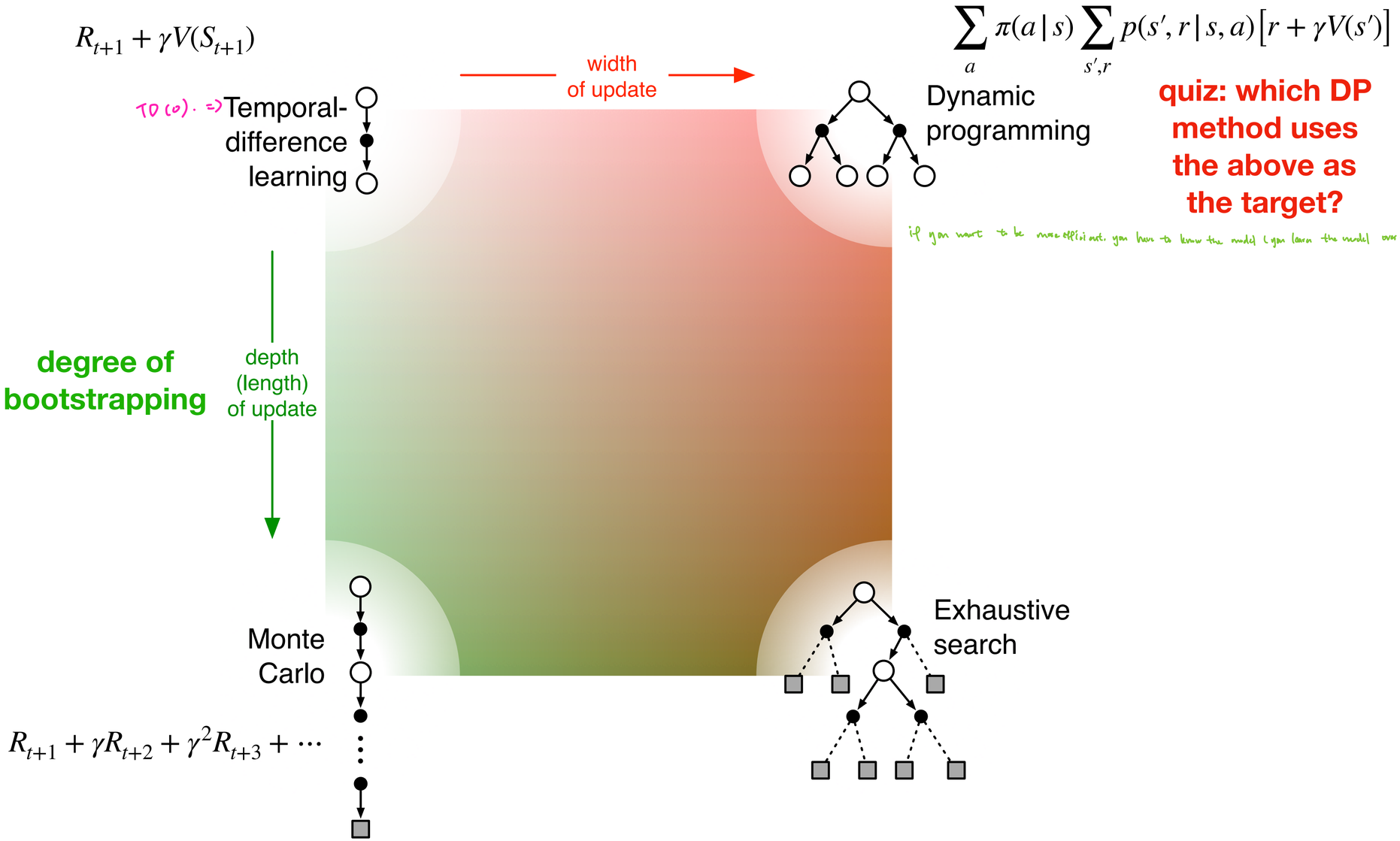
Planning, Learning and Acting

Rupam Mahmood

March 9, 2020

First: Summary of Part 1



Learning and planning

Model of the environment: used by agent to predict environment's response to actions

For example, if the agent has the transition probabilities $p(s',r|s,a)$ as in dynamic programming

Planning: a process that takes model as an input and produces/improves a policy

Dynamic programming methods are planning methods

A planning method uses the model to simulate the environment to produce simulated experience which is used to produce/improve a policy

Deepak Ranganatha Sastry Mamillapalli 下午1:17
对所有人说

DR so all model-based learning uses planning? => Yes. model plays the role to plan.

but if the model isn't accurate, wouldn't the simulated experience be different and biased from real experience? Yes.

Model learning: the process of using real experience to improve the model

Deepak Ranganatha Sastry Mamillapalli 下午1:24
对所有人说

DR if we are making updates to both the model and value functions, then they should both converge to their true values right?

For example, if we are estimating the transition probabilities based on real experience

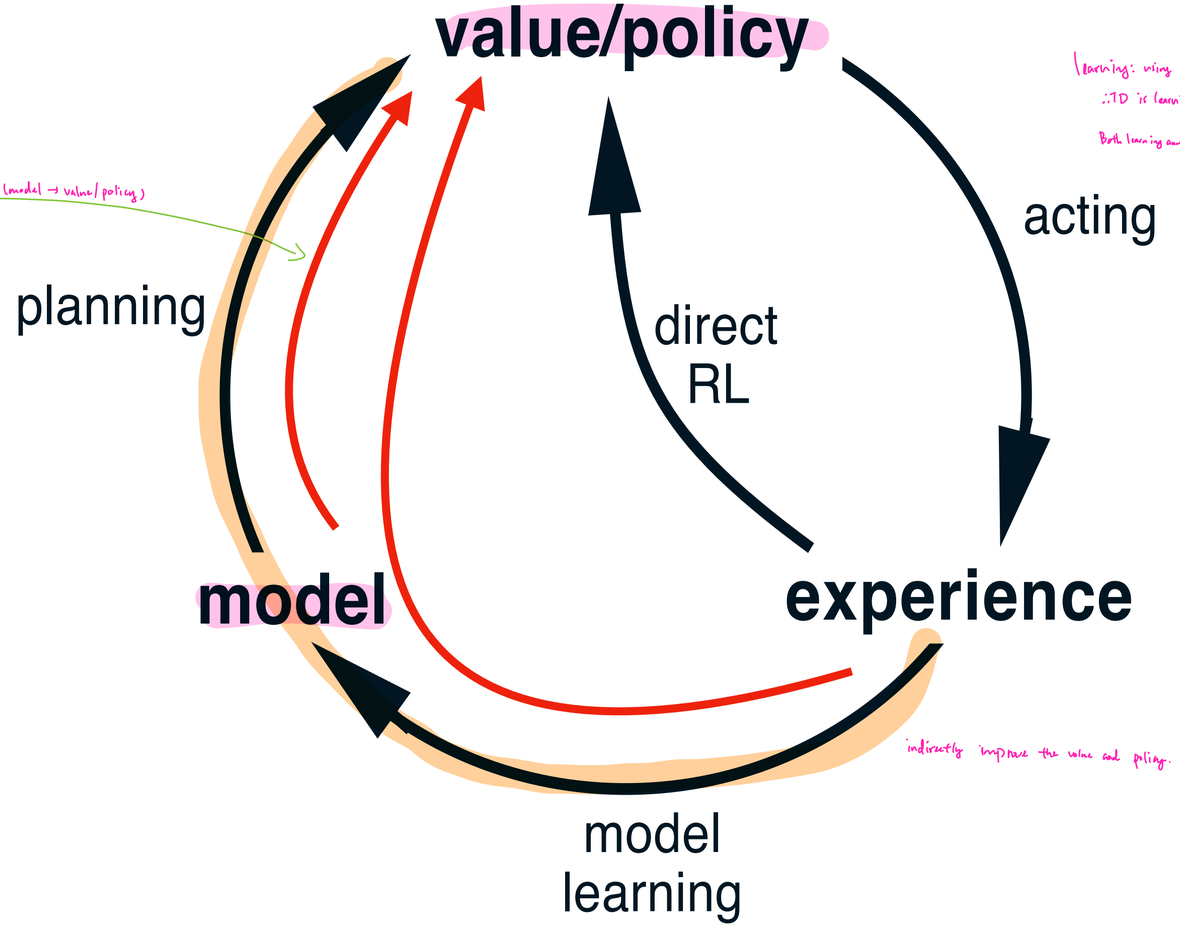
Direct reinforcement learning: the process of using real experience to directly improve the value function and policy

For example, TD or MC methods for prediction and control

Indirect reinforcement learning: the process of using real experience to improve the value function and policy using the model through planning

An architecture for planning, learning and acting at the same time

quiz: indicate indirect RL with an arrow (which of the two red ones?)



Tabular Dyna-Q

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

Loop forever:

(a) $S \leftarrow$ current (nonterminal) state

(b) $A \leftarrow \varepsilon$-greedy(S,Q)

(c) Take action A; observe resultant reward, R, and state, S'

(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)

(f) Loop repeat n times:

$S \leftarrow$ random previously observed state

$A \leftarrow$ random action previously taken in S

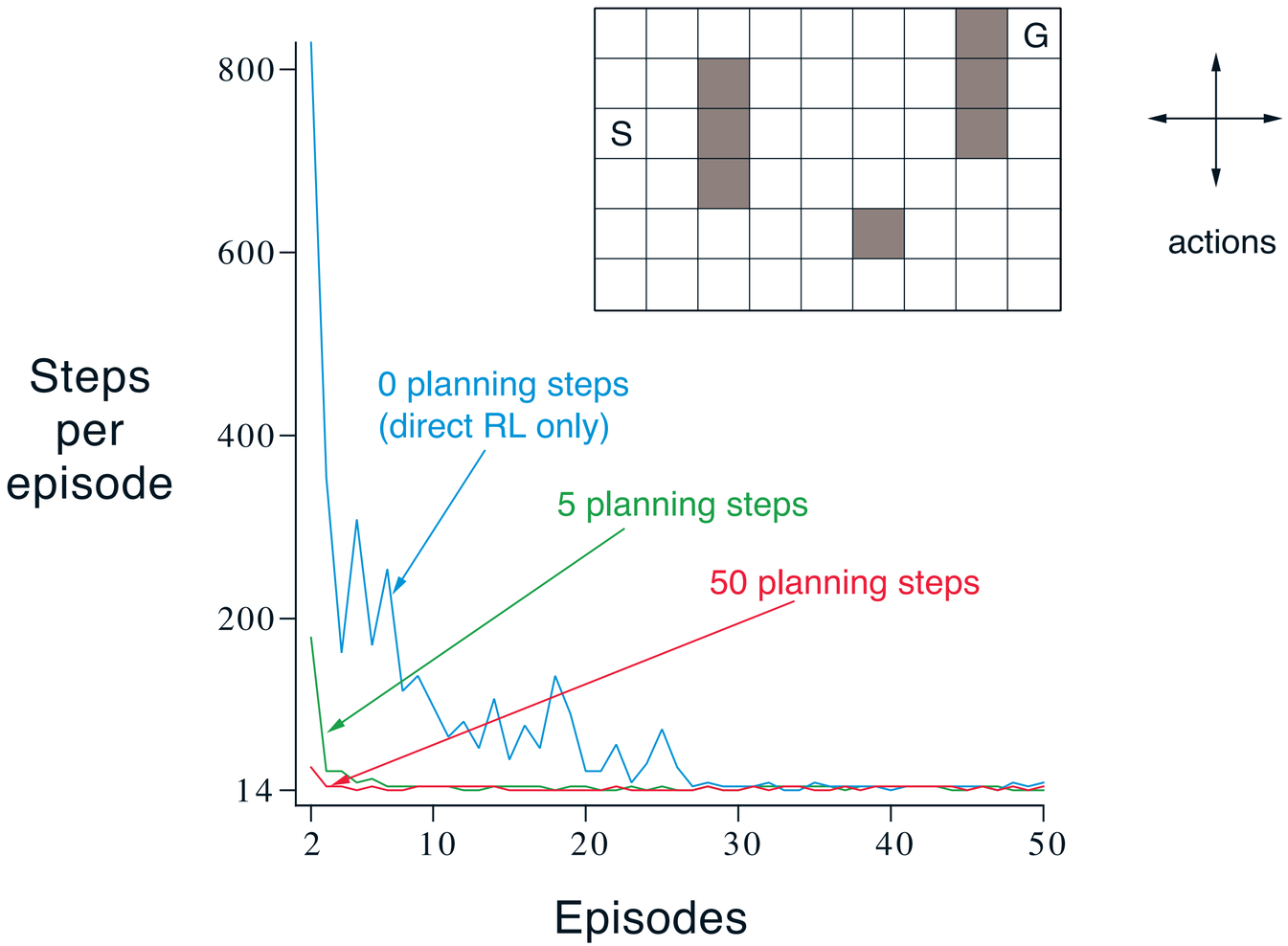
$R, S' \leftarrow Model(S,A)$

$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

Which part is direct RL? What is this method called?

Which part is model learning?

Which part is planning?


G
S
actions
Steps per episode
800
600
400
200
14
0 planning steps
(direct RL only)
5 planning steps
50 planning steps
2
10
20
30
40
50
Episodes

How many planning steps should we take?

Tabular Dyna-Q

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

Loop forever:

(a) $S \leftarrow$ current (nonterminal) state

(b) $A \leftarrow \varepsilon$-greedy(S,Q)

(c) Take action A; observe resultant reward, R, and state, S'

(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment) => storing the sample.

(f) Loop repeat n times:

$S \leftarrow$ random previously observed state

$A \leftarrow$ random action previously taken in S

$R, S' \leftarrow Model(S,A)$

$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

Dyna-Q+

1. Adds a bonus $\kappa\sqrt{\tau(s,a)}$ to reward in planning

$\tau(s,a)$ denotes the number of time steps (s,a) has not been tried

2. Actions that have not been tried from a previously visited state are allowed to be considered in planning

Where would you put these steps in Dyna-Q to get Dyna-Q+?

Tabular Dyna-Q

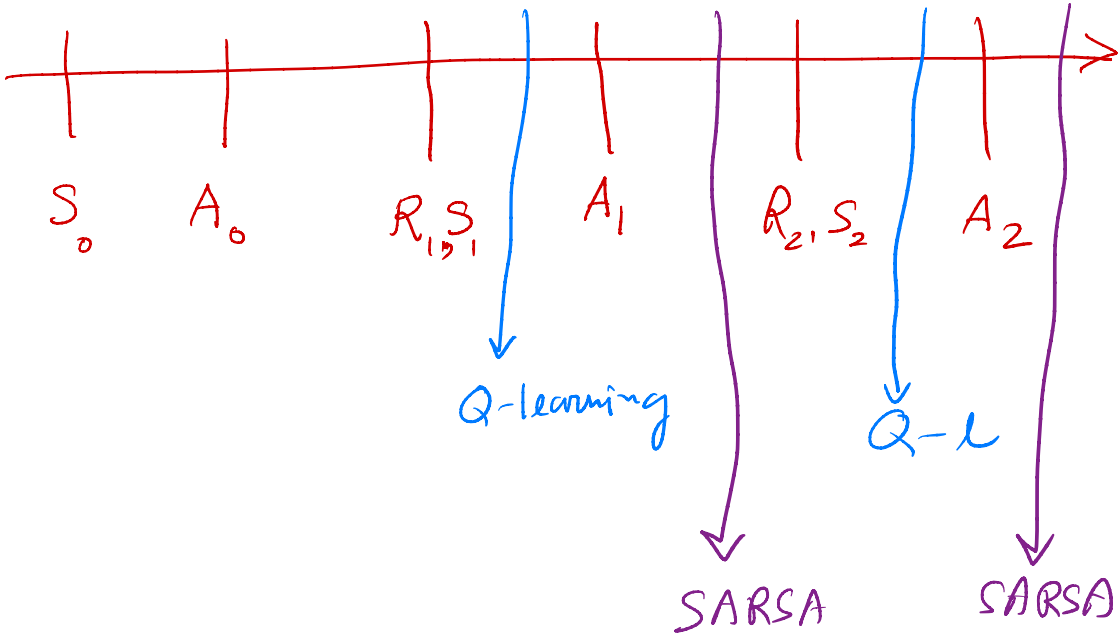
Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

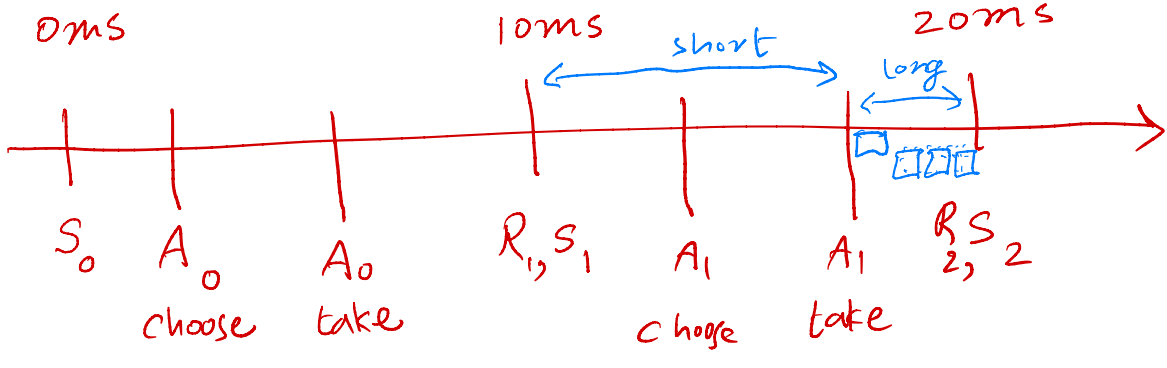
Dyna-Q+: calculating visitation counts

Consider an MDP with one actions (L) and two states with the following episode

S_0	A_0	S_1	A_1	S_2	A_2	S_3	A_3	S_4	A_4
y	L	x	L	x	L	y	L	x	L

Calculate $\tau(s, a)$ for all state-action pairs at each step


S_0
A_0
R_1, S_1
A_1
R_2, S_2
A_2
Q-learning
Q-l
SARSA
SARSA


0ms
10ms
20ms
short
long
S_0
A_0
choose
A_0
take
R_1, S_1
A_1
choose
A_1
take
R_2, S_2

① model is incorrect because 1. environment is 随机的.
(more and more test, more and more sample → converge to true model. 但在之前 you are using the wrong model).
2. model learn using function approximation that generalized imperfectly.
3. sub optimal policy computed by planning → discovering the modeling error.
If the model is optimistic, you can quickly discover the error although it's wrong.

② wrong model will lead you to discover the error in the model.

③. Dyna Q+: add bonus at update. (# of time step you try to make the transition : the bonus). $r + k\sqrt{\tau}$ (time steps)
与 Dyna Q 的唯一区别是 a small value is added to reward. 其他都一样.
is Dyna Q with an exploration bonus that encourages exploration.
从一开始表现就比 Dyna Q 好, 一直持续到最后.

④ Dyna Q never discover the optimal policy, never change its behavior. shortcut is an example of better environment.
Dyna Q doesn't explore as much and just exploits.

⑤ Dyna Q and Dyna Q+ 在 1000 time steps 后 have same slope, so they have same performance.
slope 小, 则证明更慢
大 (陡) 就更快.

⑥

Worksheet 9: Planning, Learning & Acting — CMPUT 397, March 19, 2021

6. (*Exercise 8.2 S&B*) Why did the Dyna agent with exploration bonus, Dyna-Q+, perform better in the first phase as well as in the second phase of the blocking experiment in Figure 8.4?

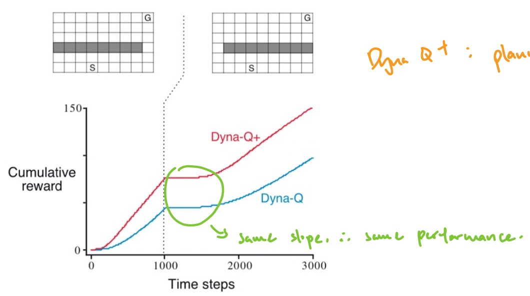


Figure 8.4: Average performance of Dyna agents on a blocking task. The left environment was used for the first 1000 steps, the right environment for the rest. Dyna-Q+ is Dyna-Q with an exploration bonus that encourages exploration. ■

Dyna Q+ : planning $r + k\sqrt{\tau}$ (bonus reward) ⇒ not included in performance measure.

A: At the first episode, (1000 step 以前), Dyna Q+ agent has the exploration bonus. ∴ 它不断探索新的 action. ∴ visits S|A pairs that are relatively unvisited. ⇒ ∴ 它更有机会比 Dyna Q 更早找到 goal state.
而 Dyna Q 一开始用的是 random policy, 不一定会随机到没 visit 过的 S|A pair.

Worksheet 9: Planning, Learning & Acting

CMPUT 397
March 19, 2021

7. (*Exercise 8.3 S&B*) **Challenge Question:** Careful inspection of Figure 8.5 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this?

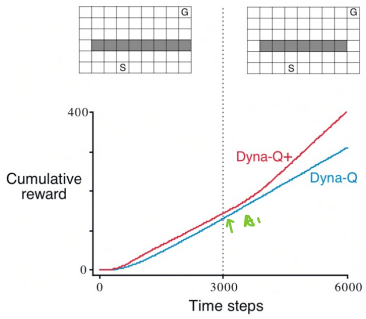


Figure 8.5: Average performance of Dyna agents on a shortcut task. The left environment was used for the first 3000 steps, the right environment for the rest.

■

A: ∵ 随着 Dyna Q+ keep explore. i.e. it doesn't always follow the optimal path. ∴ it narrowed slightly in performance.

Worksheet 9: Planning, Learning & Acting

CMPUT 397
March 19, 2021

3. Modify the Tabular Dyna-Q algorithm so that it uses Expected Sarsa instead of Q-learning. Assume that the target policy is ϵ-greedy. What should we call this algorithm?

with ε-greedy target policy

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

A: 将 Tabular Dyna Q 变为 Expected Sarsa:

将 d) 和 f):

将 d) 变为 Expected Sarsa update: $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \sum_{a'} \pi(a'|S') Q(S',a) - Q(S,A)\big]$

将为 ε-greedy 作为 target policy, ∴ Before each update, calculate π: $\pi(a|S) = \begin{cases} 1-\varepsilon+\frac{\varepsilon}{|A|} & \text{if } a = \arg\max_a Q(S,A) \\ \frac{\varepsilon}{|A|} & \text{otherwise} \end{cases}$

变为 dyna - expected Sarsa.

How many planning steps should we take?

As many as possible within the time / budget constraint.

Takes time away from actually interacting with the environment.

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
- (a) $S \leftarrow$ current (nonterminal) state
- (b) $A \leftarrow \varepsilon$-greedy(S,Q)
- (c) Take action A; observe resultant reward, R, and state, S'
- (d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
- (e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
- (f) Loop repeat n times:
  - $S \leftarrow$ random previously observed state
  - $A \leftarrow$ random action previously taken in S
  - $R, S' \leftarrow Model(S,A)$
  - $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$