27/oct/

Chapter 6 : Temporal - Difference Learning

No model is assumed (Interplay between dynamic programming and monte-carlo techniques)

Used for problem of prediction - find value function corresponding to policy π.

Look-Up table methods : Estimated values of all states are stored in an array.

Basic Algorithm: $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$

α - step size parameter.

$V(S_t)$ - non ~~sta~~ stationary env

$$V_{t+1}(S_t) = V_t(S_t) + \alpha_t(G_t - V_t(S_t))$$

suppose α_t, $t \geq 0$ are such that

$$\sum_t \alpha_t = \infty \ , \ \sum_t \alpha_t^2 < \infty \qquad \text{eg: } \frac{1}{t+1} \ , \ \frac{1}{(t+1)^\beta} \ \forall t \geq 0$$

α_t eg:

$$\text{eg: } \frac{\log(t+2)}{(t+2)} \ , \ t \geq 0$$

$$V_{t+1}(S_t) = V_t(S_t) + \alpha_t\left(E[G_t - V_t(S_t) \mid S_t] + M_{t+1}\right)$$

where $M_{t+1} = (G_t - V_t(S_t)) - E[(G_t - V_t(S_t)) \mid S_t]$

$$E[M_{t+1} \mid S_0, S_1 \dots S_t] = 0 \quad a.s$$

If the rewards are bounded, $\sup_t |r_t| \leq B < \infty$

then, $\sup_t |G_t - V_t(S_t)| < \infty$ as well

then $N_t = \sum_{m=0}^{t} \alpha_m M_{m+1}$

$\{N_t\}$ – martingale sequence which is convergent

thus $\alpha_m M_{m+1} \longrightarrow 0$ as $m \longrightarrow \infty$

ODE:

$$\dot{V}_t(S_t) = E[G_t - V_t(S_t) | S_t]$$

Suppose $\{S_t\}$ has a unique stationary distribution $d^\pi = (d^\pi(s), s \in S)$ where π is the policy

$$\dot{V}_t(s_t) = \sum_{s \in S} d^\pi(s) \left(\sum_{s', r} p(s', r | s, \pi)(r + \gamma V_t(s') - V_t(s) \right)$$

Stable fixed points of the ODE will correspond to

$$\sum_{s \in S} d^\pi(s) V^*(s) = \sum_{s \in S} d^\pi(s) \left(\sum p(s', r | s, \pi)(r + \gamma V^*(s')) \right)$$

<u>TD(0) for estimating V_π</u>

Input: Policy π to be evaluated, step size at $\alpha \in (0,1]$ & discount factor γ

Initialize $V(s)$, $\forall s \in S^+$, except $V(\text{terminal state}) = 0$

Loop (for each episode)

Initialize s

Scanned by CamScanner

loop for each step of episode

$A \leftarrow$ action given by π for s

Take action A, observe R, S'

$$V(S) \leftarrow V_S + \alpha\left(R + \gamma V(S') - V(S)\right)$$

$$s \leftarrow s'$$

Note: $v_\pi(s) = E_\pi[G_t \mid S_t = s]$ — ~~Mote~~ Monte Carlo Estimate

$$= E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

$$= E_\pi[R_{t+1} + \underline{\gamma v_\pi(S_{t+1})} \mid S_t = s]$$

TD estimate

Boot strapping method.

$$\delta_t \triangleq \left(R_t + \gamma V(S_{t+1}) - V(S_t)\right)$$

TD term error

Note: $E[\delta_t \mid S_t] = 0$.

$$G_t - V(S_t) = R_{t+1} + \gamma G_{t+1} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1})$$

$$= \delta_t + \gamma\left(G_{t+1} - V_{S(t+1)}\right)$$

$$= \delta_t + \gamma \delta_{t+1} + \gamma^2 (G_{t+2} - V(S_{t+2}))$$

$$\vdots$$

$$= \delta_t + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2} \ldots \gamma^{T-t-1}\delta_{T-1} + \gamma^{T-t}(\underbrace{G_T - V(S_T)}_{=0})$$

$$= \sum_{k=t}^{T-1} \gamma^{k-t} \delta_k$$

Scanned by CamScanner

TD (λ)

Given a trajectory of state $i_0, i_1 \ldots i_N$ with $i_N = 0$ [terminal state]

$S = \{1, 2 \ldots N\}$ & 0 is the terminal state.

assume policy is fixed as policy π.

When $i_N = 0$, $i_k = 0 \quad \forall k > N$

Set $r(i_k, i_{k+1}) = 0 \quad \forall k \geq N$

expected reward Value current state $= i_k$ & next state i_{k+1}

$V(i_k)$ = Value estimate when starting from i_k

$$V(i_k) := V(i_k) + \gamma\left(r(i_k, i_{k+1}) + r(i_{k+1}, i_{k+1}) \ldots + \ldots + r(i_{N-1}, i_N) - V(i_k)\right) \quad (2)$$

$$= V(i_k) + \alpha\Big(\big(r(i_k, i_{k+1}) + V(i_{k+1}) - V(i_k)\big)$$
$$+ \big(r(i_{k+1}, i_{k+2}) + V(i_{k+2}) - V(i_{k+1})\big)$$
$$+ \ldots +$$
$$+ \big(r(i_{N+1}, i_N) + V(i_N) - V(i_{N+1})\big)\Big)$$

If $\delta_k = r(i_N, i_{N+1}) + V(i_{k+1}) - V(i_k)$

then, $V(i_k) := V(i_k) + \alpha(\delta_k + \delta_{k+1} \ldots \delta_{N-1})$

Scanned by CamScanner

Mercedes-Benz R & D India

Incremented Updates.

$$V(i_k) := V(i_k) + \alpha \delta_k \quad \Big| \; \gamma = 1 \text{ discount factor}$$

TD(λ)

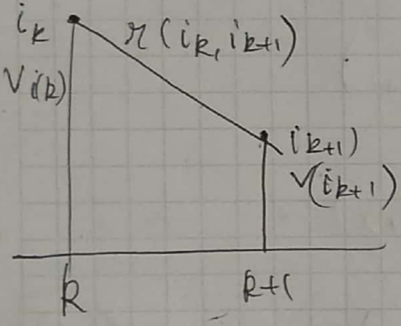
$$V(i_k) = E\left[\sum_{m=0}^{\infty} r(i_{k+m}, i_{k+m+1}) \,\Big|\, i_k\right], \quad i_k \in S$$

* Under the given policy, the process terminates in a finite (possibly random) time w.p 1 [proper policy]

Bellman Eqn:

$$V(i_k) = E\left[r(i_k, i_{k+1}) + V(i_{k+1})\right]$$

$$= E\left[r(i_k, i_{k+1}) + r(i_{k+1}, i_{k+2}) + V(i_{k+2})\right]$$



$$= E\left[\sum_{m=0}^{l} r(i_{k+m}, i_{k+m+1}) + V(i_{k+l+1})\right]$$

Since the value of l is arbitrary, we form a weighted average of the bellman Equations. With ~~wt λ~~ weight λ

Note that: $\sum_{l=0}^{\infty} (1-\lambda)\lambda^l = 1$ We then form a weighted bellman Equation.

$$V(i_k) = (1-\lambda) E\left[\lambda^l \sum_{l=0}^{\infty} \left(r(k+m, i_{k+m+1}) + V(i_{k+l+1})\right)\right] \quad (1)$$

$$(1-\lambda) \sum_{l=0}^{\infty} \lambda^l V(i_k) = V(i_k)$$

$$\therefore V(i_k) = E\left[\sum_{m=0}^{\infty} r(i_{m+k}, i_{k+m+1}), V(i_{k+l+1})\right], \quad l \geq 0$$

Scanned by CamScanner

continuing with ①

$$= (1-\lambda) E\left[\sum_{l=0}^{\infty} \lambda^l \sum_{m=0}^{l} r(i_{k+m}, i_{k+m+1})\right] + (1-\lambda) E\left[\sum_{l=0}^{\infty} \lambda^l V(i_{k+l+1})\right]$$

I II

$$\mathrm{I} = (1-\lambda) E\left[\sum_{l=0}^{\infty} \lambda^l \sum_{m=0}^{l} r(i_{k+m}, i_{k+m+1})\right]$$

$$= (1-\lambda) E\left[\sum_{m=0}^{\infty} \boxed{\sum_{l=m}^{\infty} \lambda^l} r(i_{k+m}, i_{k+m+1})\right] \qquad \sum_{l=m}^{\infty} \lambda^l = \frac{\lambda^m}{1-\lambda}$$

$$= E\left[\sum_{m=0}^{\infty} \lambda^m r(i_{k+m}, i_{k+m+1})\right]$$

$$\mathrm{II} \quad (1-\lambda) E\left[\sum_{l=0}^{\infty} \lambda^l (V i_{k+l+1})\right]$$

$$= E\left[\sum_{l=0}^{\infty} (\lambda^l - \lambda^{l+1}) V(i_{k+l+1})\right.$$

$$= E\left[\cancel{\sum_{k=0}} (1-\lambda) V(i_{k+1}) + (\lambda-\lambda^2) V(i_{k+2}) \ldots\right.$$

$$= E\left[V(i_{k+1}) - V(i_k) + \lambda(V(i_{k+2}) - V(i_{k+1}) \ldots + V(i_k)\right]$$

$$= E\left[\sum_{m=0}^{\infty} \lambda^m (V(i_{k+m+1}) - V(i_{k+m})\right] + V(i_k)$$

Thus, $E\left[\sum_{m=0}^{\infty} \lambda^m r(i_{k+m}, i_{k+m+1})\right] +$

$$E\left[\sum_{m=0}^{\infty} \lambda^m (V(i_{k+m+1}) - V(i_{k+m})\right] + V(i_k)$$

$$V(i_k) = E\left[\sum_{m=0}^{\infty} \lambda^m (r(i_{k+m}, i_{k+m+1}) + V(i_{k+m+1}) - V(i_{k+m}))\right] + V(i_k)$$

Scanned by CamScanner

$$\Rightarrow E\left[\sum_{m=0}^{\infty} \lambda^m \left(r(i_{k+m}, i_{k+m+1}) + V(i_{k+m+1}) - V(i_{k+m})\right)\right] = 0$$

letting $\delta_k = r(i_{k+m}, i_{k+m+1}) + V(i_{k+m+1}) - V_k(i_{k+m})$

Then, $E\left[\sum_{m=0}^{\infty} \lambda^m \delta_{k+m}\right] = 0$

or $E\left[\sum_{m=k}^{\infty} \lambda^{m-k} \delta_m\right] = 0$

not surprising since $E[\delta_m] = 0$ anyways.

Algorithm

$$V(i_k) := V(i_k) + \gamma \sum_{m=k}^{\infty} \lambda^{m-k} \delta_m$$

$\lambda = 1$ TD(1) monte carlo based policy estimation.

$$V(i_k) := V(i_k) + \alpha\left(r(i_k, i_{k+1}) + r(i_{k+1}, i_{k+2}) \ldots r(i_{N-1}, i_N) - V(i_k)\right)$$

$\lambda = 0$ TD(0)

$$V(i_k) := V(i_k) + \gamma \delta_k$$

Check

Example 6.1 Driving Home:

How does one Select λ

Bootstrapping happens when $\lambda < 1$

Better efficiency of algorithm

don't wait for termination, updation happens on the fly