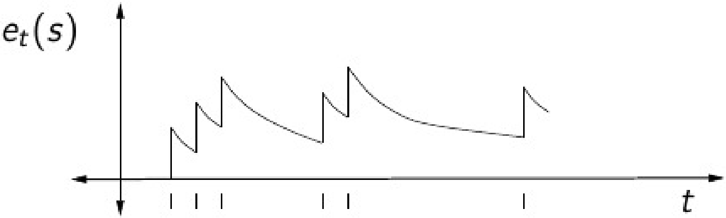


Figure Source: Sutton and Barto

Algorithm : TD(λ)

Algorithm TD(λ) : Algorithm

```
Initialize e(s) = 0 for all s, V(s) arbitrarily
for For each episode do
    Let s be a start state for episode k
    for For each step of the episode do
        Take action a recommended by policy π from state s
        Collect reward r and reach next state s'
        Form the one-step TD error δ ← r + γV(s') − V(s)
        Increment eligibility trace of state s, e(s) ← e(s) + 1
        for For all states S ∈ 𝒮 do
            Update V(S): V(S) ← V(S) + αe(S)δ
            Update eligibility trace: e(S) ← λγe(S)
        end for
        Move to next state: s ← s'
    end for
end for
```

Certainity Equivalence Estimate

Certainity Equivalence Estimate

- Model based policy evaluation with model estimted from samples
- Given an experience quadruple (s, a, r, s'), we can estimate
  - ★ Compute MLE for model using (s, a, s')

$$\hat{P}(s'|s,a) = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{L_k - 1} \mathbb{1}(s_{k,t} = s, a_{k,t} = a, s_{k,t+1} = s')$$

  - ★ Compute MLE estimate of the reward function

$$\hat{R}(s,a,s') = \frac{1}{N(s,a,s')} \sum_{k=1}^{K} \sum_{t=1}^{L_k - 1} \mathbb{1}(s_{k,t} = s, a_{k,t} = a, s_{k,t+1} = s')\ r_{t,k}$$

- Once MLE estimates of $\hat{P}(s'|s,a)$ and $\hat{R}(s,a,s')$ use a known policy evaluation method to compute a MLE based estimate for V^{π}.

Off Policy Learning

Off-Policy Learning

- Can we estimate the value V^π or Q^π of a *target policy* π ...
- While following a *behavior policy* μ?
  - Trajectories or transitions are sampled from μ
  - Expected values have to be estimated w.r.t. π
- Possible benefits:
  - Learn by observing other agents
  - Re-use previous experience generated from earlier policies
  - Learn about optimal policy while following exploratory policy
  - Learn about multiple policies while following one policy

Importance Sampling : Review

Let $P(x)$ be the target distribution and $Q(x)$ be the behaviour distribution for some random variable x

$$\begin{aligned}
\mathbb{E}_{X\sim P}[f(X)] &= \sum_i P(x_i)f(x_i) \\
&= \sum_i Q(x_i)\left[\frac{P(x_i)}{Q(x_i)}f(x_i)\right] \\
&= \mathbb{E}_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]
\end{aligned}$$

- We have samples of X drawn from Q, but we wish to estimate expectation under P

$$\mathbb{E}_{X\sim P}[f(X)] \simeq \frac{1}{n}\sum_{i=1}^{n}\left[\frac{P(X_i)}{Q(X_i)}f(X_i)\right], \quad X_i \sim Q$$

- Caveat: Q should not assign zero probability to any outcome that is assigned non-zero probability by P

Importance Sampling : Review

The ratio $\frac{P(x)}{Q(x)}$ is the importance sampling (IS) weight for x.

What about the variance of IS estimator $\widehat{\mu}_Q \approx \frac{1}{N}\sum_{x_i \in D}\left[\frac{P(x_i)}{Q(x_i)}f(x_i)\right]$?

$$\begin{aligned}
\mathrm{var}_Q\left[\widehat{\mu}_Q\right] &= \mathrm{var}_Q\left[\frac{P(x)}{Q(x)}f(x)\right] \\
&= \left[\mathbb{E}_Q\left[\left(\frac{P(x)}{Q(x)}f(x)\right)^2\right] - \mathbb{E}_Q\left(\left[\frac{P(x)}{Q(x)}f(x)\right]\right)^2\right] \\
&= \mathbb{E}_P\left[\left(\frac{P(x)}{Q(x)}f(x)^2\right)\right] - \mathbb{E}_P\left(f(x)\right)^2
\end{aligned}$$

If the likelihood ratio $\frac{P(x)}{Q(x)}$ is large, the variance of the estimator explodes

Off-Policy TD using Importance Sampling

- Evaluate target policy π using TD targets $r_{t+1} + \gamma V(s_{t+1})$ generated from μ
- Weigh each TD target by the importance sampling factor $\pi(s_t, a_t)/\mu(s_t, a_t)$

$$V(s_t) \leftarrow V(s_t) + \alpha_t \left[\frac{\pi(s_t|a_t)}{\mu(s_t|a_t)} (r_{t+1} + \gamma V(s_{t+1})) - V(s_t) \right]$$

- π may be deterministic and greedy, μ should be stochastic and exploratory
- The case $\mu = \pi$ is called *on-policy* learning
- On Policy Learning : Learn about policy π from experience sampled from π
- Off Policy Learning : Learn about policy π from experience sampled from μ