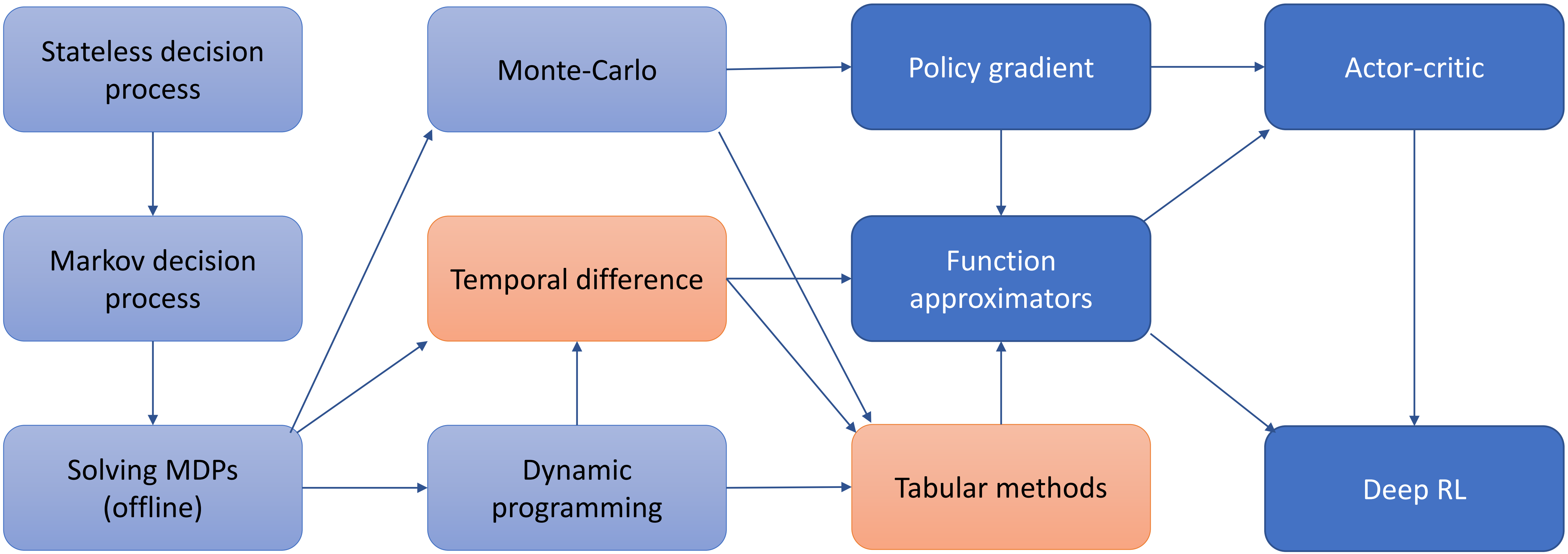
CSCE-642 Reinforcement Learning
Chapter 6: Temporal-Difference Learning

Instructor: Guni Sharon

CSCE-689, Reinforcement Learning



Solving MDPs so far

Dynamic programming

- ✓ Off policy
- ✓ local learning, propagating values from neighbors (Bootstrapping)
- ✗ Model based

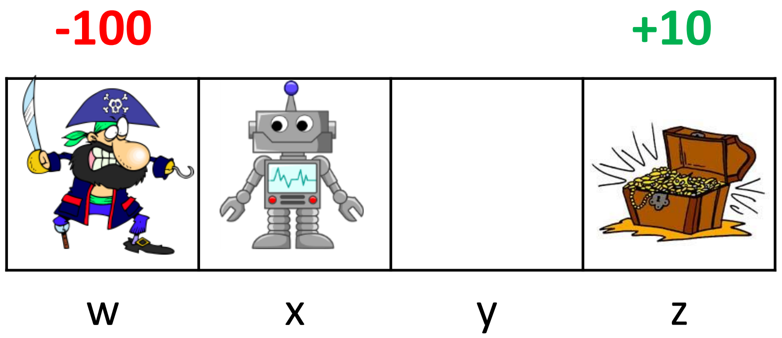
Monte-Carlo

- ✗ On-policy (though important sampling can be used)
- ✗ Requires a full episode to train on
- ✓ Model free, online learning

- $Q(z, exit) = 10$
- $Q(y, \rightarrow) = 0 + \gamma \max_a Q(z, a)$
- $Q(x, \rightarrow) = 0 + \gamma \max_a Q(y, a)$

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

$\gamma = 0.9$



- Episode = $\{x, y, z, exit\}$
- $Q(z, exit) = 10$
- $Q(y, \rightarrow) = 9$
- $Q(x, \rightarrow) = 8.1$

Fuse DP and MC

Dynamic programming

- ✓ Off policy
- ✓ local learning, propagating values from neighbors (Bootstrapping)
- ✗ Model based

Monte-Carlo

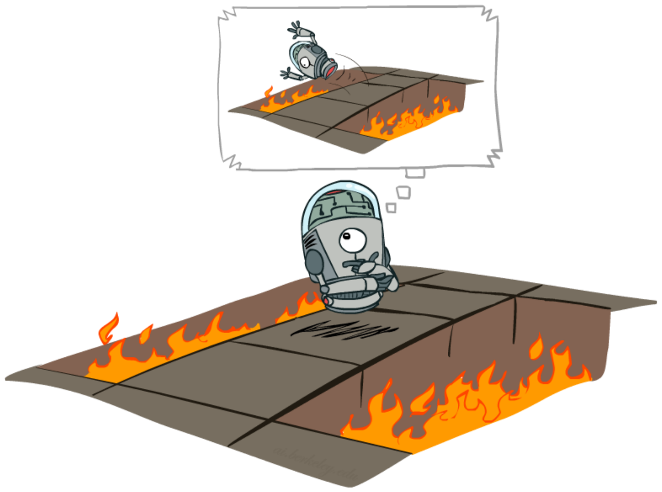
- ✗ On-policy (though important sampling can be used)
- ✗ Requires a full episode to train on
- ✓ Model free, online learning

TD Learning

- ✓ Off policy
- ✓ local learning, propagating values from neighbors (Bootsraping)
- ✓ Model free, online learning

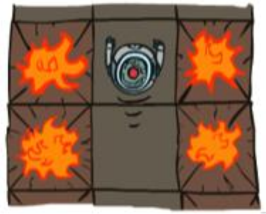
Online Bellman update

- $q^*(s,a) = \sum_{s'} {\color{red}P(s'|s,a)}\,({\color{red}R(s,a,s')} + \gamma \max_{a'}[q^*(s',a')])$
- Model and reward function are unknown
- Instead, we observe transitions: $< s_t, a_t, r_{t+1}, s_{t+1} >$

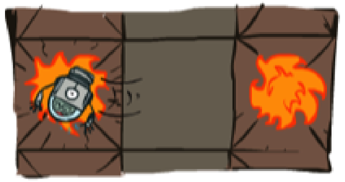


Offline Solution

$$< s \qquad a \qquad r \qquad s' >$$



$$\leftarrow \qquad -100$$



Online Learning

Online Bellman update

- $q^*(s,a) = \sum_{s'} \boxed{P(s'|s,a)} \left(R(s,a,s') + \gamma \max_{a'}[q^*(s',a')]\right)$
- Model and reward function are unknown
- Instead, we observe transitions: $< s_t, a_t, r_{t+1}, s_{t+1} >$
  - $\mathbb{E}[X] = \sum_{k=1}^{|X|} x_i \Pr\{x_i\} = \frac{\sum_{k=1}^{N} x_k}{N}$
- $q^*(s,a) = \mathbb{E}_{s' \sim P(s'|s,a)} \left[\mathbb{E}[R(s,a,s')] + \gamma \max_{a'}[q^*(s',a')]\right]$
- For a single sample: $= r_{t+1} + \gamma \max_{a'}[q^*(s_{t+1},a')]$
- = unbiased estimation of the Bellman update

Online Bellman update

- $q^*(s,a) = r_{t+1} + \gamma \max_{a'}[q^*(s_{t+1}, a')]$
- Assume that we visited state s and performed action a 5 times
  - $< s, a, 3, x >, < s, a, 5, x >, < s, a, 1, y >, < s, a, 2, y >, < s, a, 3, y >$
  - Assume $\gamma = 0.9$ and $\max_{a}[q^*(x,a)] = \max_{a}[q^*(y,a)] = 10$
- Compute an approximation of $q^*(s,a)$:
  - Average of unbiased estimations
  - $\left((3+9)+(5+9)+(1+9)+(2+9)+(3+9)\right)\frac{1}{5}$
- Can $Q(s,a)$ be updated online?
- Yes, moving average:
  - $Q(s,a) = Q(s,a) + \alpha\left(r_{t+1} + \gamma \max_{a'}[q^*(s', a')] - Q(s,a)\right)$

Temporal difference learning

- $Q(s,a) = Q(s,a) + \alpha\left(R_{t+1} + \gamma \max_{a'}[q^*(s',a')] - Q(s,a)\right)$
- But we don’t know $q^*(s',a)$
- Use the estimation $Q(s',a)$
- $Q(s,a) = Q(s,a) + \alpha\left(\underbrace{R_{t+1} + \gamma \max_{a'}[Q(s',a')] - Q(s,a)}_{\text{Temporal difference error: }\delta}\right)$

Temporal difference learning

- $Q(s,a) = Q(s,a) + \alpha\left(R_{t+1} + \gamma \max_{a'}[Q(s',a')] - Q(s,a)\right)$
  - $V(s) = V(s) + \alpha\big(R_{t+1} + \gamma V(s') - V(s)\big)$
- Update estimate based on other estimates
- Model free
- Online, incremental learning
- Guaranteed to converge to the true value!
  - Some conditions on the step size, α (see slide #19 in 2Multi_armed_bandits.pptx)
- Usually converges faster than MC methods

SARSA: On-policy TD Control

- $Q(s,a) = Q(s,a) + \alpha\left(R_{t+1} + \gamma \max_{a'}[Q(s',a')] - Q(s,a)\right)$
- Replace $\max_{a'}[Q(s',a')]$ with values from the observed transition
  - $< s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1}, >$
- $Q(s_t,a_t) = Q(s_t,a_t) + \alpha\big(r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)\big)$
- SARSA converges with probability 1 to an optimal policy and action-values as long as all state–action pairs are visited infinitely often, and the policy converges in the limit to the greedy policy
  - Which can be arranged, for example, with ε-greedy policies by setting ε= 1/t

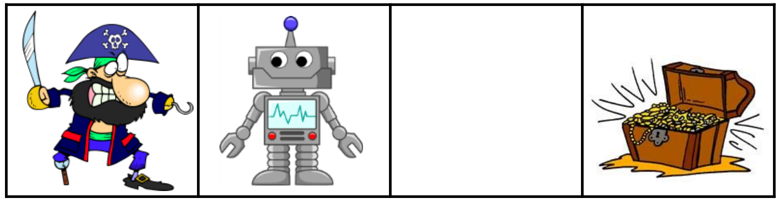
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

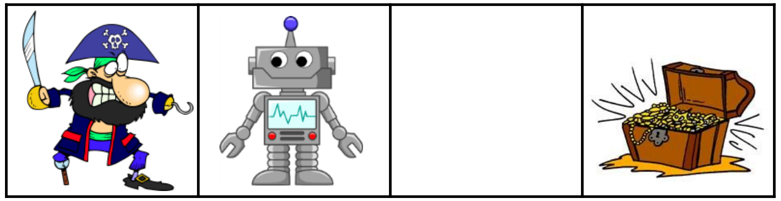
SARSA: On-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

$Q =$

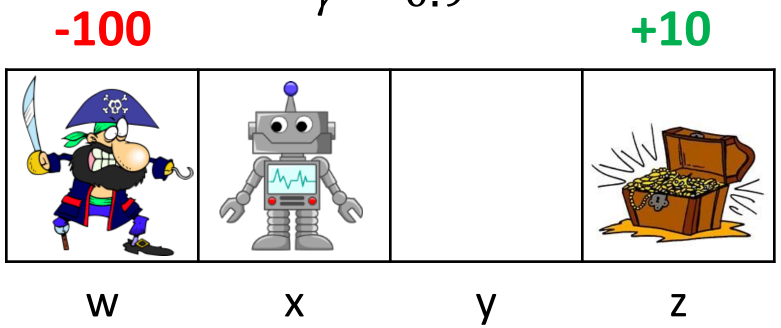
0	0,0	0,0	0
w	x	y	z

SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$



x	$\leftarrow$	**0**	w	$null$
S	A	R	S'	A'

$Q =$

0	**0,0**	**0,0**	**0**
w	x	y	z

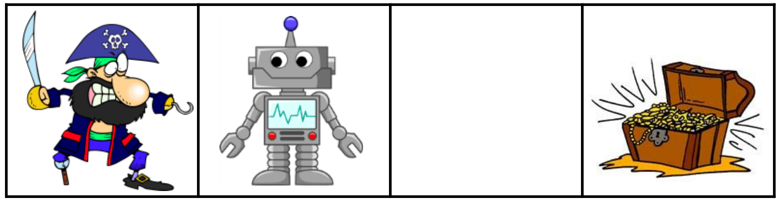
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

S	A	R	S'	A'
x	$\leftarrow$	0	w	$exit$

$Q =$

w	x	y	z
0	0,0	0,0	0

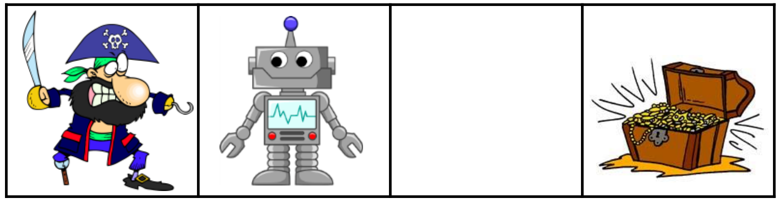
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

x	$\leftarrow$	0	w	$exit$
S	A	R	S'	A'

$Q =$

0	0,0	0,0	0
w	x	y	z

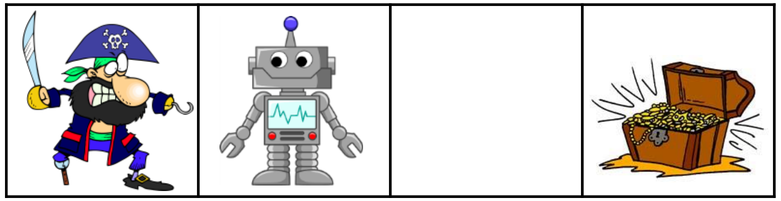
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

w	$exit$	0	w	$exit$
S	A	R	S’	A’

$Q =$

0	0,0	0,0	0
w	x	y	z

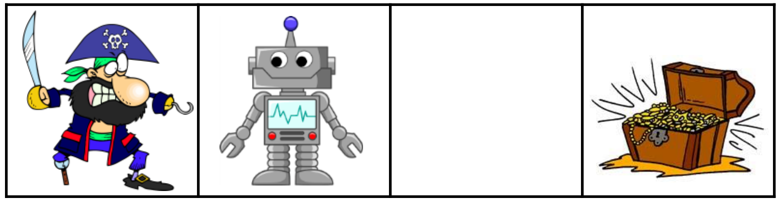
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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)
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \underbrace{\cancel{\gamma Q(S',A')}}_{0} - Q(S,A)\big]$
    $S \leftarrow S'$; $A \leftarrow A'$;
  until S is terminal

$\gamma = 0.9$

-100 **+10**



w x y z

S	A	R	S'	A'
w	$exit$	-100	ter	$exit$

$Q =$

w	x	y	z
0	0,0	0,0	0

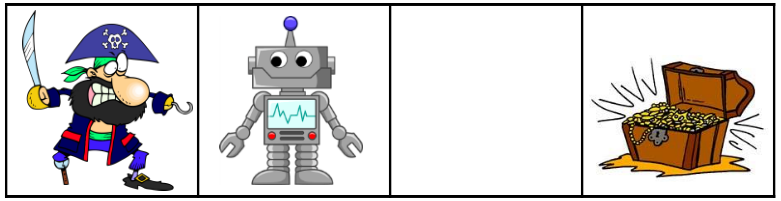
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

w	$exit$	-100	ter	$\cdot$
S	A	R	S’	A’

$Q =$

-100	0,0	0,0	0
w	x	y	z

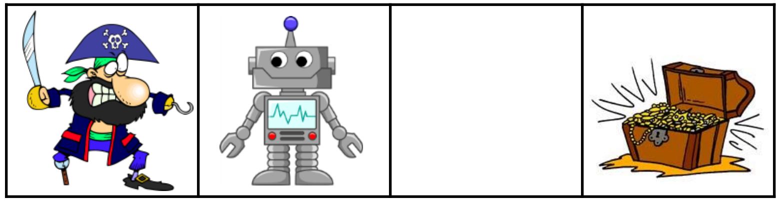
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

x	$\leftarrow$	-100	ter	$\cdot$
S	A	R	S’	A’

$Q =$

-100	0,0	0,0	0
w	x	y	z

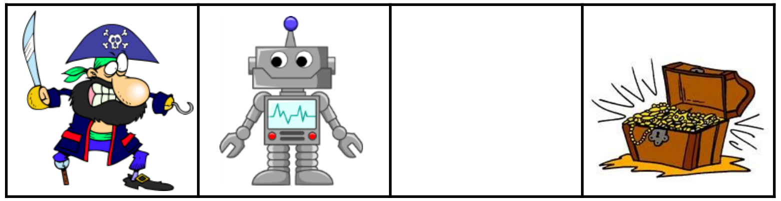
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

x	$\leftarrow$	0	w	$exit$
S	A	R	S'	A'

$Q =$

-100	0,0	0,0	0
w	x	y	z

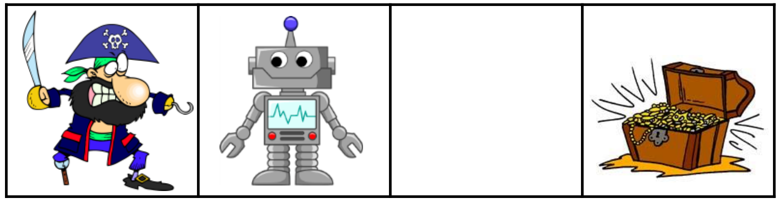
SARSA: On-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

$\gamma = 0.9$

-100 +10



w x y z

x	$\leftarrow$	0	w	$exit$
S	A	R	S'	A'

$Q =$

-100	-90,0	0,0	0
w	x	y	z

SARSA: On-policy TD Control

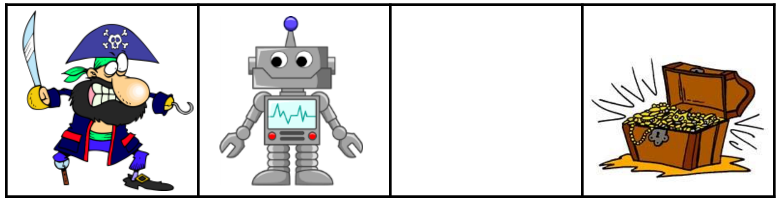
Sarsa (on-policy TD control) for estimating $Q \approx q_*$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, 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

And so on…

$\gamma = 0.9$

-100 +10



w x y z

w	$exit$	0	w	$exit$
S	A	R	S'	A'

$Q =$

-100	-90,0	0,0	0
w	x	y	z

Q-learning: Off-policy TD Control

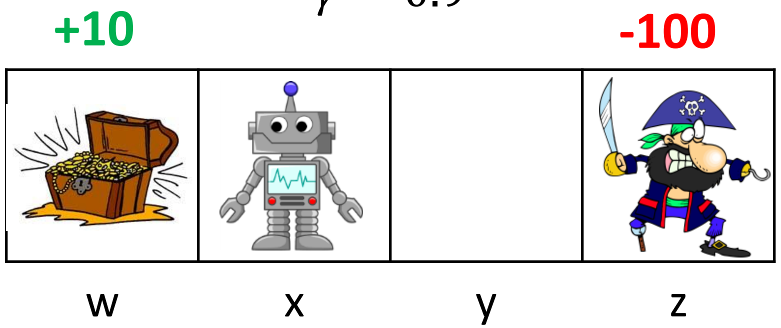
- Use the original TD update rule
- $Q(s,a) = Q(s,a) + \alpha\left(R_{t+1} + \gamma \max_{a'}[Q(s',a')] - Q(s,a)\right)$
- Converge on the state-action value for the optimal policy, i.e., q^*
  - Assuming that every state-action pair is visited infinitely often
- Follows from the proof of convergence for the Bellman function
  - See slides #25,26 in "3MDPs+DP.pptx"

Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



S	A	R	S'
x	$null$	$null$	$null$

$Q =$

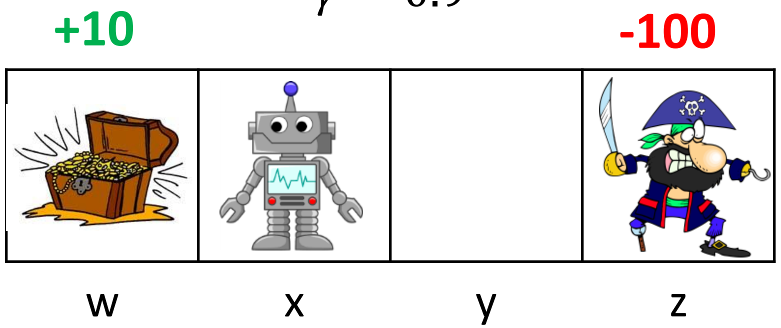
w	x	y	z
0	0,0	0,0	0

Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



x	$\rightarrow$	0	y
S	A	R	S'

$Q =$

0	0,0	0,0	0
w	x	y	z

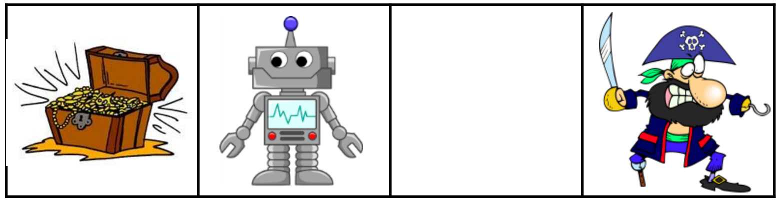
Q-learning: Off-policy TD Control

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

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

$\gamma = 0.9$

+10   -100



w   x   y   z

x	$\rightarrow$	0	y
S	A	R	S'

$Q =$

0	0,0	0,0	0
w	x	y	z

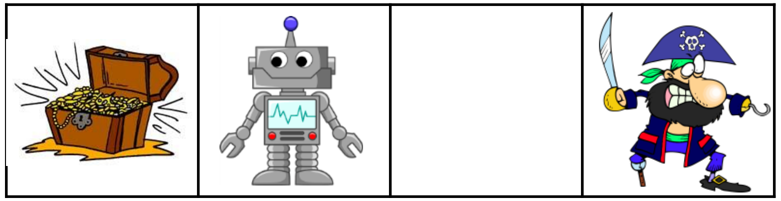
Q-learning: Off-policy TD Control

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

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

$\gamma = 0.9$

+10 -100



w x y z

y	$\rightarrow$	0	y
S	A	R	S'

$Q =$

0	0,0	0,0	0
w	x	y	z

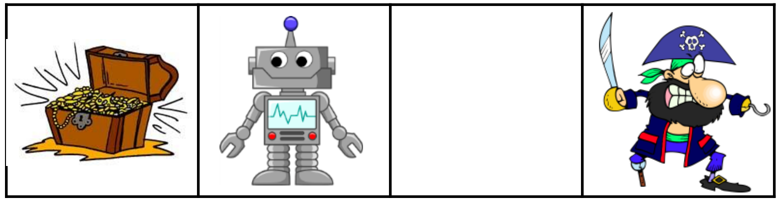
Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$

+10 -100



w x y z

y	$\rightarrow$	0	z
S	A	R	S'

$Q =$

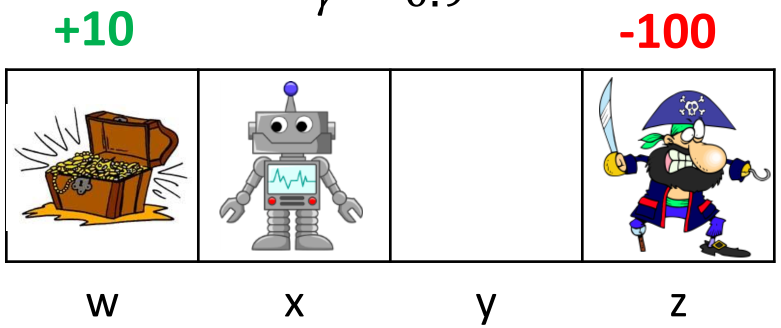
0	0,0	0,0	0
w	x	y	z

Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



S	A	R	S'
z	$\rightarrow$	0	z

$Q =$

w	x	y	z
0	0,0	0,0	0

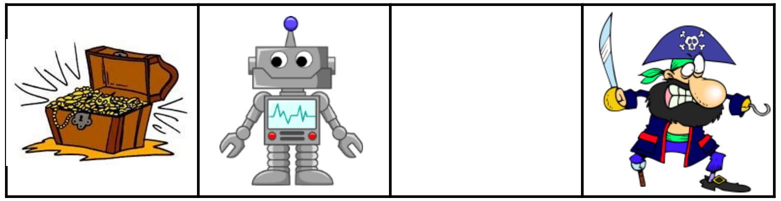
Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$

+10 -100



w x y z

z	exit	−100	·
S	A	R	S’

$Q =$

0	0,0	0,0	0
w	x	y	z

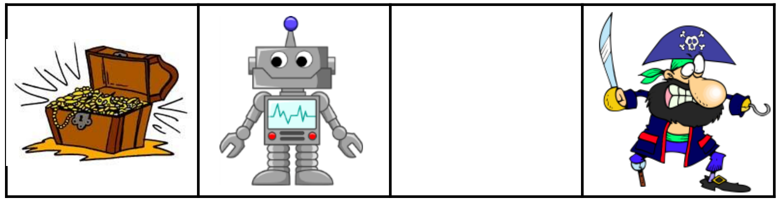
Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (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

$\gamma = 0.9$

+10 -100



w x y z

S	A	R	S'
z	$exit$	-100	$\cdot$

$Q =$

w	x	y	z
0	0,0	0,0	-100

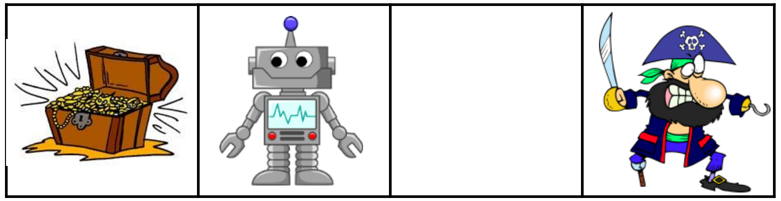
Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$

+10 -100



w x y z

x	$exit$	-100	$\cdot$
S	A	R	S'

$Q =$

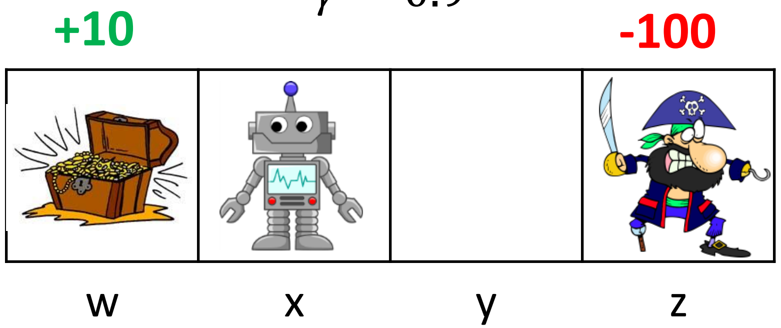
0	0,0	0,0	-100
w	x	y	z

Q-learning: Off-policy TD Control

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

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

$\gamma = 0.9$



x	$\rightarrow$	0	y
S	A	R	S'

$Q =$

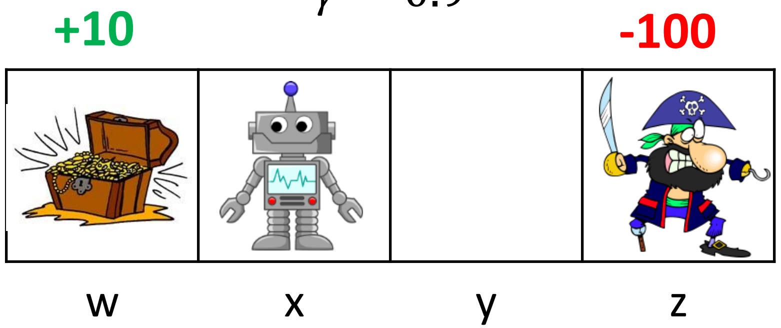
0	0,0	0,0	-100
w	x	y	z

Q-learning: Off-policy TD Control

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

Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



y	$\rightarrow$	0	z
S	A	R	S'

$Q =$

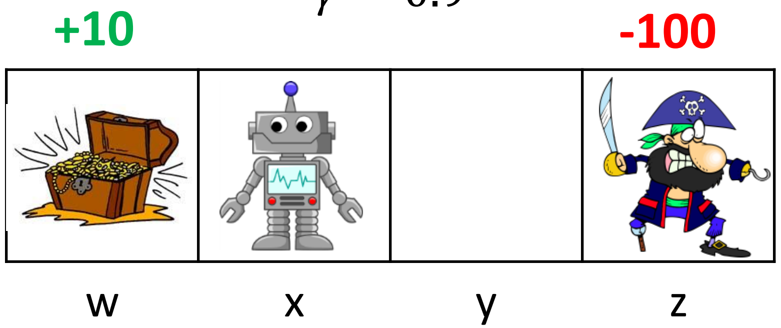
0	0,0	0,0	-100
w	x	y	z

Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



y	$\rightarrow$	0	z
S	A	R	S'

$Q =$

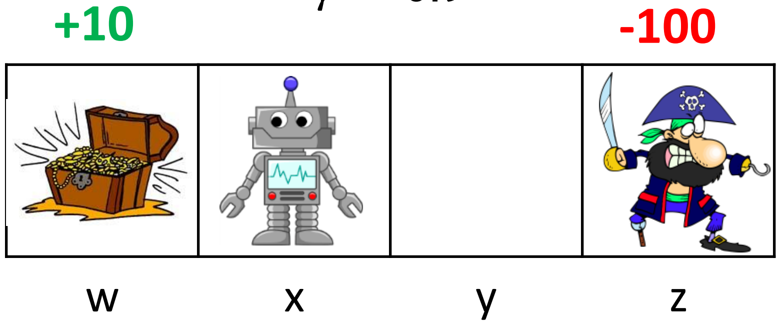
0	0,0	0,-90	-100
w	x	y	z

Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



S	A	R	S'
x	$\rightarrow$	0	z

$Q =$

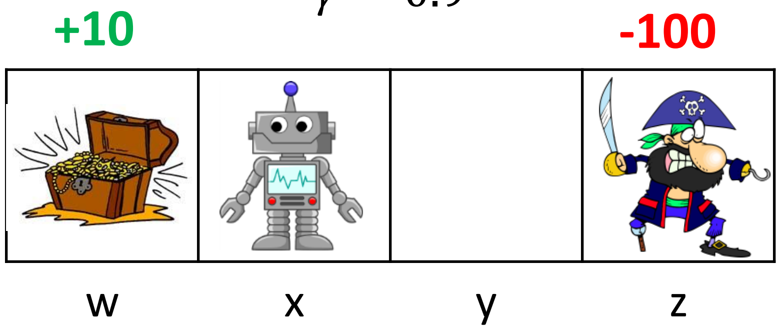
w	x	y	z
0	0,0	0,-90	-100

Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$



S	A	R	S'
x	$\rightarrow$	0	y

$Q =$

w	x	y	z
0	0,0	0,-90	-100

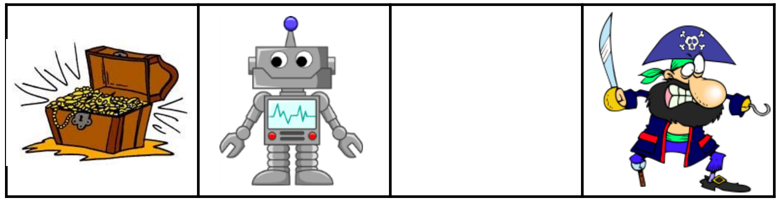
Q-learning: Off-policy TD Control

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

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (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

$\gamma = 0.9$

+10 **-100**



w x y z

x	$\rightarrow$	0	y
S	A	R	S'

$Q =$

0	0,0	0,-90	-100
w	x	y	z

Q-learning: Off-policy TD Control

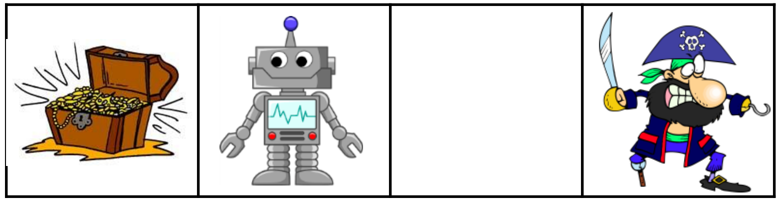
Q-learning (off-policy TD control) for estimating $\pi \approx \pi_*$

Initialize $Q(s, a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
    Initialize S
    Repeat (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

And so on…

$\gamma = 0.9$

+10 -100



w x y z

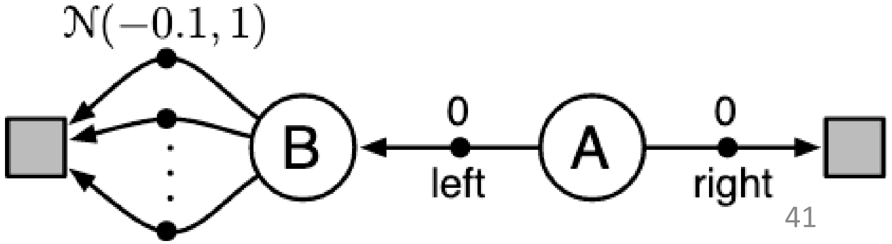
x	$\rightarrow$	0	y
S	A	R	S'

$Q =$

0	0,0	0,-90	-100
w	x	y	z

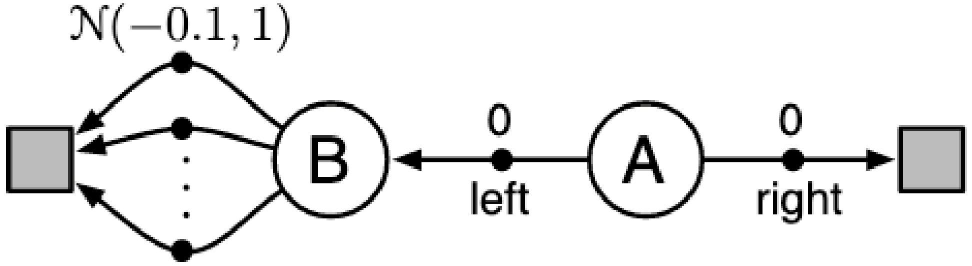
Maximization bias in Q-learning

- Consider an MDP with two non-terminal states A and B
- Episodes start in A with a choice between two actions, left and right
- The right action transitions immediately to the terminal state with a reward and return of zero
- The left action transitions to B, also with a reward of zero
- From B there are many possible actions all of which cause immediate termination with a reward drawn from a normal distribution with mean −0.1 and variance 1.0
- Thus, the expected return for any trajectory starting with left is −0.1, and thus taking left in state A is a mistake



Maximization bias in Q-learning

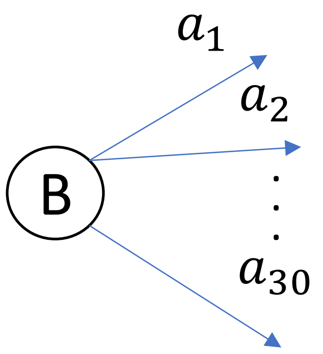
- Assume we observed 4 episodes:
  - $\{A, \rightarrow, 0\}$
  - $\{A, \leftarrow, 0, B, a_1, -1.1\}$
  - $\{A, \leftarrow, 0, B, a_2, 0.9\}$
  - $\{A, \leftarrow, 0, B, a_3, -0.1\}$



- $\max_a [Q(B, a)] = 0.9$ yet $q^*(B, \cdot) = -0.1$
- Maximization bias is common when action outcomes are noisy
- Some actions will be evaluated following a lower-than-expected sampled reward and some with a higher-than-expected sampled reward
- The max operator biases towards higher-than-expected sampled reward

Double Q-learning

- Assume visiting state B a total of 60 times (2 times per action)
- $a_i \sim \mathcal{N}(-0.1, 1)$
- $\mathbb{E}\left(\max_a [Q(B,a)]\right) = \sim^* 1.3 \neq -0.1$
- How can we fix this bias?
  - Store and update 2 independent Q tables: Q_1, Q_2
  - For each observed transition update one Q table (and not the other!)
  - Use one for choosing maximizing action and the other for retrieving the value
- $\mathbb{E}\left(Q_1(B, \operatorname{argmax}_a [Q_2(B,a)])\right) = -0.1$



$*$ evaluated through simulation

Double Q learning is unbiased

- We are actually interested in the action that maximizes the expected reward
  - $\max_a \left[\mathbb{E}[Q(s,a)]\right]$
- Initially $\mathbb{E}\left(\max_a [Q(s,a)]\right) \neq \max_a \left[\mathbb{E}[Q(s,a)]\right]$
  - Due to the maximization bias
- Double Q learning $\mathbb{E}\left(Q_1(s, \operatorname{argmax}_a [Q_2(s,a)]\right) = \max_a \left[\mathbb{E}[Q(s,a)]\right]$
  - Because the returned value (from Q_1) is independent of the maximizing action (from Q_2)

Double Q-learning

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

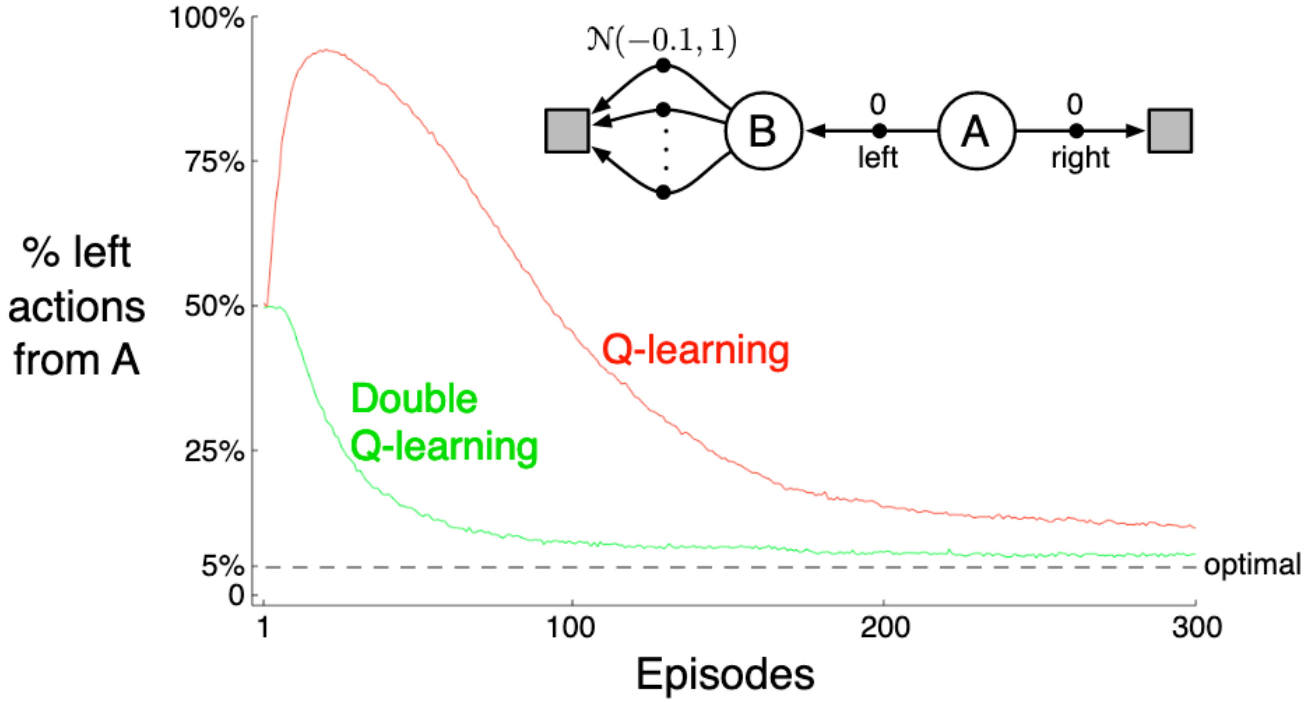
Initialize $Q(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, 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

Double Q-learning

Initialize $Q_1(s,a)$ and $Q_2(s,a)$, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily
Initialize $Q_1(\textit{terminal-state}, \cdot) = Q_2(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q_1 and Q_2 (e.g., 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

Double Q-learning

- From B there are many possible actions all of which cause immediate termination with a reward drawn from $\mathcal{N}(-0.1,1)$
- Taking left in state A is always a mistake



What did we learn?

- Temporal difference = online computation of the TD error, δ
- Allows us to perform unbiased online Bellman updates without any knowledge of the model
- In many cases converges faster than MC (no need to complete an episode)
- SARSA (on policy) provides unbiased TD learning through on policy estimations
- Q learning (off policy) might suffer from a maximization bias that can be addressed with double Q learning
- Both SARSA and Q learning are guaranteed to converge to the optimal state/action values in the tabular case + appropriate learning rate and epsilon decay

What next?

- **Class**: n-step Bootstrapping
- **Assignments**:
  - Tabular Q-Learning
  - SARSA
  - Due by October 7, EOD, through Canvas
- **Quiz (on Canvas)**:
  - TD learning
  - By Sep 18, EOD
- **Project**:
  - Define your research hypothesis/question