hw 4 Solution

chenyu wu

Nov 2023

Exercise 5.6 (15/12 pts):

Let the $\mathcal{T}$ track the state-action pair so that we have

$$Q(s,a) \doteq \frac{\sum_{t\in\mathcal{T}(s,a)} \rho_{t:T(t)-1} G_t}{\sum_{t\in\mathcal{T}(s,a)} \rho_{t:T(t)-1}}.$$

Exercise 5.8 (15/12 pts):

For every visit, we need to compute

$$\mathbb{E}_b\left[\left(\frac{1}{T-1}\sum_{k=1}^{T-1}\prod_{t=0}^{k}\frac{\pi(A_t|S_t)}{b(A_t|S_t)}G_0.\right)^2\right]$$

Note that, here we only have one state so that we can simply divide by $T-1$. Then we have

$$\begin{aligned}
&\mathbb{E}_b\left[\left(\frac{1}{T-1}\sum_{k=1}^{T-1}\prod_{t=0}^{k}\frac{\pi(A_t|S_t)}{b(A_t|S_t)}G_0.\right)^2\right] \\
&= 0.5\cdot 0.1\cdot\left(\frac{1}{2}\right)^2 \\
&+ \frac{1}{2}\left[0.5\cdot 0.9\cdot 0.5\cdot 0.1\cdot\left(\frac{1}{2}\right)^{2\cdot 2} + 0.5\cdot 0.1\cdot\left(\frac{1}{2}\right)^2\right] \\
&+\cdots \\
&= 0.1\sum_{k=1}^{\infty}\frac{1}{k}\sum_{\ell=0}^{k-1}0.9^{\ell}\cdot 2^{\ell}\cdot 2 \\
&= 0.2\sum_{k=1}^{\infty}\frac{1}{k}\sum_{\ell=0}^{k-1}1.8^{\ell} = \infty
\end{aligned}$$

Exercise 6.7 (15/12 pts):

The TD(0) update is

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)].$$

Denote the A_t as the action take at the step t, we have

$$V(S_t) \leftarrow V(S_t) + \alpha[\frac{\pi(A_t|S_t)}{b(A_t|S_t)}(R_{t+1} + \gamma V(S_{t+1})) - V(S_t)].$$

Exercise 6.8 (15/12 pts):

$$\begin{aligned}
G(S_t, A_t) - Q(S_t, A_t) &= R_{t+1} + \gamma G(S_{t+1}, A_{t+1}) - Q(S_t, A_t) + \gamma Q(S_{t+1}, A_{t+1}) - \gamma Q(S_{t+1}, A_{t+1}) \\
&= \delta_t + \gamma(G(S_{t+1}, A_{t+1}) - Q(S_{t+1}, A_{t+1})) \\
&= \delta_t + \gamma\delta_{t+1} + \gamma^2(G(S_{t+1}, A_{t+1}) - Q(S_{t+1}, A_{t+1})) \\
&= \cdots \\
&= \sum_{k=t}^{T-1} \gamma^{k-t}\delta_k
\end{aligned}$$

Exercise 6.10 (25/20 pts):

See the attached code for reference.

Exercise 6.12 (15/12 pts):

No. Let's go back to these two algorithms:

We may focus on how these two algorithms select the actions. We denote the initial $Q_0(S, A)$, where the subscript denotes the iterative of $Q(S, A)$, i.e. $Q_1(S, A)$ is the value function after the first TD(0) update. Then we use the $A_i^{(j)}(S)$ to denote the action we take at step i, the superscript j means the action is taken based on the state-action value function $Q_j(S, A)$.

- **Salsa:** $Q_0(S, A) \to A_0^{(0)}(S) \to S' \to A_1^{(0)}(S') \to Q_1(S, A) \to \cdots$
- **Q-learning:** $Q_0(S, A) \to A_0^{(0)}(S) \to S' \to Q_1(S, A) \to A_1^{(1)}(S') \to \cdots$

We can see that in the Salsa, we update the Q after we select the next action, while in the Q-learning, we update the Q before we select the next action.

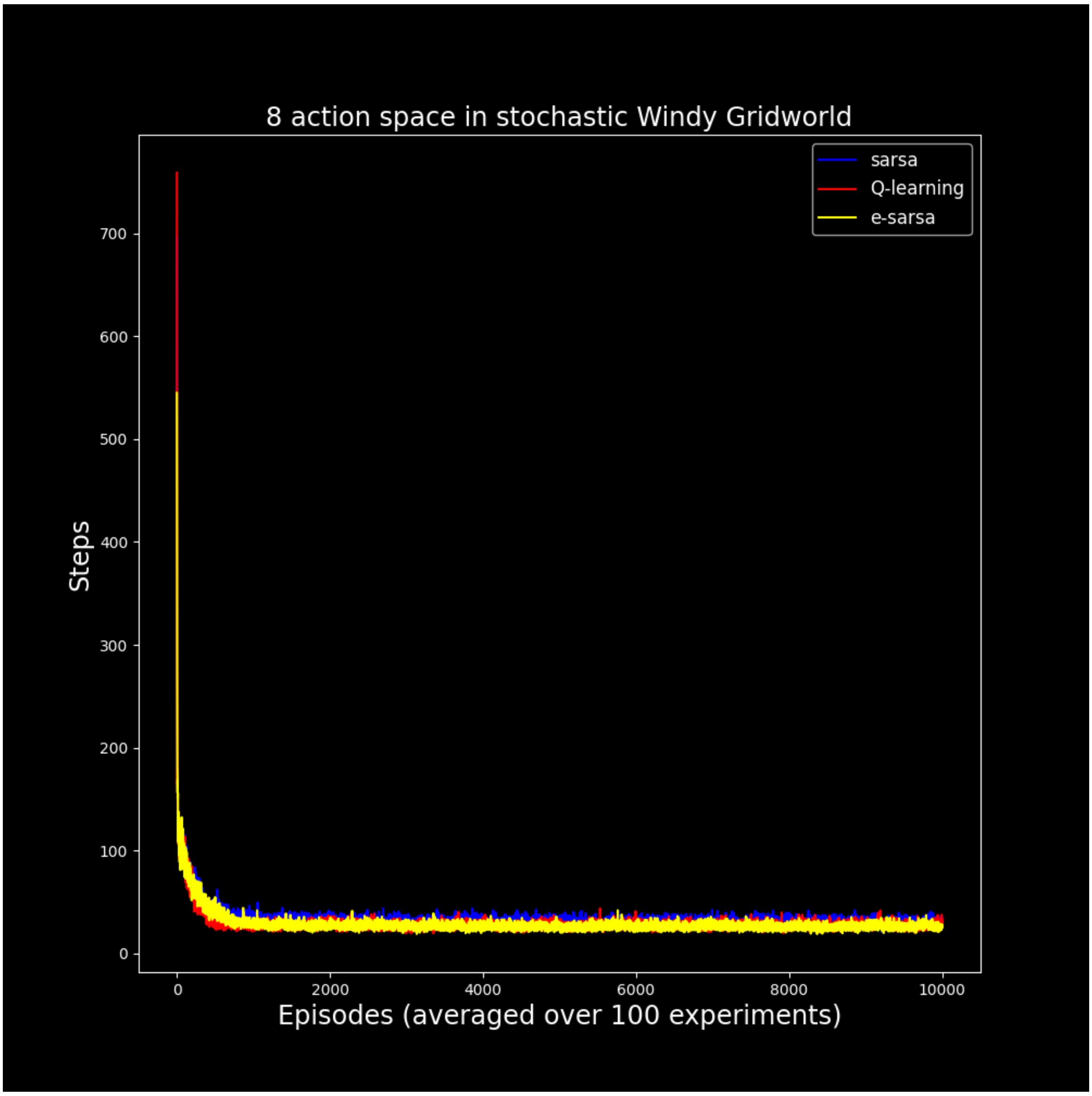

Figure 1: Fix epsilon

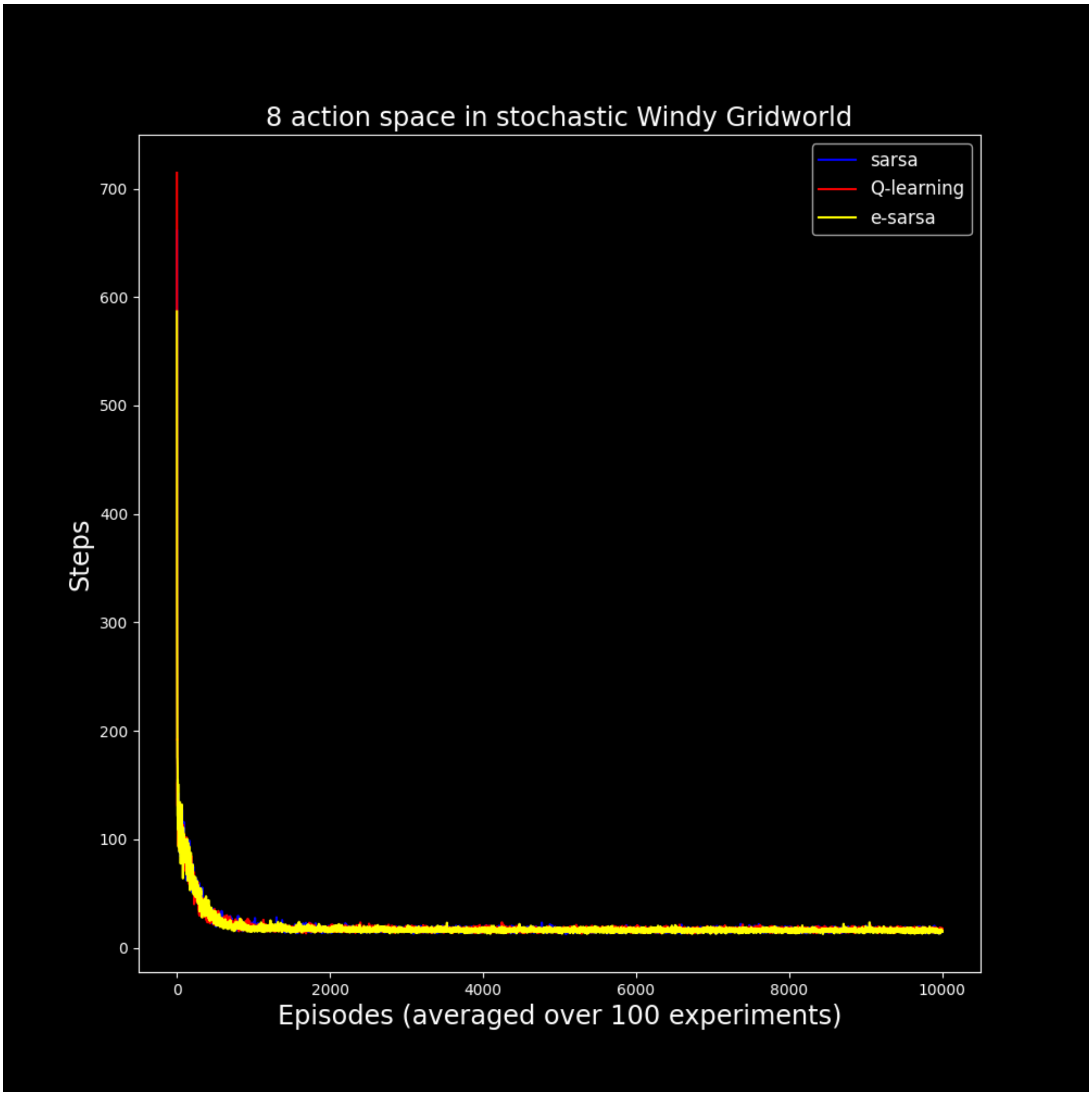

Figure 2: Dynamic epsilon

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

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Choose A from S using policy derived from Q (e.g., ε-greedy)
  Loop 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

Figure 3: Salsa

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

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, arbitrarily except that $Q(terminal, \cdot) = 0$

Loop for each episode:
  Initialize S
  Loop 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

Figure 4: Q-learning