November 14, 2022

8 Exercises

8.1 Question

The nonplanning method looks particularly poor in Figure 8.3 because it is a one-step method; a method using multi-step bootstrapping would do better. Do you think one of the multi-step bootstrapping methods from Chapter 7 could do as well as the Dyna method? Explain why or why not.

Answer

n-step learning methods would do just as good as n-step planning methods. n-step learning methods make better use of samples by updating multiple states.

n-step planning uses random state actions to update state values while n-step learning methods updates states leading to a reward. n-step planning uses generated model to update states while n-step methods needs samples to make the updates.

8.2 Question

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 and shortcut experiments?

Answer

Assuming Example 8.2 and 8.3 use the same reward structure as the example 8.1.

In the first phase, DynaQ+ may have found the shortest path faster thanks to its more robust exploration strategy;

DynaQ may rely on ϵ greedy policy to explore. Sometimes random exploration will result in exploring same places multiple times which may result in slows down the search for the best path.

In the second phase, DynaQ+ has a clearer advantage, thanks to its more robust exploration strategy. DynaQ relies solely on ϵ-greedy search and not able to find new shorter path fast.

DynaQ+ exploration strategy may help greatly in such cases but often times it will just waste CPU times.

8.3 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?

Answer

Both DynaQ and DynaQ+ have reportedly found the shortest path in the first 1000 steps. After finding the shortest path DynaQ mostly exploits it. DynaQ+ on the other hand, continues exploration. Longer episodes mean less rewards.

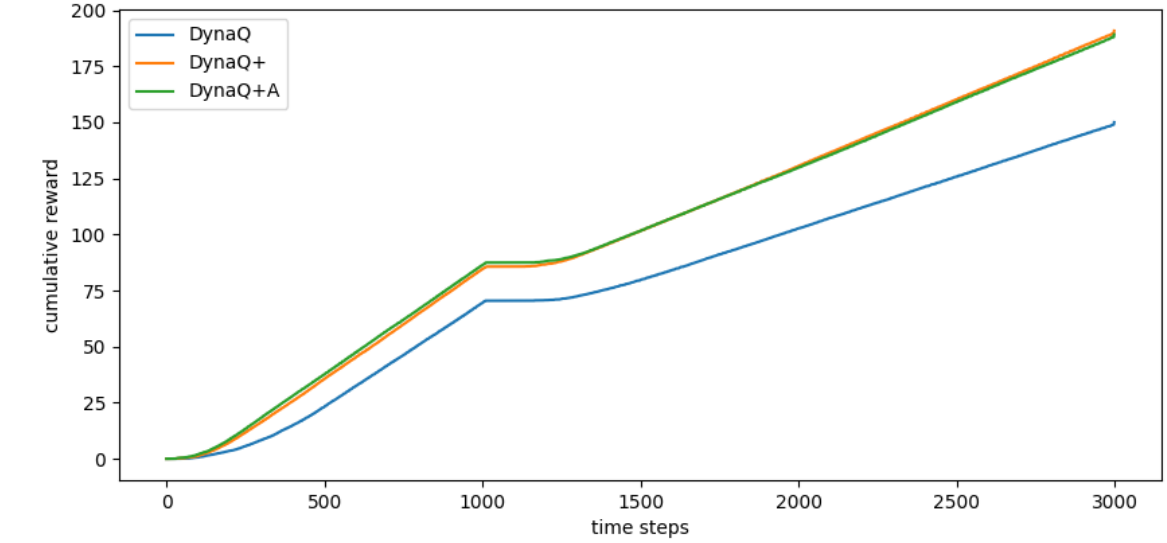
Exploring fast is beneficial first, but it is not desirable once the environment is mostly explored. Eventually longer episodes may have let the DynaQ to narrow the gap.

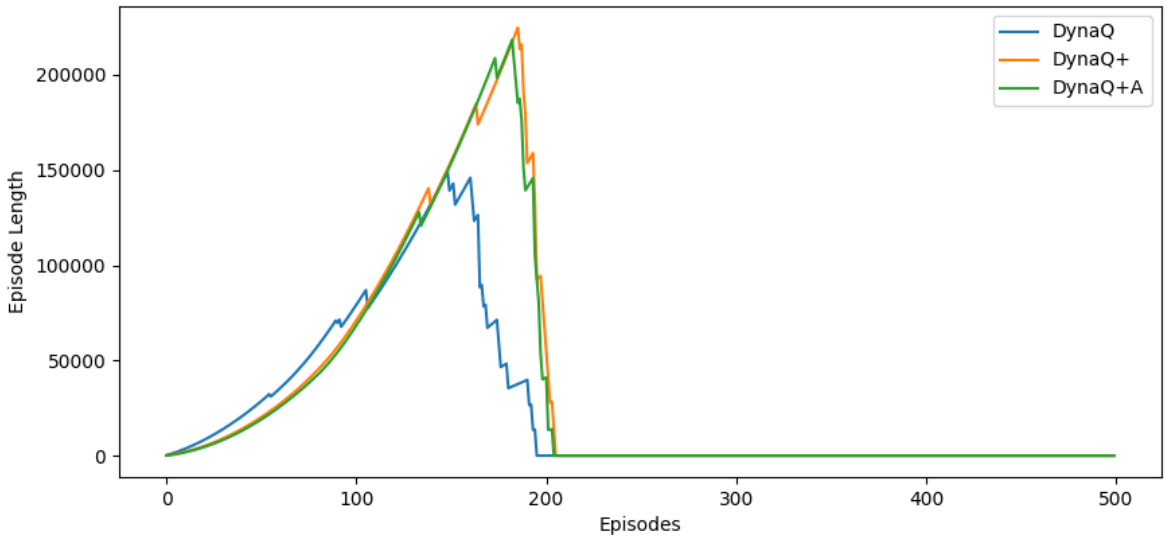
8.4 Question

(programming) The exploration bonus described above actually changes the estimated values of states and actions. Is this necessary? Suppose the bonus was used not in updates, but solely in action selection. That is, suppose the action selected was always that for which Q(St,a) + K * tau(S t , a) was maximal. Carry out a gridworld experiment that tests and illustrates the strengths and weaknesses of this alternate approach.

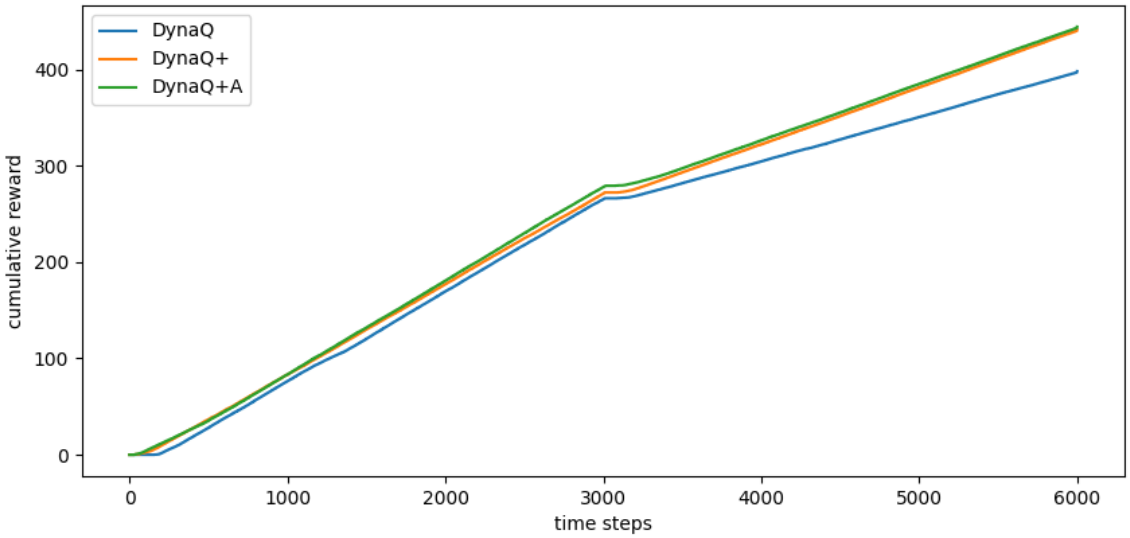
Answer

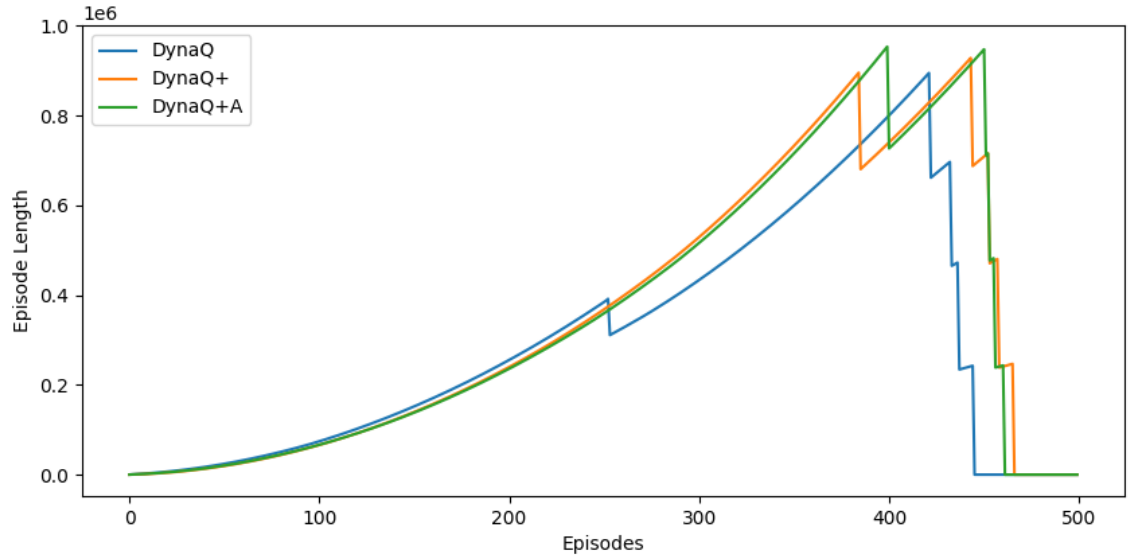
Example 8.4 configuration.





Example 8.5 configuration.





The alternate DynaQ+ approach shows similar behaviour to that of DynaQ+.

The alternate approach does not modify the q values. While the DynaQ+ approach encourage exploration of stages and new actions, the alternate approach only encourages exploring actions. Less calculation in planning phase may slightly reduce planning phase cost.

8.5 Question

How might the tabular Dyna-Q algorithm shown on page 164 be modified to handle stochastic environments? How might this modification perform poorly on changing environments such as considered in this section?

How could the algorithm be modified to handle stochastic environments and changing environments?

Answer

Deterministic algorithm keeps track of state action pairs and their q value. A stochastic algorithm should also keep track the number of times an action is taken and which next state is visited. Q value is formed as an expected value of possible next states and rewards.

It may be necessary to apply expected value updates in planning as well.

This approach is not suitable for changing environments. New transitions only will look like rare transitions to the algorithm. It will take a lot of time until new transitions gain significance.

A solution would be to give more weight to recent transitions.