Wednesday, October 21, 2020

Lecture 18

CS 59000 - RL

MDPs

- Learning
  - Q-Learning
  - SARSA
  - Temporal Difference.

---

change notation : γ for discount factor.

$$Q^*(x,a) = \bar{r}(x,a) + \gamma \sum_{x'} P(x'/x,a) \max_{a'} Q^*(x',a')$$

$$= E^{\pi^*}\left[\sum_{t>0} \gamma^t r_t \mid x,a\right]$$

Look at $\sum_{t>0} \gamma^{t-1} r_t$ $\Rightarrow$ Monte Carlo estimate of $Q^*(x,a)$

It is also unbiased

$$Q^*(x,a) = E^{\pi^*}\left[r_1 + \gamma \max_{a'} Q^*(X_1,a')\right]$$

$$\rightarrow r_1 + \gamma \max_{a'} Q^*(X_1,a')$$

$$Q_t \rightarrow Q_{t+1}(x,a) \leftarrow r_t + \gamma \max_{a'} Q_t(X_{t+1},a')$$

Q-learning:

- Initialize Q_0

1) look at a tuple (X_t, A_t, r_t, X_{t+1})

2) $Q_{t+1}(X_t, A_t)$

$$= \left(1 - \alpha_t(X_t, A_t)\right) Q_t(X_t, A_t) + \alpha(X_t, A_t)\left(r_t + \gamma \max_{a'} Q_t(X_{t+1}, a')\right)$$

Learning rate

-> This algorithm works when we visit all the state and action pairs frequently enough.

Def: An MDP is called strongly connected or communicating if for any pair of $x, x' \in \mathcal{X}$, there exists a policy π, such that, when starting

from x, following $\pi \in \Pi^{MD}$, there is a positive probability of reaching x' in finite time.

If the MDP is not strongly connected, there is no hope in general learning.

Your agent might get stuck in a some bad subset of states, and stays there for ever.

Def: The diameter of an MDP M is

$$D(M) = \max_{x' \neq x} \min_{\pi \in \Pi^{MD}} E^{\pi}\left[\min\{t \geq 1 : X_t = x' \mid X_1 = x\}\right] - 1$$

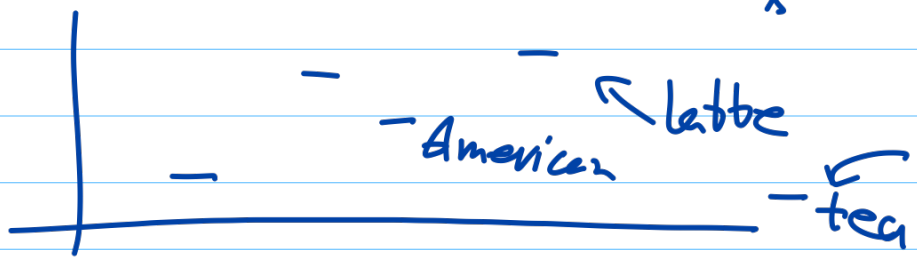
For a finite state MDP M, $D(M) < \infty$ if and only if it is strongly communicating.

Consider the following epsilon-greed (ε-greedy) exploration-exploitation algorithm:

Initialize Q_0

- At time t
  - Choose a $a_t^* \in \underset{a}{\operatorname{argmax}}\, Q_t(X_t, a)$
  - Follow ε-greedy policy Π_t

$$\Pi_t = \begin{cases} \frac{\varepsilon}{|A|} & a = a_t^* \\ 1-\varepsilon_t \frac{\varepsilon}{|A|} & \text{a.w} \end{cases}$$



- Choose $A_t \sim \Pi_t(\cdot | X_t) = \Pi_t(X_t)$
- Apply A_t and observe r_t, X_{t+1}

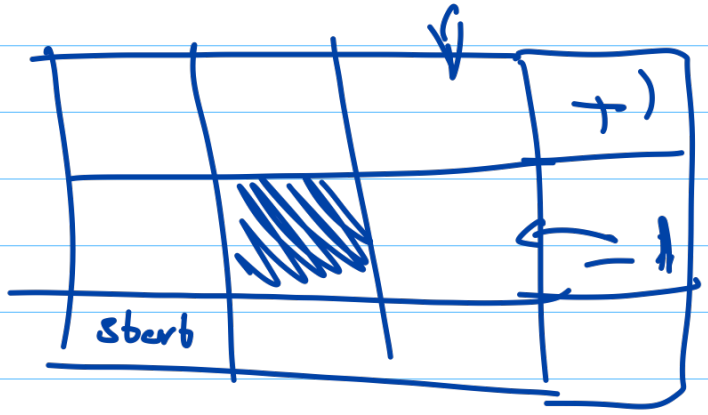
Q update: $Q_{t+1} = Q_t$

$$\underbrace{Q_{t+1}(X_t, A_t)} = \left(1 - \alpha_t(X_t, A_t)\right) Q_t(X_t, A_t) + \alpha_t(X_t, A_t)\left(r_t + \gamma \max_{a'} Q_t(X_{t+1}, a')\right)$$

$$= Q_t(X_t, A_t) + \alpha_t(X_t, A_t)\left(r_t + \gamma \max_{a'} Q(X_{t+1}, a') - Q(X_t, A_t)\right)$$

Wednesday, October 21, 2020

Consider this grid world



If the agent chooses →

it goes → with probability 0.8

and ↑ ↓ each with probability 0.1

If → hits a wall, the agent stays.

SARSA (State Action Reward State Action)

Follow e.g. ε-greedy policy

observe $X_t, A_t, R_t, X_{t+1}, A_{t+1}$

- $Q_{t+1} = Q_t$

$$Q_{t+1}(X_t, A_t) = Q_t(X_t, A_t) + \alpha_t(X_t, A_t)\left(r_t + \gamma Q_t(X_{t+1}, A_{t+1}) - Q_t(X_t, A_t)\right)$$

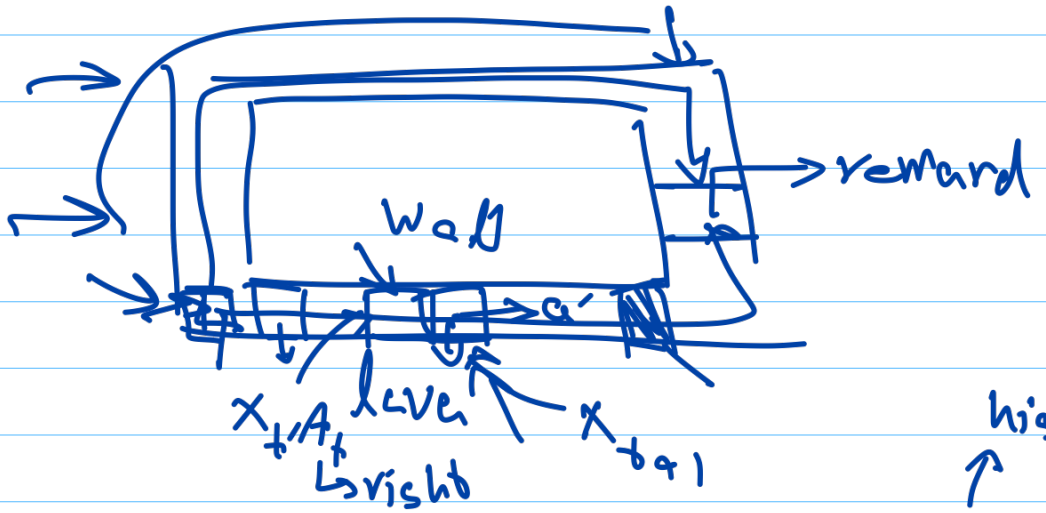
Q learning: $\max_{a'} Q_t(X_{t+1}, a')$

On Policy vs off policy

Update based on the data of the current policy

update based on data of some other policy

(SARSA) (Q-Learning)



high value

$r_t + \max_{a'} Q_t(X_{t+1}, a')$

a' Low

$Q(X_t, A_t) \leftarrow r_t + Q_t(X_{t+1}, A_{t+1})$

Wednesday, October 21, 2020

Temporal Difference (TD)

TD(0)

For a policy π

- $V_{t+1} = V_t$

$$V_{t+1}(X_t) = V_t(X_t) + \alpha_t(X_t)\left(r_t + \gamma V_t(X_{t+1}) - V_t(X_t)\right)$$

---

similarly for Q. (on-policy)

- $Q_{t+1} = Q_t$

$$Q_{t+1}(X_t, A_t) = Q_t(X_t, A_t) + \alpha_t(X_t, A_t)\left(r_t + \gamma Q_t(X_{t+1}, A_{t+1}) - Q_t(X_t, A_t)\right)$$

(write down the off policy version)

Back to V^π.

starting from $X_t = x \rightarrow \sum_{k=1} \gamma^{k-1} r_{t+k-1}$ → high variance

is an unbiased estimate of V^π.

Instead we use $r_t + \gamma V_t^\pi(X_{t+1})$ → biased, low variance

can we make a trade-off?

Wednesday, October 21, 2020

Var ↓ bias ↑ $\rightarrow R_t = r_t + \gamma V_t(X_{t+1})$

$R_t^2 = r_t + \gamma r_{t+1} + \gamma^2 V_t^{\pi}(X_{t+2})$

⋮

variance ↑ bias ↓ $\rightarrow R_t^n = r_t + \gamma r_{t+1} + \cdots + \gamma^n V_t^{\pi}(X_{t+n})$

It is called n-step TD.

How about using

$$R_t^{\lambda} = (1-\lambda)\sum_{i>0} \lambda^{i-1} R_t^i$$

$\rightarrow TD(\lambda)$

$$V_{t+1}(X_t) = V_t(X_t) + \alpha(X_t)\left(R_t^{\lambda} - V_t(X_t)\right)$$

And so many other stochastic approximation methods.

Q-learning: Watkins 1989

TD(λ): Sutton 1988