Policy Evaluation

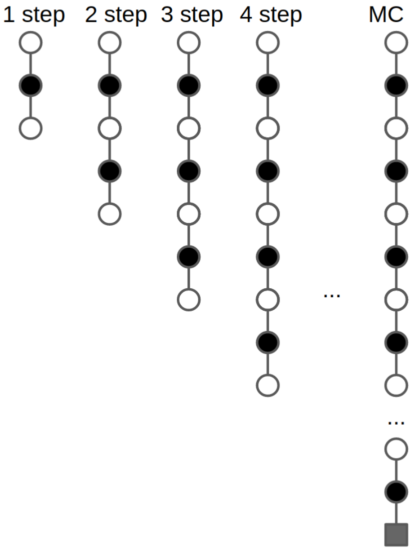
TD(λ)

Marius Lindauer

Leibniz Universität Hannover

Automated Machine Learning Hannover

TD vs. MC



n-Step Return

- Defining n-step returns for different n

$$
\begin{aligned}
n &= 1 \quad (TD) \quad & G_t^{(1)} &= R_{t+1} + \gamma V(s_{t+1}) \\
n &= 2 & G_t^{(2)} &= R_{t+1} + \gamma R_{t+2} + \gamma^2 V(s_{t+2}) \\
&\vdots & &\vdots \\
n &= \infty \quad (MC) \quad & G_t^{(2)} &= R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T
\end{aligned}
$$

- General n-step return

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V(s_{t+n})$$

- n-step temporal-difference learning

$$V(s_t) \leftarrow V(s_t) + \alpha \left(G_t^{(n)} - V(s_t) \right)$$

Averaging n-Step Return

- Hard to say what best n is
- The agent plays the episode anyway and therefore, all updates are possible in principle
- One solution could be to average different n-step updates, e.g.,

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

- Combines information from two different time steps
- Could we combine information from all time steps?

λ-Return

- The λ-return G_t^λ combines all n-steps returns $G_t^{(n)}$
- Using weight $(1-\lambda)\lambda^{n-1}$

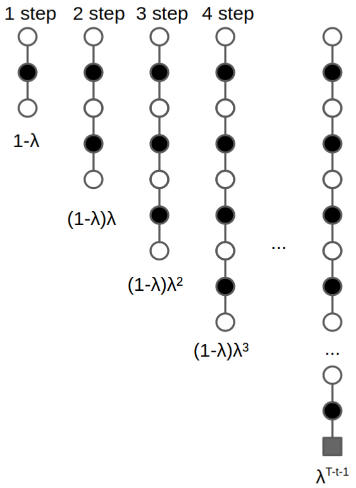
$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

$$\sum_{n=1}^{T-t-2} (1-\lambda)\lambda^{n-1} + \lambda^{T-t-1} = 1$$

- Forward-view TD(λ)

$$V(s_t) \leftarrow V(s_t) + \alpha \left(G_t^\lambda - V(s_t) \right)$$

$\rightsquigarrow$ Like MC, can only be computed from complete episodes



Automated Machine Learning Hannover

Backward View TD(λ)

- Forward view provides theory
- Backward view provides mechanism
- Update online, every step, from incomplete sequences

Eligibility Traces

- Episode: Bell, Bell, Bell, Light, Shock
- Credit assignment problem: Was the bell or the light responsible for the shock at the end?

Automated Machine Learning Hannover

Eligibility Traces

- Episode: Bell, Bell, Bell, Light, Shock
- Credit assignment problem: Was the bell or the light responsible for the shock at the end?
- Frequency heuristic: assign credit to most frequent states
- Recency heuristic: assign credit to most recent states
- Eligibility traces combine both heuristics:

$$
\begin{aligned}
E_0(s) &= 0 \\
E_t(s) &= \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s)
\end{aligned}
$$

$\leadsto$ decrease of importance exponentially proportional to time in the past

$\leadsto$ boost of importance for each time the state was visited

Backward View TD(λ)

- Keep an eligibility trace for every state s
- Update value $V(s)$ for every state s
- In proportion to TD-error δ_t and eligibility trace $E_t(s)$

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \\ V(s) &\leftarrow V(s) + \alpha \delta_t E_t(s) \end{aligned}$$

Automated Machine Learning Hannover

MC, TD(0) and TD(λ)

- When $\lambda = 0$, only the current state is updated
- When $\lambda = 1$, the same as the total update of MC

Automated Machine Learning Hannover