Note Title 10/26/2017

230 2017-10-26

TD: $v(S_t) \leftarrow v(S_t) + \alpha(R_t + \gamma v(S_{t+1}) - v(S_t))$ Tabular (one v entry per state s)

$w \leftarrow w + \alpha(R_t + \gamma w^T \phi(S_{t+1}) - w^T \phi(S_t))$ Linear Function Approximation

$w \leftarrow w + \alpha(\underbrace{R_t + \gamma v_w(S_{t+1}) - v_w(S_t)}_{\text{TD error}}) \frac{\partial v_w(S_t)}{\partial w}$ General Function Approximation

Properties: Tabular: converges to v^π a.s. if α decreased properly

Linear: converges to w_∞ s.t. $MSE(w_\infty) \le \frac{1}{1-\gamma} MSE(w^*)$

$\arg\min_w MSE(w)$

General: Can diverge.

	DP	MC	TD
must know P+R	Y	N	N
must wait until episode end	N.A.	Y	N

MC vs. TD

- What makes a better target, G_t or $R_t + \gamma v(S_{t+1})$?
- Each of these is an estimator of $v^\pi(S_t)$
- Mean Squared Error (MSE) is one measure of estimator quality

For random X and $\theta \in \mathbb{R}$,

$$MSE(X) = \mathbb{E}[(X-\theta)^2] = (\mathbb{E}[X]-\theta)^2 + Var(X)$$
$$= bias(X)^2 + Var(X)$$

An optimal estimator balances the bias/variance tradeoff.

$$MSE(G_t) = \underbrace{bias(G_t)^2}_{\downarrow\ 0} + \underbrace{Var(G_t)}_{Var(R_t + \gamma R_{t+1} + \dots)}$$

$$MSE(R_t + \gamma v(S_{t+1})) = \underbrace{bias(R_t + \gamma v(S_{t+1}))^2}_{\text{small when } v \text{ is fairly accurate}} + Var(R_t + \gamma v(S_{t+1}))$$

MC vs TD

$$\hat{P}(s,a,s') = \frac{\#\ (s,a,s')\ \text{transitions}}{\#\ (s,a)\ \text{events}} \qquad \hat{R}(s,a,s') = \text{mean}(r \mid s,a,s')$$

- These estimates of P+R maxi$\underbrace{\text{mize the likeli}}_{\text{ML model of the MDP}}$hood of the observed data
- Given a fixed batch of data – (s, a, r, s') tuples
  - If every state observed one or more times
  - Then TD applied to convergence gives $v^{\pi}_{\hat{P},\hat{R}}$ if $\hat{P} + \hat{R}$ were the transition + reward functions

Sarsa: Using TD for policy improvement/control

Idea: Use TD to estimate q^π & simultaneously change π to be greedy w.r.t to q^π.

$$S_t \xrightarrow{A_t} S_{t+1} \xrightarrow{A_{t+1}} \cdots$$

View states as (s,a) pairs: (S_t, A_t)

$$((S_t, A_t), R_t, (S_{t+1}, A_{t+1})) \Rightarrow \cancel{r}\, q(S_t, A_t)$$

estimates $\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \,\middle|\, S_t = s, A_t = a, \pi\right]$

originally:

$(S_t, R_t, S_{t+1}) \Rightarrow v(S_t)$, which estimates $\mathbb{E}\left[\gamma^k R_{t+k} \,\middle|\, S_t = s, \pi\right]$

Updates:

$$q(s,a) \leftarrow q(s,a) + \alpha(r + \gamma \underbrace{q(s',a')}_{} - q(s,a))$$

TD for policy evaluation. Properties as before.

Can replace with $\sum_{\bar{a}} \pi(s', \bar{a}) \cdot q(s', \bar{a})$ — but not necessary.

General form:

$$w \leftarrow w + \alpha(r + \gamma q_w(s',a') - q_w(s,a)) \frac{\partial q_w(s,a)}{\partial w}$$

Control using TD (Sarsa):

Init: $q(s,a) \leftarrow$ arbitrary

Repeat for each episode

$s \sim d_0$

Choose a from state s using a policy derived from q, such as ϵ-greedy or softmax.

Choose a with prob. $\dfrac{e^{\sigma q(s,a)}}{\sum_a e^{\sigma q(s,a)}}$

with prob $1-\epsilon$ choose an action $a \in \arg\max_u q(s,u)$; if many u's maximize q, choose among them with equal prob. With prob. ϵ select uniformly randomly from the full action set $\mathcal{A}$.

Repeat for each step in the episode:

- Take action a, observe r and s'
- Choose a' using our q policy.
- $q(s,a) \leftarrow q(s,a) + \alpha(r + \gamma q(s',a') - q(s,a))$
- $a \leftarrow a'$, $s \leftarrow s'$

Name Sarsa from (s,a,r,s',a')

Func. Approx. Form:

Init: $w \leftarrow 0$ (or maybe random)

Update: $w \leftarrow w + \alpha(r + \gamma q_w(s',a') - q_w(s,a)) \cdot \dfrac{\partial q_w(s,a)}{\partial w}$

Sarsa properties:

- Converges a.s. to the optimal action value function if:
  1) Tabular
  2) Every (s,a) pair visited infinitely often
  3) Adjust hyperparameters to move toward a greedy policy ($\epsilon \to 0$, $\sigma \to$ large)

  Fight each other. Can use $\epsilon_t = 1/t$ .... But in practice ϵ or σ not adjusted.

  "Greedy in the limit with infinite exploration" = GLIE
- What if $\epsilon = 0$? Won't necessarily see all (s,a) pairs.
  - What about a pessimistic initial value function? (initial q less than actual q) Can easily get stuck on a value that increased a bit.
  - What about an optimistic initial value function? Encourages exploration. Can help even if $\epsilon > 0$.
- Sarsa is "on-policy": Always estimating q for the current policy (the one generating actions).

- Linear func. approx: Converges a.s. (not clear to what).
- Non-linear (general) func. approx: Can diverge.

Q-learning: Off-policy TD control

(s, a, r, s')

Use this update:

$$q(s,a) \leftarrow q(s,a) + \alpha\left(r + \gamma \max_{a'} q(s',a') - q(s,a)\right)$$

$$w \leftarrow w + \alpha\left(r + \gamma \max_{a'} q_w(s',a') - q_w(s,a)\right) \frac{\partial q_w(s,a)}{\partial w}$$