

# **COMP-767: Reinforcement Learning**

## **Dyna-Q and count-base exploration**

Gabriele Prato

McGill University

# Dyna-Q

## Tabular Dyna-Q

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

Do forever:

- (a)  $S \leftarrow$  current (nonterminal) state
- (b)  $A \leftarrow \epsilon$ -greedy( $S, Q$ )
- (c) Execute action  $A$ ; observe resultant reward,  $R$ , and state,  $S'$
- (d)  $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
- (e)  $Model(S, A) \leftarrow R, S'$  (assuming deterministic environment)
- (f) Repeat  $n$  times:
  - $S \leftarrow$  random previously observed state
  - $A \leftarrow$  random action previously taken in  $S$
  - $R, S' \leftarrow Model(S, A)$
  - $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

Q-learning

Planning

# Dyna-Q+

- Adds a reward bonus  $R + K\sqrt{T}$

## Tabular Dyna-Q

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

Do forever:

- (a)  $S \leftarrow$  current (nonterminal) state
- (b)  $A \leftarrow \epsilon$ -greedy( $S, Q$ )
- (c) Execute action  $A$ ; observe resultant reward,  $R$ , and state,  $S'$
- (d)  $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
- (e)  $Model(S, A) \leftarrow R, S'$  (assuming deterministic environment)
- (f) Repeat  $n$  times:
  - $S \leftarrow$  random previously observed state
  - $A \leftarrow$  random action previously taken in  $S$
  - $R, S' \leftarrow Model(S, A)$
  - $R \leftarrow R + K\sqrt{T}$
  - $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

# Greedy Dyna-Q+[<sup>1</sup>]

- Apply reward bonus in action selection

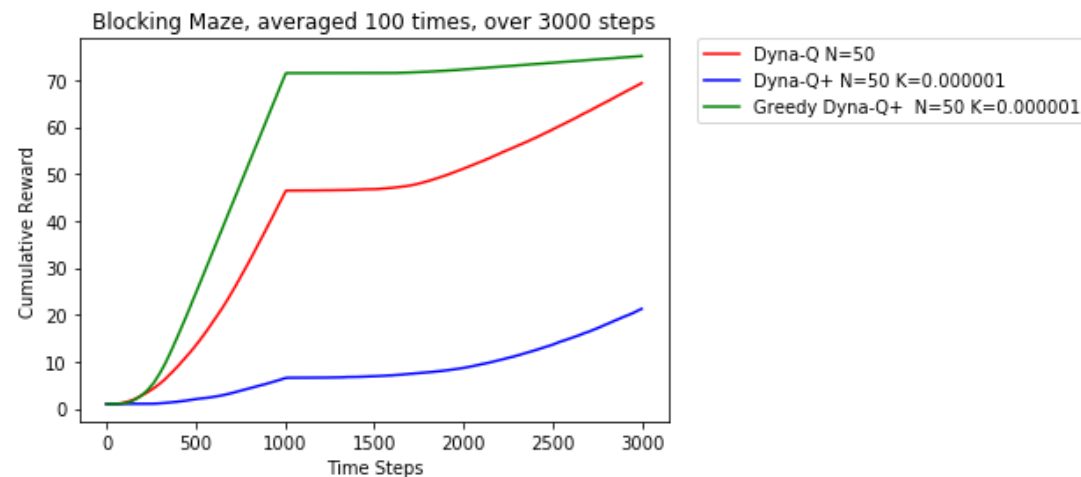
## Tabular Dyna-Q

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

Do forever:

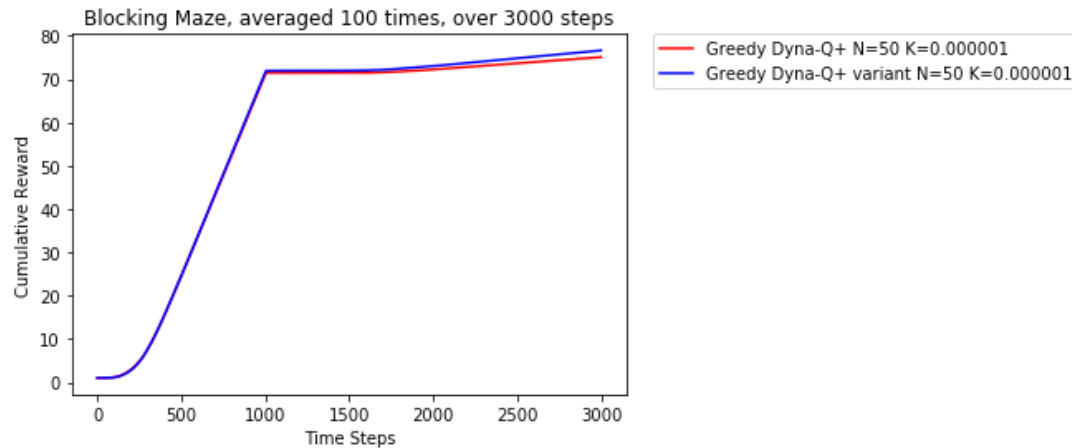
- $S \leftarrow$  current (nonterminal) state
- $A \leftarrow \epsilon\text{-greedy}(S, Q) \leftarrow \max_a Q(S, a) + K\sqrt{T_{Sa}}$
- Execute action  $A$ ; observe resultant reward,  $R$ , and state,  $S'$
- $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
- $Model(S, A) \leftarrow R, S'$  (assuming deterministic environment)
- Repeat  $n$  times:
  - $S \leftarrow$  random previously observed state
  - $A \leftarrow$  random action previously taken in  $S$
  - $R, S' \leftarrow Model(S, A)$
  - ~~$R \leftarrow R + K\sqrt{T}$~~
  - $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$

[1] Richard S. Sutton and Andrew G. Barto, "Reinforcement learning: An introduction", Second Edition, MIT Press, in preparation p.177 Exercise 8.4

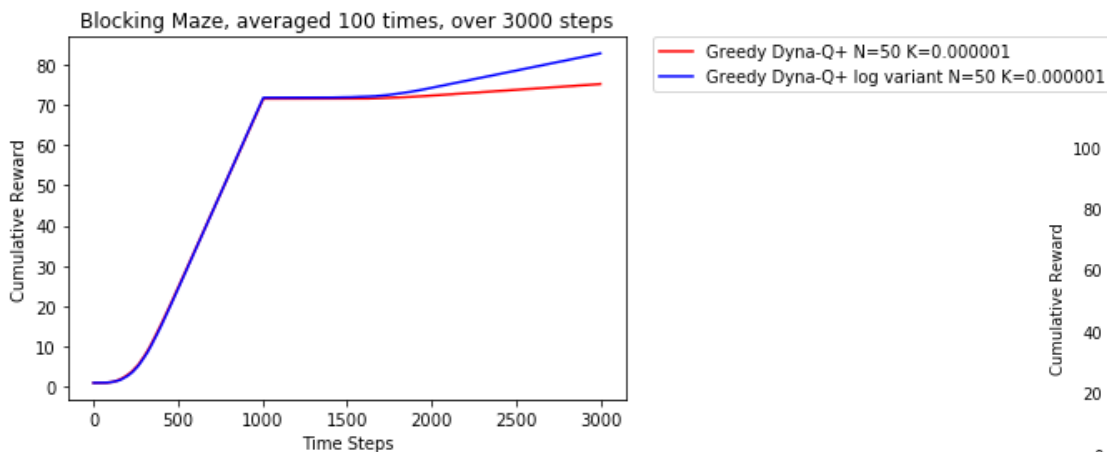


# Greedy Dyna-Q+ Improvements?

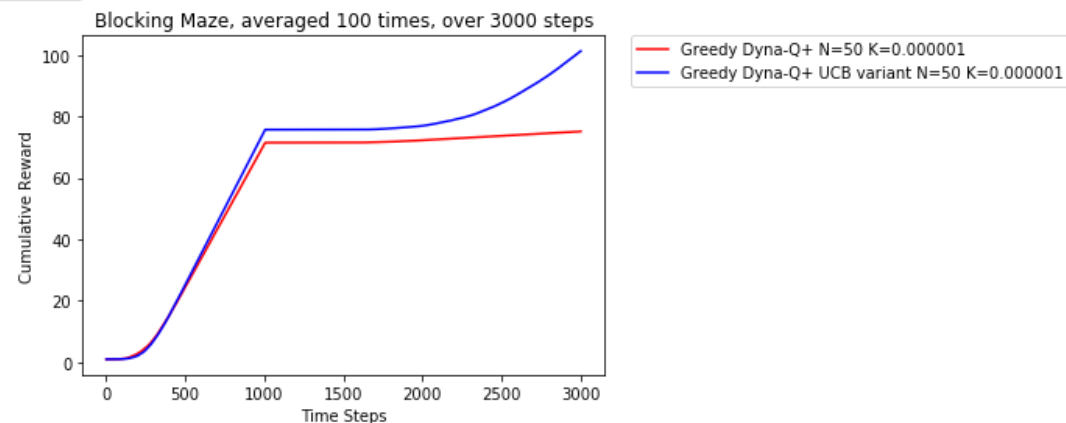
- Increase  $T_s$  only when in state  $S$



- $K \sqrt{\log T}$

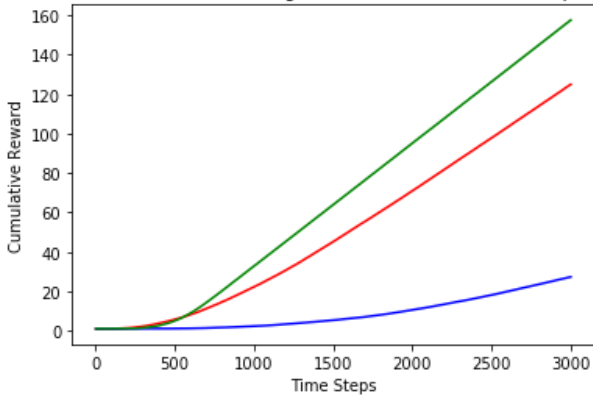


- UCB  $K \sqrt{\frac{\log t}{N_t(a)}}$



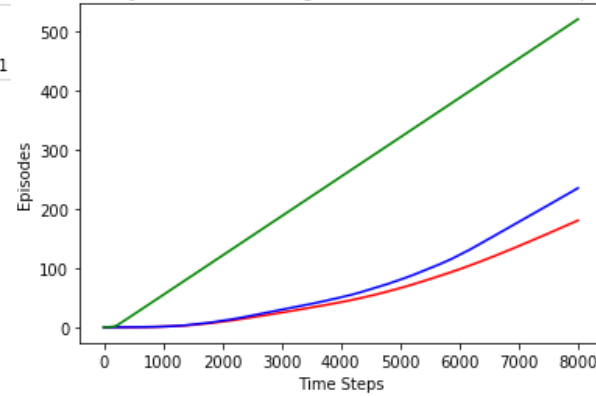
# Greedy Dyna-Q+ UCB variant

Shortcut Maze, averaged 100 times, over 3000 steps



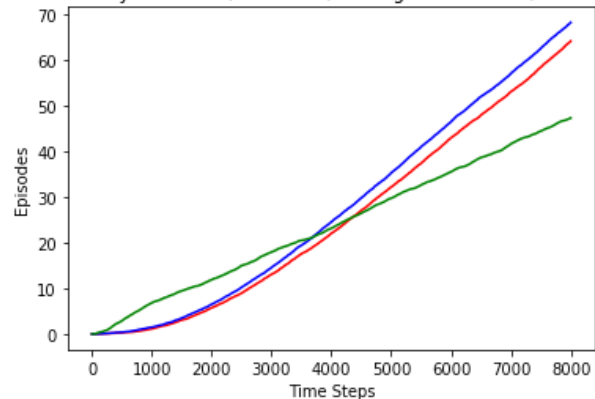
— Dyna-Q N=50  
— Dyna-Q+ N=50 K=0.000001  
— Greedy Dyna-Q+ UCB variant N=50 K=0.000001

Windy Gridworld, averaged 100 times, over 8000 steps



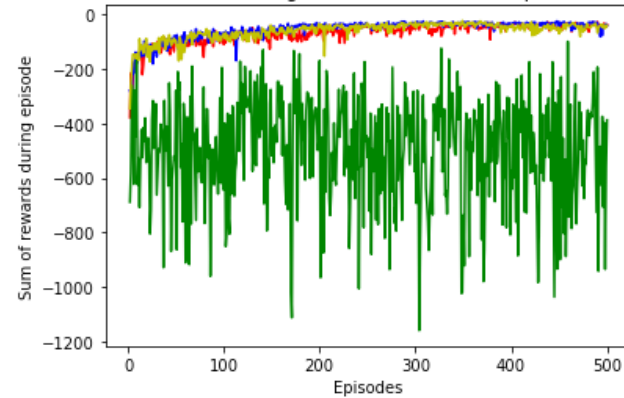
— Sarsa epsilon=0.1  
— Q-learning epsilon=0.1  
— Greedy Dyna-Q+ UCB variant N=50 K=0.000001

Stochastic Windy Gridworld, 8 actions, averaged 100 times, over 8000 steps



— Sarsa epsilon=0.1  
— Q-learning epsilon=0.1  
— Greedy Dyna-Q+ UCB variant N=50 K=0.000001

The Cliff, averaged 20 times, over 500 episodes



— Sarsa  
— Q-learning  
— Greedy Dyna-Q+ UCB variant N=50 K=0.000001  
— Expected Sarsa