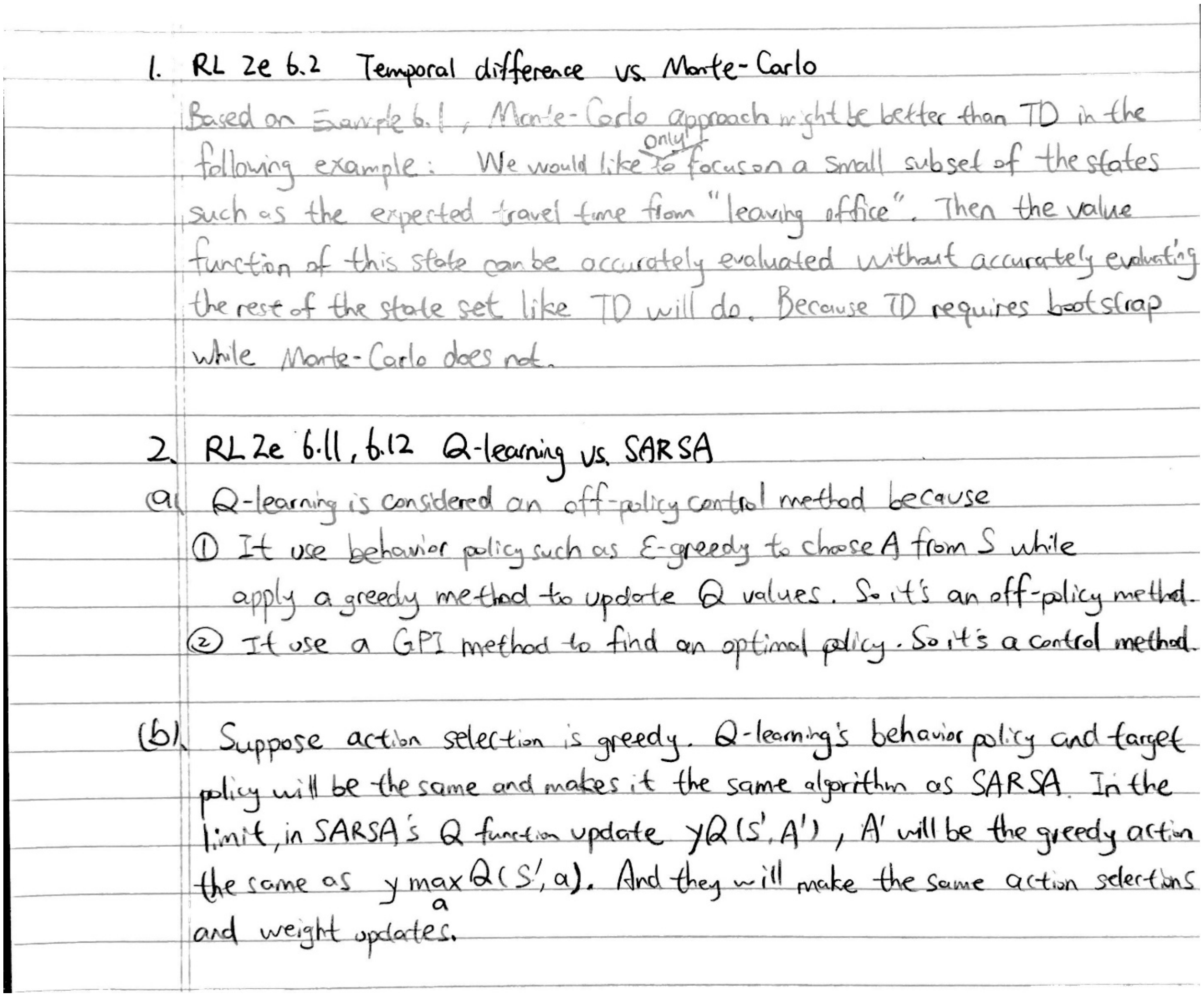
'''

CS 5180 Fall 2022

Exercise 5: Temporal-Difference Learning

Hongyan Yang

'''



1. RL 2e 6.2 Temporal difference vs. Monte-Carlo

Based on Example 6.1, Monte-Carlo approach might be better than TD in the following example: We would like to only focus on a small subset of the states such as the expected travel time from "leaving office". Then the value function of this state can be accurately evaluated without accurately evaluating the rest of the state set like TD will do. Because TD requires bootstrap while Monte-Carlo does not.

2. RL 2e 6.11, 6.12 Q-learning vs. SARSA

(a) Q-learning is considered an off-policy control method because

① It use behavior policy such as ε-greedy to choose A from S while apply a greedy method to update Q values. So it's an off-policy method.

② It use a GPI method to find an optimal policy. So it's a control method.

(b) Suppose action selection is greedy. Q-learning's behavior policy and target policy will be the same and makes it the same algorithm as SARSA. In the limit, in SARSA's Q function update $\gamma Q(S', A')$, A' will be the greedy action the same as $\gamma \max_a Q(S', a)$. And they will make the same action selections and weight updates.

3, RL 2e 6.4, 6.5 Random-walk task

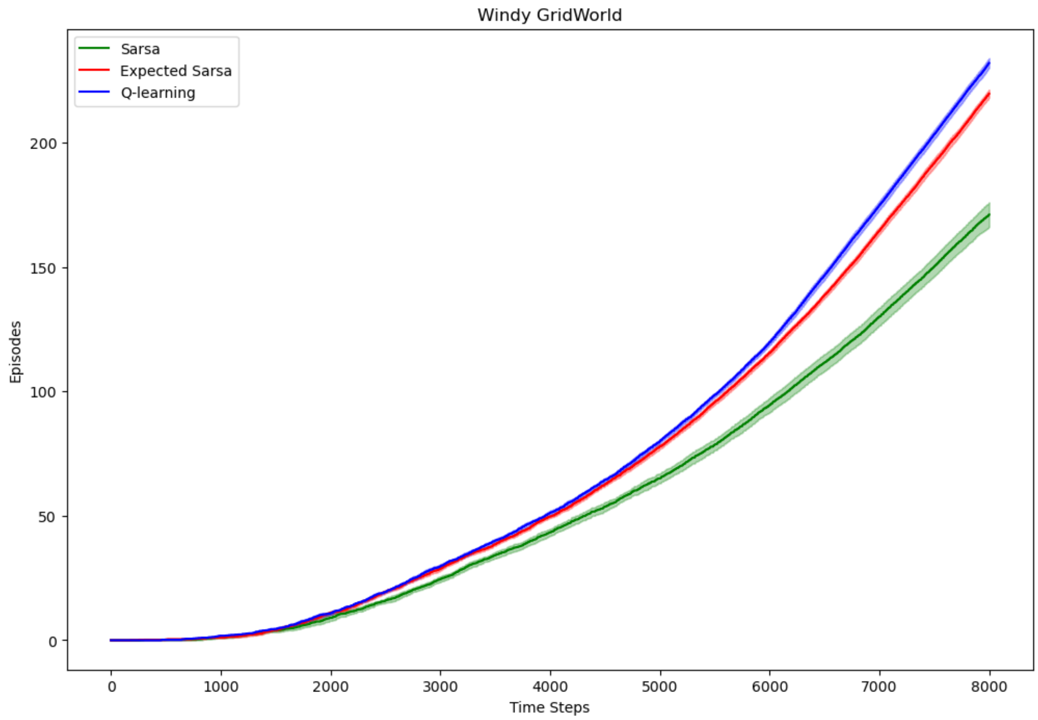
a. ① The conclusions about which algorithm is better would not be affected by a wider range of α. As can be seen from the graph in Example 7.1. TD(0) method outperforms TD(512), which is nearly the MC method in each but very small α in the first 10 episodes. But in the limit, with such a small alpha, both methods will eventually converge to the true value function with low RMS error.

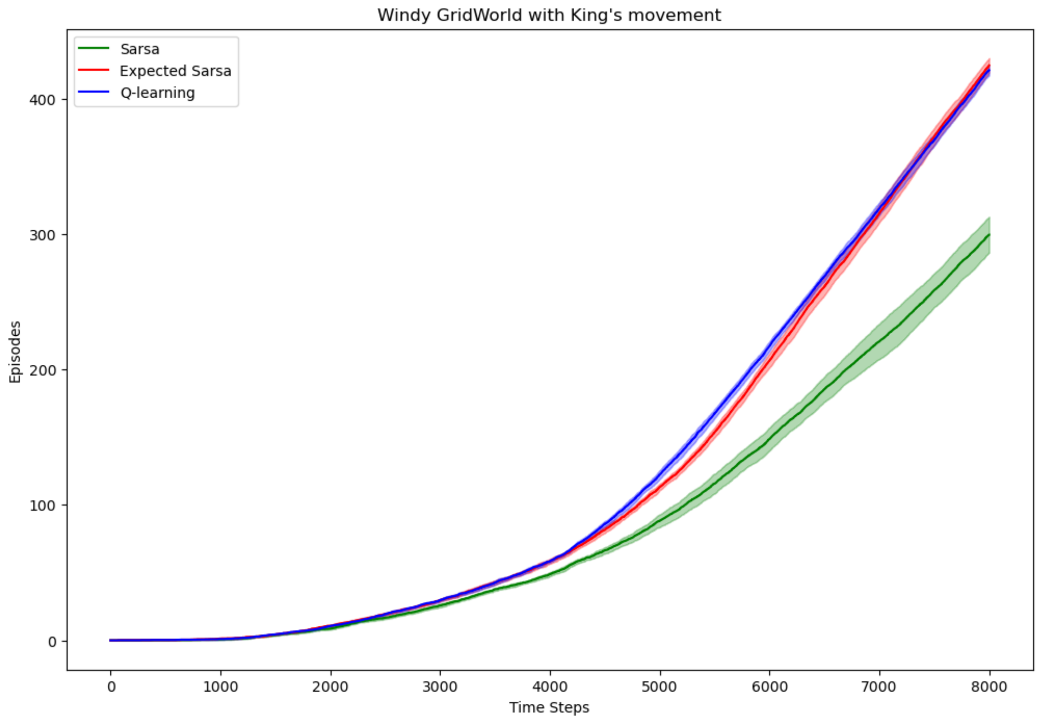
② Both methods' performance will be affected by the α setting as shown in Example 7.1. As the concave shape suggests, an intermediate α setting will let both algorithms perform better than other αs.

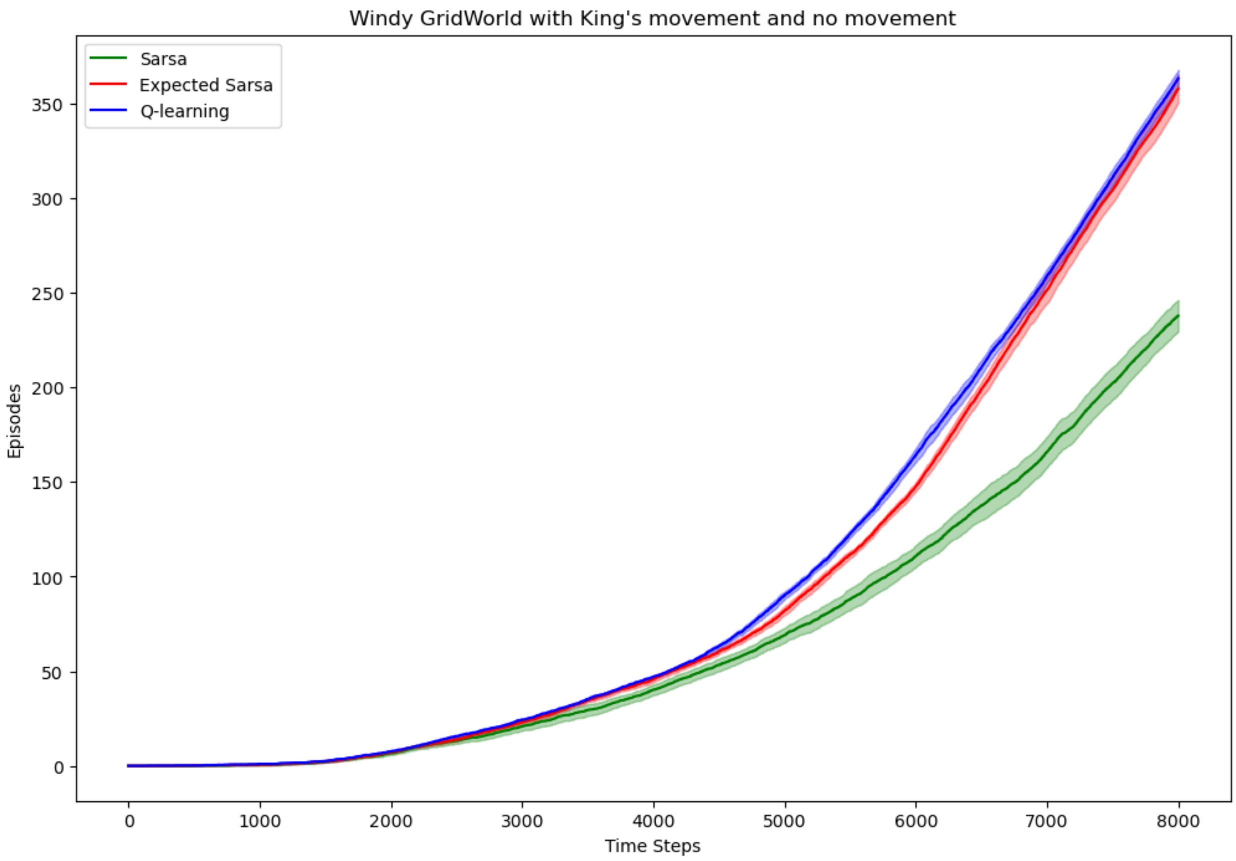
(b). ① Going down then go up at high α is caused by the bootstrap with S' to update the estimate of the value function. An α too large results in the fluctuate owing to giving recent update too much weight.

② This specific "go down then go up" pattern is related to the initialization. With a $V(s)=0.5$ for all states to initialize and a large α. Value functions are updated towards the true value in large steps. Making the RMS error go down at early episodes then go up due to putting too much weight on recent observations and the estimated value fluctuate with large variance.

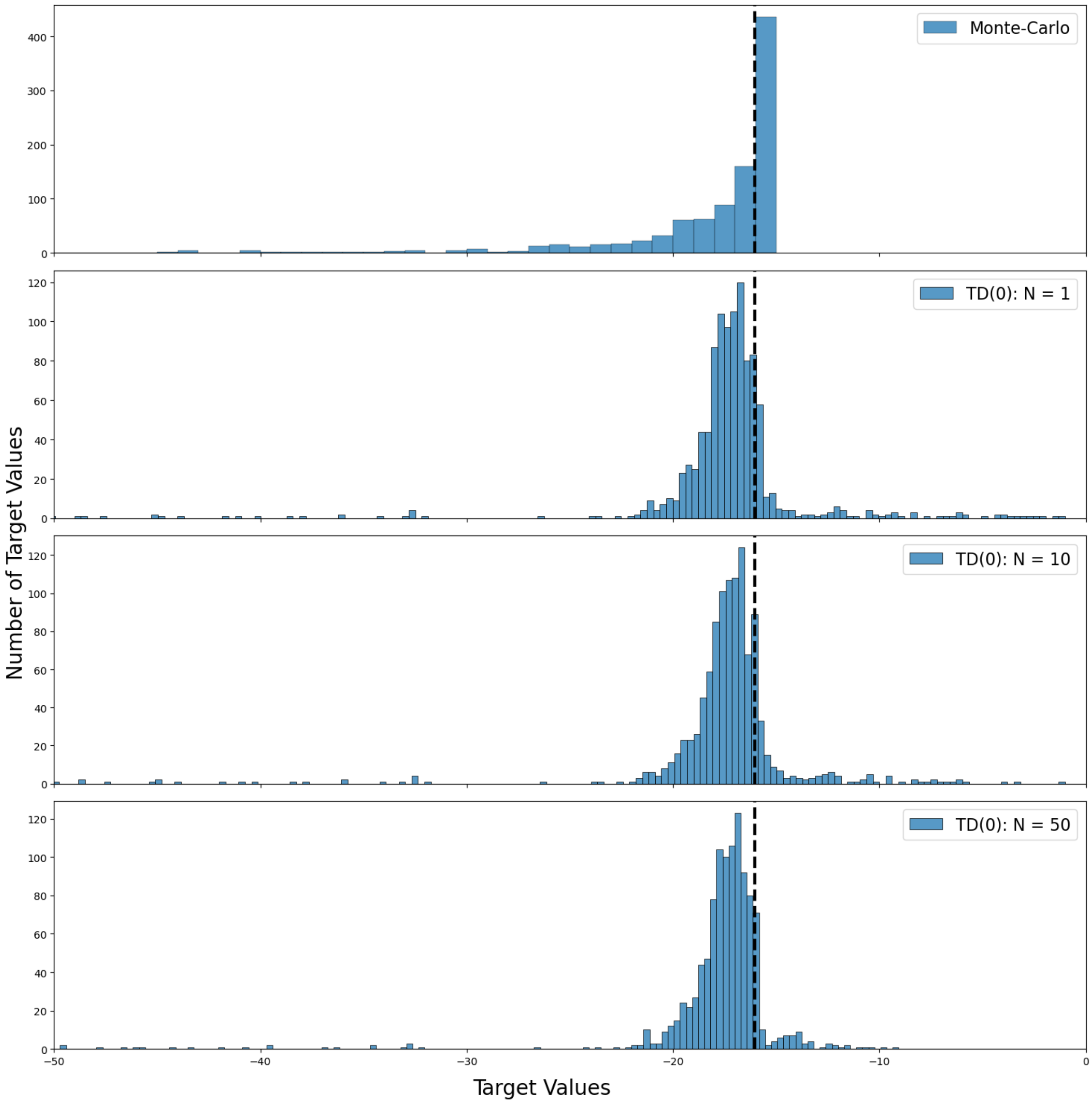
4. 4 points. (RL2e 6.9) Windy gridworld.






Windy GridWorld with King's movement and no movement
Sarsa
Expected Sarsa
Q-learning
Episodes
0
50
100
150
200
250
300
350
Time Steps
0
1000
2000
3000
4000
5000
6000
7000
8000

5. 2 points. [5180] Bias-variance trade-off.



(b) Written:

From my observation from the histograms:

1. Monte-Carlo method is unbiased but has a fat left tail due to its sampling nature and has a high-variance because each sample is i.i.d.
2. TD method is obviously biased but it's more clustered in the center and have a low variance.
3. After trained with more episodes. TD method performs better with lower variance.
4. Monte-Carlo method does not depend on the number of training because it's target G does not depend on other state values, it does not bootstrap.

(c) [Extra credit. 0.5 points]:

If we considered the scenario of control, Monte-Carlo method will still be unbiased and with relatively high variance. And on-policy TD(0) method will still suffer from it's bootstrap nature and mathematically affected by the parameter alpha. The result will not be changed.