687 2017-10-24

Note Title 10/24/2017

Pop Quiz:

$$\delta = r + \gamma v(s') - v(s)$$

$$v(s) \leftarrow v(s) + \alpha[r + \gamma v(s') - v(s)]$$

$$= v(s) + \alpha\delta$$

$$\boxed{\delta_t = R_t + \gamma v(S_{t+1}) - v(S_t)}$$

TD update not quite a gradient update rule. We will now take an excursion into function approximation. (A foreshadowing...)

Q: How can we estimate v^π (or q^π) if the states (or actions) are continuous?

E.g., $S_t \in \mathbb{R}$ or $S_t \in \mathbb{R}^n$

⇒ Problem: infinite – cannot visit all states – or even most! Need to estimate unseen states.

Problem: Cannot store the fn anyway!

Do we care about infinite $\mathcal{S}$?

Yes! Most problems are of this kind.

Assume fn is smooth (almost everywhere).

How to approximate?

Use a parameterized fn (a fn approximator).

E.g. $v_w(s)$ – w a weight vector

$w \in \mathbb{R}^n$ (n not necessarily dimension of $\mathcal{S}$)

Example: Linear fn approx:

$v_w(s) \triangleq w^T \phi(s)$ where $\phi(s) \in \mathbb{R}^n$, called a "feature vector"

E.g. – Polynomial regression $\phi(s) = \begin{bmatrix} 1 \\ s \\ \vdots \\ s^{n-1} \end{bmatrix}$

– Fourier Series $\phi(s) = \begin{bmatrix} \cos(0) \\ \cos(\pi s) \\ \cos(2\pi s) \\ \vdots \\ \cos((n-1)\pi s) \end{bmatrix}$ ← for $s \in [0,1]$

– Tabular:

$$\phi(s) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$s \in \{1, 2, \ldots, m\}$

a 1 only in the sth place

2) Neural network:

$V_w(s)$ = output of a neural network with weights w and input s.

Drawbacks:

– V^π may not be representable

– Thus, fewer convergence guarantees

Benefits & Drawbacks of Function Approximators

Benefits:

– Generalization: can estimate $V(s)$ from states similar to s.

– Can handle continuous states (and actions).

– Will allow us to handle partial observability (Because: We do not assume $\phi(s)$ is a Markovian state representation.)

Back to TD:

$$V(s) \leftarrow V(s) + \alpha\left(r + \gamma V(s') - V(s)\right)$$

For gradient:

$$\mathbb{E}\left[\frac{1}{2}\left(V^\pi(s_{t+1}) - V_w(S_t)\right)^2\right] = \mathbb{E}\left[\frac{1}{2}\delta^2\right]$$

Under fn. approx. w/TD:

$$\mathbb{E}\left[\frac{1}{2}\left(R_t + \gamma V_w(S_{t+1}) - V_w(S_t)\right)^2\right]$$

$$w \leftarrow w - \alpha \frac{\partial L(w)}{\partial w} = w - \alpha \nabla L(w)$$

$$\frac{\partial L(w)}{\partial w} = \mathbb{E}\left[\left(R_t + \gamma V_w(S_{t+1}) - V_w(S_t)\right) \cdot \frac{\partial}{\partial w}\left(R_t + \gamma V_w(S_{t+1}) - V_w(S_t)\right)\right]$$

$$= \mathbb{E}\left[\underbrace{\left(R_t + \gamma V_w(S_{t+1}) - V_w(S_t)\right)}_{\delta_t}\left[\gamma \frac{\partial V_w(S_{t+1})}{\partial w} - \frac{\partial V_w(S_t)}{\partial w}\right]\right]$$

$$w \leftarrow w - \alpha \delta_t \left(\gamma \frac{\partial V_w(S_{t+1})}{\partial w} - \frac{\partial V_w(S_t)}{\partial w}\right)$$

$$= w + \alpha \delta_t \left(\underbrace{\frac{\partial V_w(S_t)}{\partial w}} - \gamma \frac{\partial V_w(S_{t+1})}{\partial w}\right)$$

direction of greatest increase with changing w:

But, subtracting $\gamma \frac{\partial v_w(S_{t+1})}{\partial w}$ is effectively penalizing that next state — potentially bad since we have not seen rewards from S_{t+1}.

So: TD update:

$$\boxed{\begin{array}{c} w \leftarrow w + \alpha \delta_t \frac{\partial v_w(S_t)}{\partial w} \\ \text{TD update with function approximation.} \end{array}}$$

Gradient update converges more slowly (maybe ok near v^*, but otherwise an issue).

TD f.a. will work w/ linear approx.

$$v_w(s) = w^T \phi(s)$$

$$\frac{\partial v_w(s)}{\partial w} = \phi(s)$$

So: $w \leftarrow w + \alpha \delta_t \phi(S_t)$

In the tabular case:

$$w = \begin{bmatrix} v(s_1) \\ \vdots \\ v(s_n) \end{bmatrix}$$

Update is:

$$v(S_t) \leftarrow v(S_t) + \alpha \delta_t$$

Properties of this f.a.:

- Tabular: Converge a.s. to v^π if α decayed appropriately.
- Tabular: Converges in the mean if α sufficiently small constant
- Linear: Converges to some w_∞ s.t.

$$MSE(w_\infty) \leq \frac{1}{1-\gamma} MSE(w^*)$$

best weight vector for this approx.