AML, 5 Nov 2019

Monte Carlo methods to evaluate v_π

→ Temporal - Difference methods (TD)

Monte Carlo

Generate an episode wrt π

Calculate state values for the episode, right to left

Add new values to list $V(s)$ for each s

Finally take mean of each $V(s)$ list — Initial estimates of $V(s)$ are discarded

TD:

"Bootstrapping" – like DP – requires full model

Uses current estimates of $V(S_t)$ to update $V(S_{t+1})$

TD – hybrid of DP & MC

If we were to do MC incrementally

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

α ↑ learning rate

G_t ← global value calculated for entire episode

$G_t - V(S_t)$: observed error / deviation

Replace G_t by $\underbrace{R_{t+1}}_{\text{observed reward}} + \gamma \underbrace{V(S_{t+1})}_{\text{old estimate}}$

TD update rule

$$V(S_t) \leftarrow V(S_t) + \alpha \left[\underbrace{\left(R_{t+1} + \gamma V(S_{t+1})\right) - V(S_t)}_{\text{error}} \right]$$

(α: learning rate)

TD[0] - zero lookahead

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 $V(terminal) = 0$

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

$$v_\pi(s) = E_\pi\left[G_t \mid S_t = s\right]$$ Monte Carlo

$$R_{t+1} + \gamma G_{t+1}$$

$$v_\pi(S_{t+1})$$

$$E_\pi\left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s\right]$$ TD[0]

TD-error $$\delta_t \stackrel{def}{=} \left(R_{t+1} + \gamma V(S_{t+1})\right) - V(S_t)$$

MC-error $$G_t - V(S_t) = R_{t+1} + \gamma G_{t+1} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1})$$

MC-error

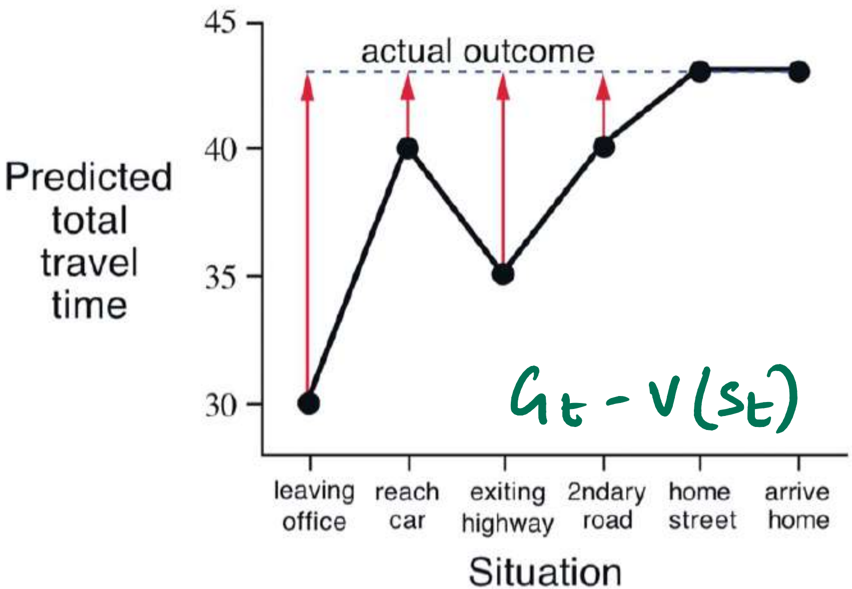
$$G_t - V(S_t) = R_{t+1} + \gamma G_{t+1} - V(S_t) + \gamma V(S_{t+1}) - \gamma V(S_{t+1})$$

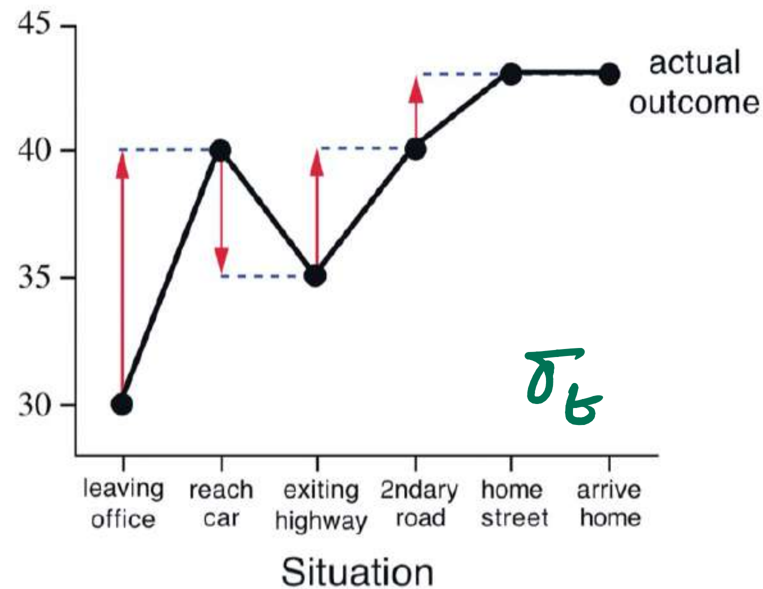
$$= \underbrace{\left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right)}_{\delta_t} + \underbrace{\gamma G_{t+1} - \gamma V(S_{t+1})}_{\gamma(G_{t+1} - V(S_{t+1}))}$$

$$= \delta_t + \gamma \cdot \delta_{t+1} + \gamma^2 \delta_{t+2} \cdots$$

$$= \sum_{k=t}^{T-1} \gamma^{k-1} \delta_k$$

State	*Elapsed Time (minutes)*	*Predicted Time to Go*	*Predicted Total Time*
leaving office, friday at 6	0	30	30
reach car, raining	5	35	40
exiting highway	20	15	35
2ndary road, behind truck	30	10	40
entering home street	40	3	43
arrive home	43	0	43





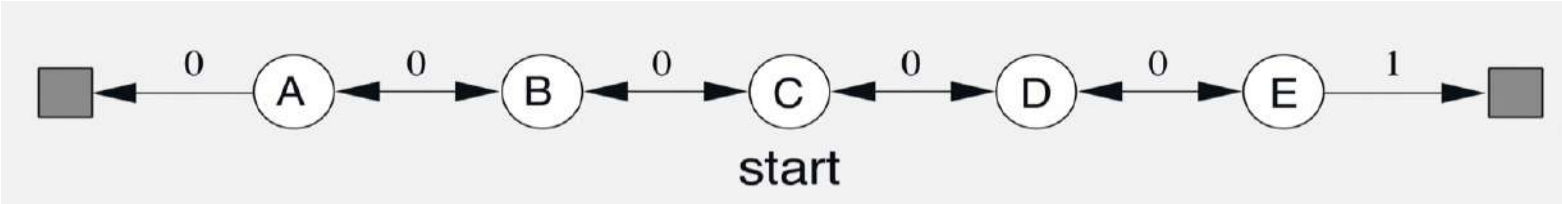
Advantage of TD vs MC

- Incremental update of V
- No need to wait till episode ends
    - May not even have finite episodes

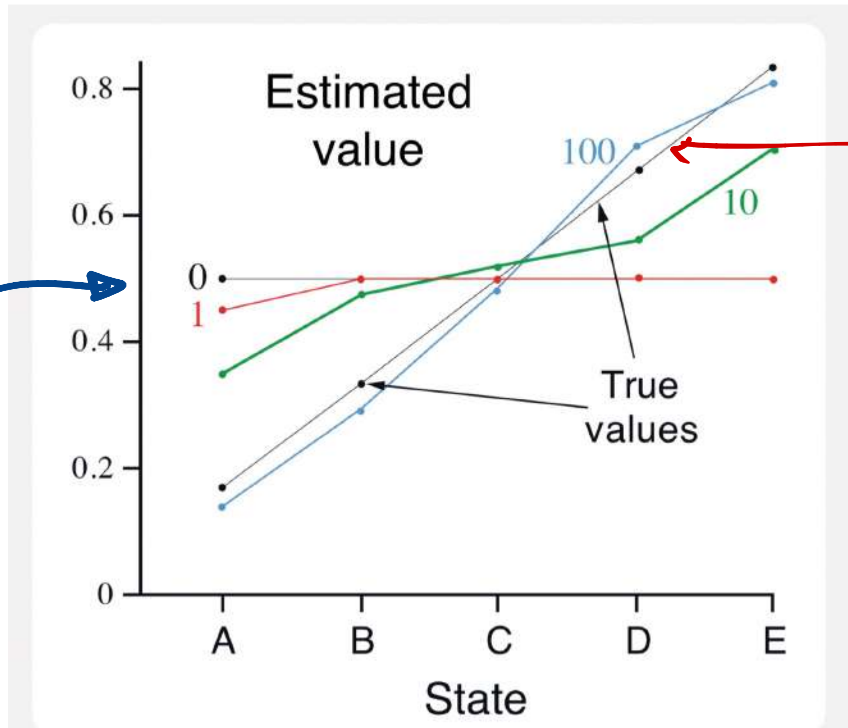
Both TD & MC converge asymptotically to correct values

Which is faster?

- Not been proven
- In practice TD appears faster

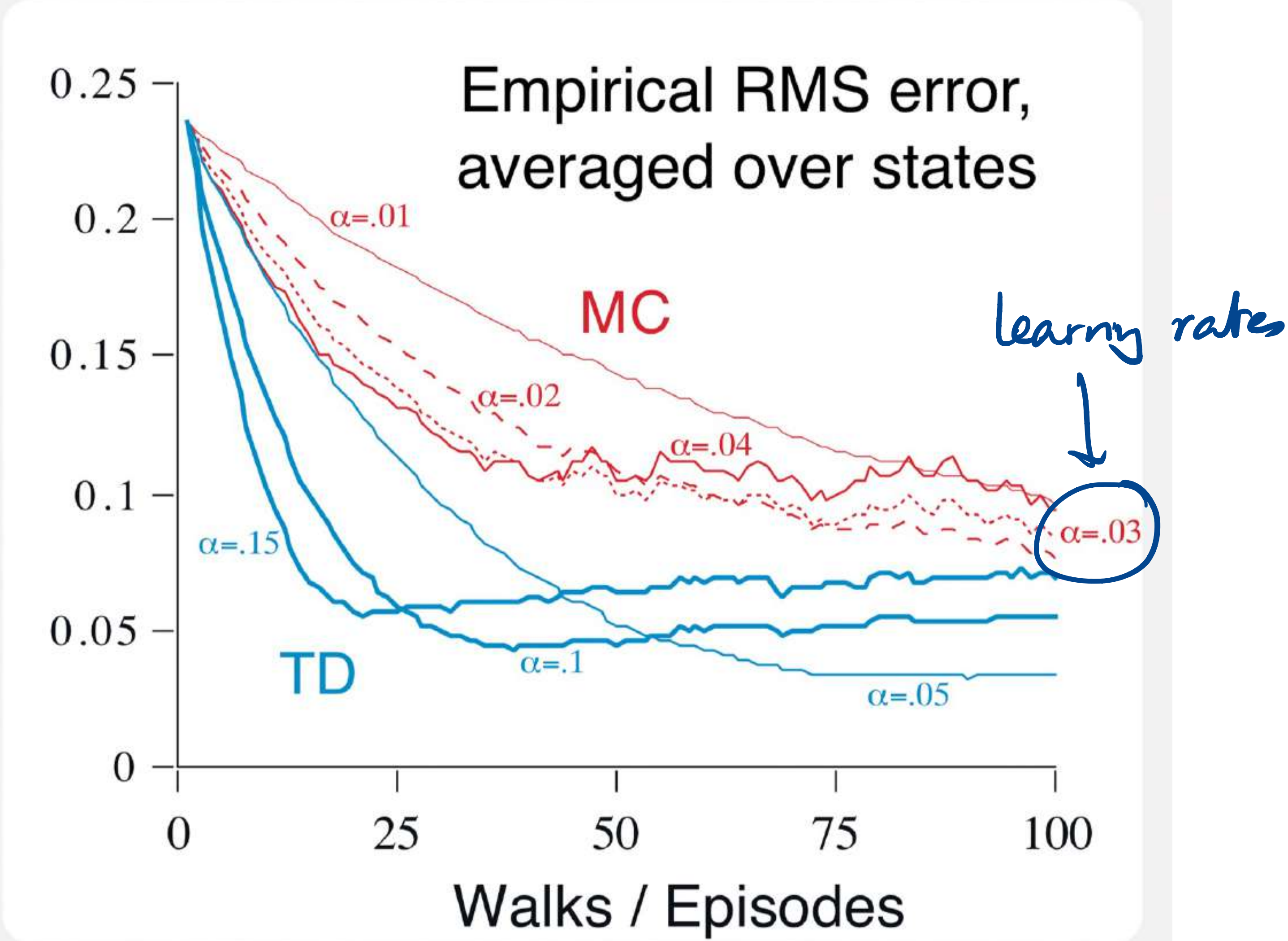


1/6 2/6 V = 1/2 4/6 5/6



TD[0] starting with V=0.5 everywhere

theoretical value


Empirical RMS error, averaged over states
0.25
0.2
0.15
0.1
0.05
0
0
25
50
75
100
Walks / Episodes
MC
TD
α=.01
α=.02
α=.04
α=.03
α=.15
α=.1
α=.05
learning rates

Batch learning

Consider entire set of episodes as a single batch update

MC & TD give different answers

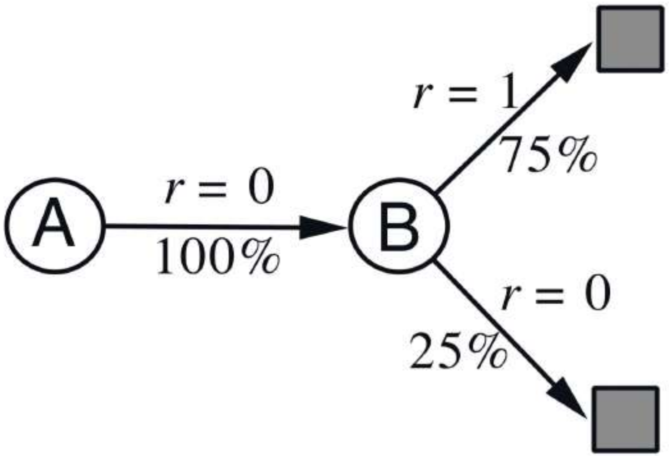
Example 8 observations

A, 0, B, 0	B, 1
B, 1	B, 1
B, 1	B, 1
B, 1	B, 0

$V(B) = \frac{6}{8} = \frac{3}{4}$

What is $V(A)$?

- MC sets $V(A) = 0$
- TD sets $V(A) = V(B)$



MDP learned by TD[0]

MC — Best estimate wrt Mean Square Error of observations

TD[0] - Best estimate wrt. MLE

From value estimation to policy iteration

MC : On policy — Same π generates runs & gets update

Off policy — runs are generated independent of π to be updated

On policy TD

Typically switch from estimating $v_\pi(s)$ to $q_\pi(s,a)$

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

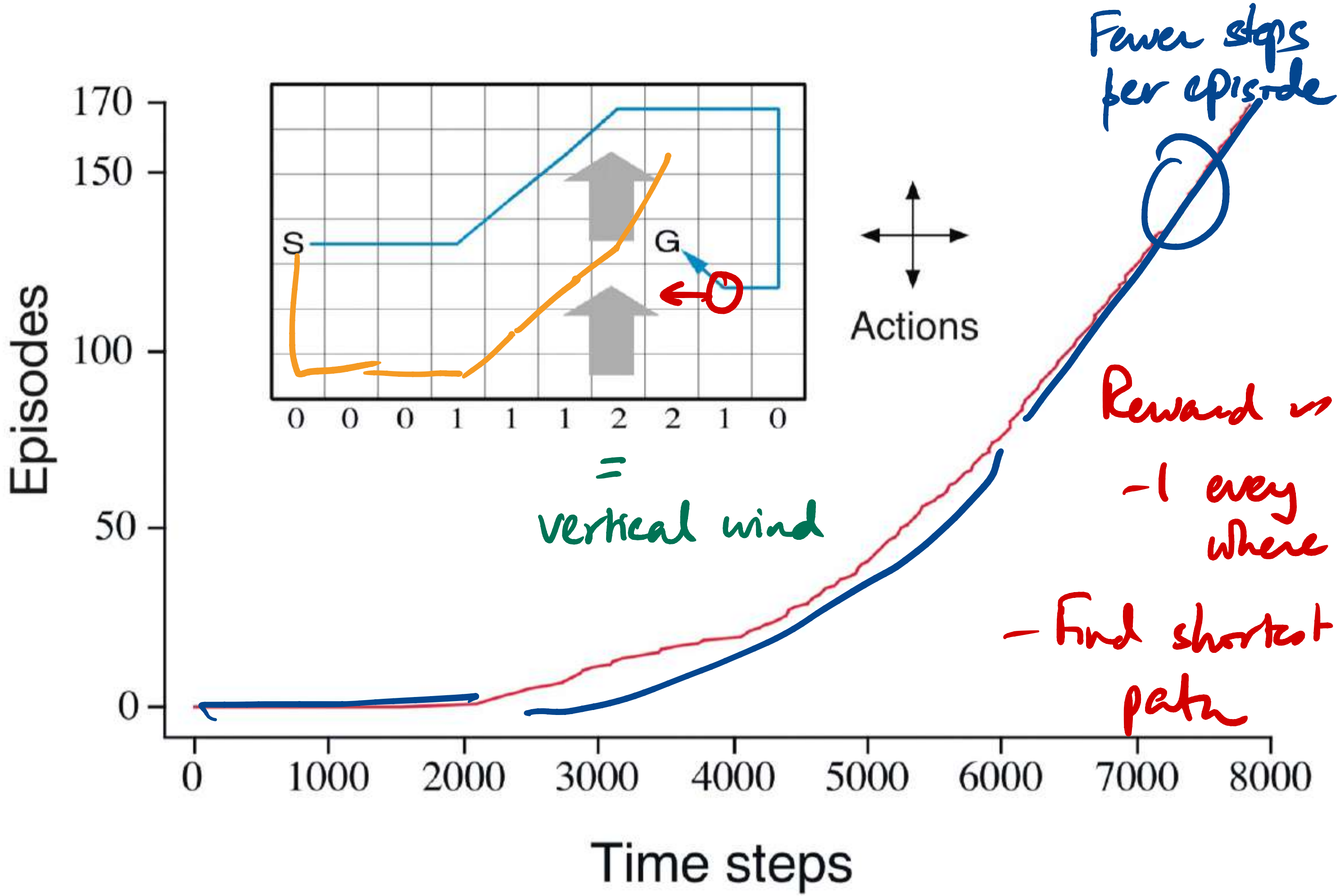
Update is a function of $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}) \Rightarrow$ SARSA

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

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
  Choose A from S using policy derived from Q (e.g., ε-greedy) $Q \rightarrow \pi$
  Loop for each step of episode:
    Take action A, observe R, S' on policy
    Choose A' from S' using policy derived from Q (e.g., ε-greedy)   Update π
    $Q(S, A) \leftarrow Q(S, A) + \alpha \left[R + \gamma Q(S', A') - Q(S, A)\right]$ Update Q
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

Update $Q(S_t, A_t)$ $\longrightarrow$ Update π


Episodes
170
150
100
50
0
0
1000
2000
3000
4000
5000
6000
7000
8000
Time steps
S
G
0 0 0 1 1 1 2 2 1 0
Actions
= vertical wind
Fewer steps per episode
Reward is -1 everywhere
- Find shortest path

<u>Off Policy</u>   Q-learning

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

next action is best
wrt Q, not
determined by 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

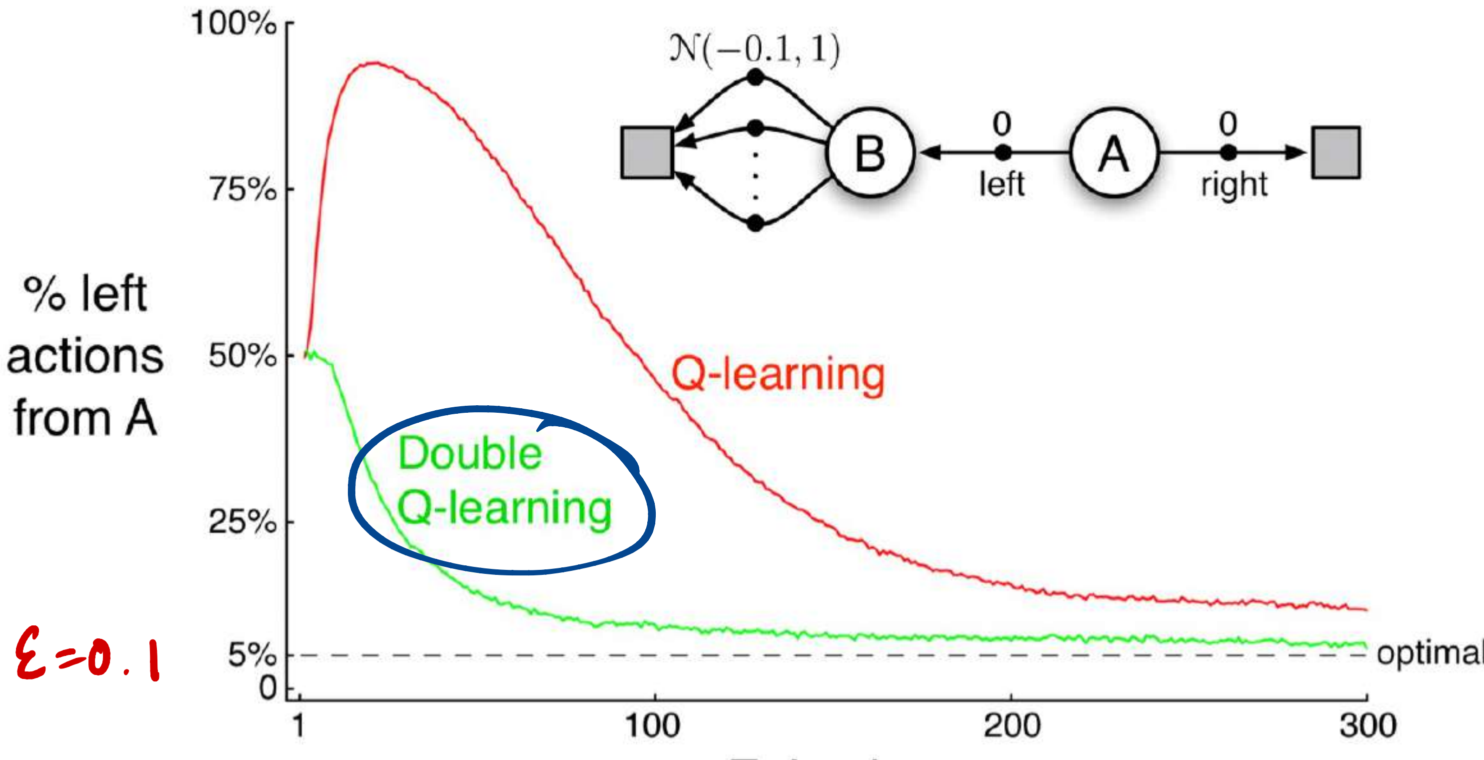
Variant of SARSA – Expected SARSA

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

$$\sum_a \pi(a \mid S_{t+1}) \, Q(S_{t+1}, a)$$

Maximization Bias

- tends to produce inflated estimates



$\varepsilon = 0.1$

Double learning – simultaneously create 2 estimates, randomly use one to update other.

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