687 2017-11-02

Note Title 11/2/2017

Going from the forward view to the backward view for TD(λ):

Idea: - Store an additional memory variable for each state, called the eligibility trace, $e_t(s)$, also called the e-trace.

- Quantifies how much s should be updated if there is a TD error in the current step

$$e_t(s) = \begin{cases} \gamma \lambda e_{t-1}(s) & \text{if } s \neq S_t \\ \gamma \lambda e_{t-1}(s) + 1 & \text{otherwise} \end{cases}$$

called accumulating traces

(Alternative is "Replacing traces".)

$e_{-1}(s) = 0$ for all s

TD(λ) Algorithm (Backward View)

$e_{-1}(s) = 0$ for all s, for each episode

$\delta = R_t + v(S_{t-1}) - v(S_t)$

$\forall s: e(s) \leftarrow \gamma\lambda e(s)$

$e(S_t) \leftarrow e(S_t) + 1$

$\forall s: v(s) \leftarrow v(s) + \alpha\delta e(s)$

$\lambda = 0 \longrightarrow$ TD

$\lambda = 1 \longrightarrow$ MC ← allows MC updates without waiting to the end of the episode.

$0 < \lambda < 1 \longrightarrow$ TD(λ)

← Easy to apply to more general case (weight updates)

Proof of equiv. of forward + backward views:
[No hard math, just long!]

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

$I_{ss_t} \triangleq 1$ if $s = s_t$ and 0 otherwise

$e_t(s) = \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{ss_k}$

$\Delta V_t^F(s) \triangleq$ update at time t of $v_t(s)$ according to the Forward view $\neq \alpha\left(G_t^\lambda - V_t(s)\right)$ (only S_t is updated)

$\Delta V_t^B(s) \triangleq$ likewise for the Backward view $= \alpha\delta e_t(s)$

Want to show: — Assumes same actions, rewards, + transitions.

$$\forall s \in \mathcal{S}: \sum_{t=0}^{\ell-1} \Delta V_t^B(s) = \sum_{t=0}^{\ell-1} \Delta V_t^F(s) I_{ss_t}$$

where ℓ is the length of the episode.

$$\sum_{t=0}^{l-1} \Delta V_t^B(s)$$

$$= \sum_{t=0}^{l-1} \alpha \delta_t e_t(s)$$

$$= \sum_{t=0}^{l-1} \alpha \delta_t \sum_{k=0}^{t} (\gamma\lambda)^{t-k} I_{sS_k}$$

$$= \sum_{k=0}^{l-1} \alpha \delta_k \sum_{t=0}^{k} (\gamma\lambda)^{k-t} I_{sS_t}$$

$$= \sum_{k=0}^{l-1} \alpha \sum_{t=0}^{k} (\gamma\lambda)^{k-t} I_{sS_t} \delta_k$$

$$= \sum_{t=0}^{l-1} \alpha \sum_{k=t}^{l-1} (\gamma\lambda)^{k-t} I_{sS_t} \delta_k$$

$$= \sum_{t=0}^{l-1} \alpha I_{sS_t} \sum_{k=t}^{l-1} (\gamma\lambda)^{k-t} \delta_k$$

Now, RHS for a single update:

$$\Delta V_t^F(s) = \alpha(G_t^\lambda - V_t(S_t))$$

$$\frac{1}{\alpha}\Delta V_t^F(s) = -V_t(S_t) + G_t^\lambda$$

$$= -V_t(S_t) + (1-\lambda)\lambda^0(R_t + \gamma V_t(S_{t+1}))$$

$$+ (1-\lambda)\lambda^1(R_t + \gamma R_{t+1} + \gamma^2 V_t(S_{t+2}))$$

$$+ \dots$$

$$= -V_t(S_t) + (1-\lambda)\sum_{i=0}^{\infty} \lambda^i R_t$$

$$+ (1-\lambda)\gamma \sum_{i=0}^{\infty} \lambda R_{t+1}$$

$$+ \quad \vdots$$

$$+ [V_t \text{ terms}]$$

$$= -V_t(S_t) + \sum_{k=0}^{\infty} (\gamma\lambda)^k R_{t+k} + \sum_{k=0}^{\infty} (1-\lambda)\lambda^k \gamma^{k+1} V_t(S_{t+k+1})$$

$$= -V_t(S_t) + \sum_{k=0}^{\infty} (\gamma\lambda)^k (R_{t+k} + \gamma V_t(S_{t+k+1}) - \gamma\lambda V_t(S_{t+k+1}))$$

$$= \sum_{k=0}^{\infty} (\gamma\lambda)^k (R_{t+k} + \gamma V_t(S_{t+k+1}) - V_t(S_{t+k}))$$

$$= \sum_{k=0}^{\infty} (\gamma\lambda)^k \delta_k$$

The proof works if δ_t always uses V_t. But real updates don't do that in the Backward view. Using the actual V, the two views are equal up to a difference of $O(\alpha^2)$.

$$\sum_{t=0}^{l-1} \Delta V_t^F(S_t) I_{sS_t} = \sum_{t=0}^{l-1} \alpha I_{sS_t} \sum_{k=t}^{l-1} (\gamma\lambda)^{k-t} \delta_k$$

Coming up: In some cases can adjust Backward View a little and get the Forward + Backward equivalent.

Will also apply TD(λ) idea to Sarsa (Sarsa(λ)) and q-learning ($q(\lambda)$).