TD prediction

Saturday, April 18, 2020 11:47 PM

Reinforcement Learning

Temporal Difference - Prediction

TD - Introduction

DP

Boot strapping.

Perfect model of the env.

MC

Model free - Samples of episodes.

G_t

TD - Prediction

$Monte - calro\ update\ rule\ (contant\ \alpha\ MC)$:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

New Est ← Old Est + LR [Target − Old Estimate]

$R_{t+1} + \gamma V(S_{t+1})$

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

$TD(0) update\ rule$:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

TD-error

TD(λ)

$S_0\ A_0\ R_1\ S_1 \mid A_1\ R_2\ S_2 \ldots$

TD(0) - Prediction Algorithm

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(\mathit{terminal}) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

SARSA

Sunday, April 19, 2020 1:23 PM

Reinforcement Learning

On-Policy TD Control : SARSA

SARSA - Introduction

$TD(0)$ *update rule for action values:*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

(S, A, R, S, A)

(S, A) R (S, A)

SARSA - Algorithm

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s, a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma Q(S', A') - Q(S, A)\right]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Q-Learning

Sunday, April 19, 2020 1:40 PM

Reinforcement Learning

Off-Policy TD Control : Q-Learning

Q-Learning : Algorithm:

π

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha\big[R + \gamma \max_a Q(S', a) - Q(S, A)\big]$
    $S \leftarrow S'$
  until S is terminal

b

π

Comparison between Q-learning and SARSA:

Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Choose A from S using policy derived from Q (e.g., ϵ-greedy)
  Repeat (for each step of episode):
    Take action A, observe R, S'
    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma Q(S',A') - Q(S,A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)\Big[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a')\Big]$$

$$\Big[R + \gamma E_\pi[Q(S_{t+1}, A_{t+1}) | S_{t+1}]\Big]$$

$$\Big(\sum_a \pi(a|S_{t+1}) \cdot Q(S_{t+1}, a)\Big)$$

Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$
Initialize $Q(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(\textit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
    $S \leftarrow S'$
  until S is terminal

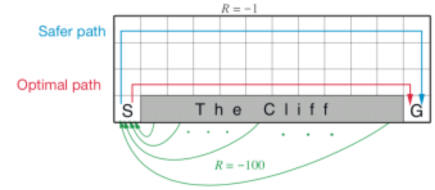
$$q_*(s,a) = \sum_{s',r} p(s',r|s,a)\big[r + \gamma \max_a q_*(s',a')\big]$$

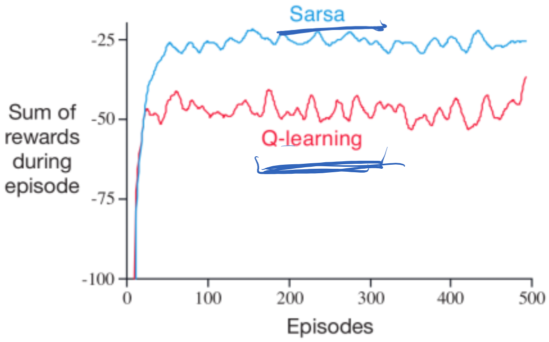
but greedy

$\pi \to$ greedy.

SARSA and Q-Learning on Cliff-Walking task:

1. Undiscounted, episodic task.
2. Task is to move from start state (S) to goal state(G)
3. Actions : Up, Down, Left and Right.
4. Deterministic environment.
5. Reward of -1 on all transitions, except those into the region marked as "The Cliff". Stepping into this region incurs a reward of -100 and sends the agent instantly to the start state.





Action selected acording to $\epsilon - greedy$ ($\epsilon = 0.1$)

Double Q-Learning

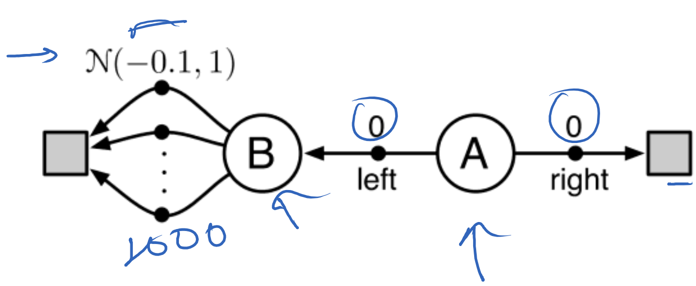
Sunday, April 19, 2020 6:39 PM

Reinforcement Learning
Double Q-Learning

Maximization Bias:

$SARSA: \quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$

$Q - learning: \quad Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$



0.5

1000

$A_1 \rightarrow -0.2$

$A_1 \rightarrow -0.1$

$A_1 \rightarrow +0.4$

$A_1 \rightarrow 0.5$

Double Learning:

$Q \rightarrow$ Best action

$\rightarrow$ Estimated value for that action.

$A^* = \text{argmax}_a\, Q_1(a)$

$Q_2(A^*)$

$q_1(a) \rightarrow Q_1(a)$

$Q_2(a)$

$Q_1(S, A_1)$

$Q_2(S_1, A_1)$

$$Q_1(S_t, A_t) \leftarrow Q_1(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q_2\left(S_{t+1}, argmax_a Q_1(S_{t+1}, a)\right) - Q_1(S_t, A_t)\right]$$

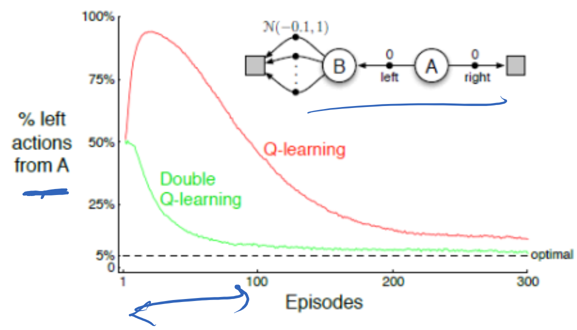
$$Q_2(S_t, A_t) \leftarrow Q_2(S_t, A_t) + \alpha[R_{t+1} + \gamma Q_1\left(S_{t+1}, argmax_a Q_2(S_{t+1}, a)\right) - Q_2(S_t, A_t)]$$

Double Q-Learning Algorithm:

Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$

Algorithm parameters: step size $\alpha \in (0,1]$, small $\varepsilon > 0$
Initialize $Q_1(s,a)$ and $Q_2(s,a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using the policy ε-greedy in $Q_1 + Q_2$
    Take action A, observe R, S'
    With 0.5 probabilility:
      $Q_1(S,A) \leftarrow Q_1(S,A) + \alpha\Big(R + \gamma Q_2\big(S', \arg\max_a Q_1(S',a)\big) - Q_1(S,A)\Big)$
    else:
      $Q_2(S,A) \leftarrow Q_2(S,A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S',a)\big) - Q_2(S,A)\Big)$
    $S \leftarrow S'$
  until S is terminal



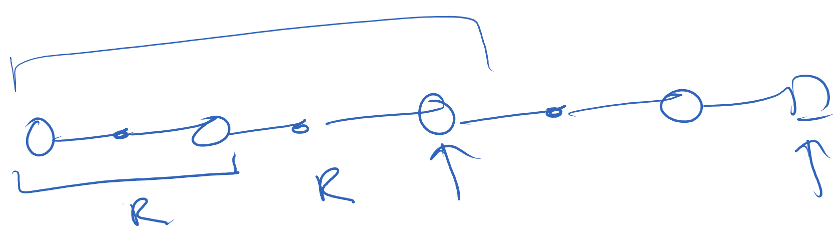
n-step TD and TD(λ)

Saturday, April 25, 2020 7:15 PM

Reinforcement Learning

n-step TD and TD(λ)

n-step Temporal Difference Learning:

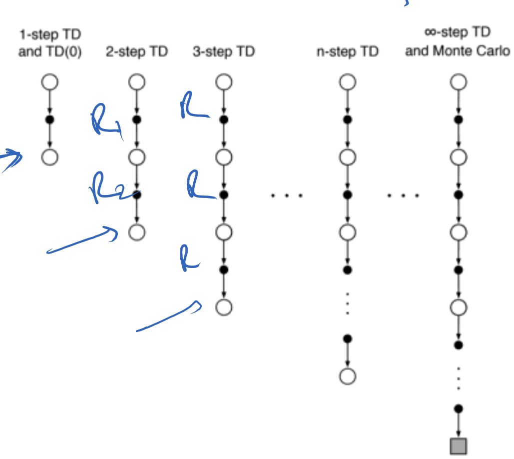


$Monte\ carlo\ update\ rule:$

$$V(s_t) = V(s_t) + \alpha[G_t - V(s_t)]$$

$TD(0)\ (one - step\ TD)\ update\ rule:$

$$V(s_t) = V(s_t) + \alpha[R_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$



TD(0)

$$\left[0.5 \times G_t^{(2)} + 0.5 \times G_t^{(5)}\right]$$

TD(0)

$0.5 \times 4 \times$

$n - step\ returns\ for\ n = 1,2,\ldots\infty$:

$n = 1 \quad : \quad G_t^{(1)} = R_{t+1} + \gamma V(s_{t+1})$

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

.

.

.

$n = \infty \quad : \quad G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-1} R_T$

$\to$ MC

$General\ n - step\ return:$

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

$n - step\ TD\ update:$

$$V(s_t) = V(s_t) + \alpha[G_t^{(n)} - V(s_t)]$$

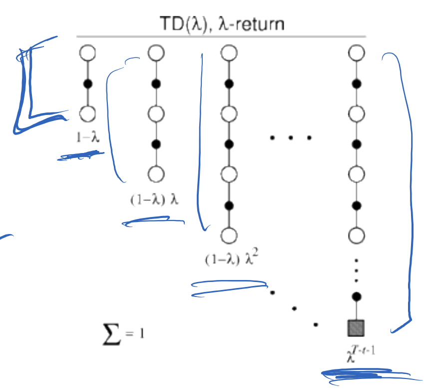
$\lambda \in [0,1]$

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

TD(λ):

$Using\ a\ weight\ of\ (1-\lambda)\lambda^{n-1}\ for\ n^{th}\ return:$

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



$TD(\lambda)\ update\ rule:$

$$V(s_t) = V(s_t) + \alpha[G_t^\lambda - V(s_t)]$$

$(1-\lambda)\lambda$

$(1-\lambda)\lambda^2$

λ^∞

$1-\lambda + (1-\lambda)\lambda + X = 1$

$1 - \lambda + \lambda - \lambda^2 + X = 1$

$\Rightarrow X = \lambda^2$