Lecture 12 - Discrete Q-learning

Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

DDA4230: Reinforcement Learning
Course Page: [Click]

Model-based v.s. Model-free Algorithms

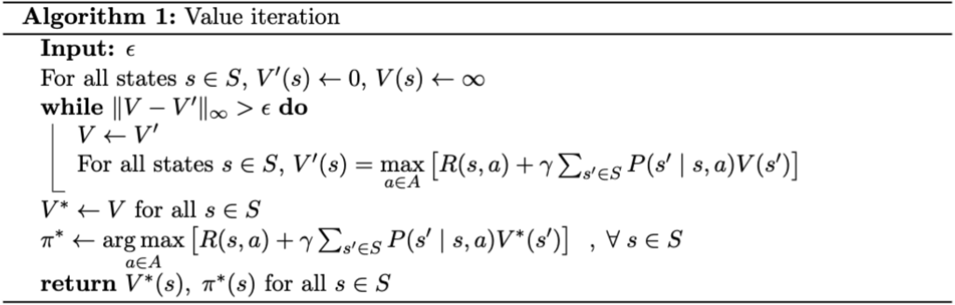
The model indicates the transition function and the reward function. This estimation could be in form of point estimation or distribution estimation like posterior sampling.

- Model-based Algorithm: maintains an estimate of the model and uses the model when interacting with the environment.
- Model-free Algorithm: does not estimate the world model.

When we do not have a reasonable estimation of the model (under large state and action spaces and continuous settings), an error will be induced by a wrongly estimated model as the model bias (maybe accumulate during learning).

Q-Learning

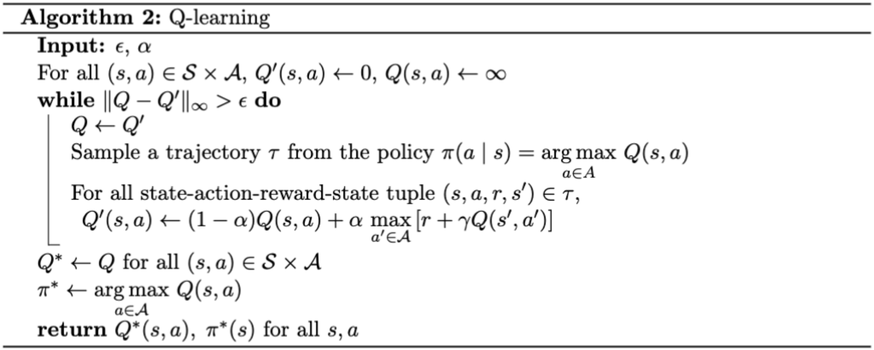
We start with the value iteration algorithm and discuss how the model could be lifted.



```
Algorithm 1: Value iteration
  Input: $\epsilon$
  For all states $s \in S$, $V'(s) \leftarrow 0$, $V(s) \leftarrow \infty$
  while $\|V - V'\|_\infty > \epsilon$ do
      $V \leftarrow V'$
      For all states $s \in S$, $V'(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V(s')\right]$
  $V^* \leftarrow V$ for all $s \in S$
  $\pi^* \leftarrow \arg\max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in S} P(s' \mid s,a) V^*(s')\right] \quad , \forall\, s \in S$
  return $V^*(s)$, $\pi^*(s)$ for all $s \in S$
```

Q-Learning

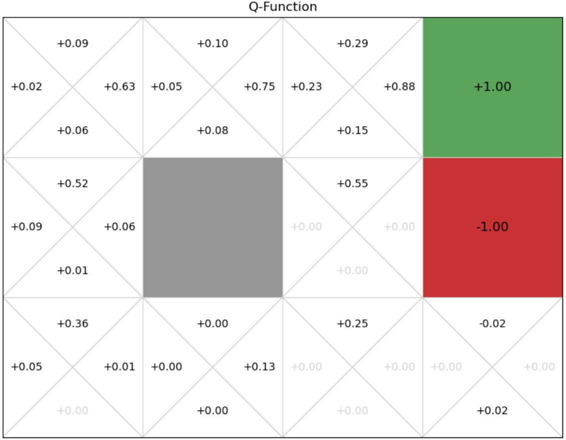
- The terms $\sum_{s' \in S} P(s' \mid s, a) V(s')$ and $\sum_{s' \in S} P(s' \mid s, a) V^*(s')$ could remove the dependency on P by representing the action values.
- Introducing the step size so that the update only takes at α portion of the action value while the $1 - \alpha$ portion of the action value remains the same.



```
Algorithm 2: Q-learning
Input: $\epsilon$, $\alpha$
For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q'(s,a) \leftarrow 0$, $Q(s,a) \leftarrow \infty$
while $\|Q - Q'\|_\infty > \epsilon$ do
    $Q \leftarrow Q'$
    Sample a trajectory $\tau$ from the policy $\pi(a \mid s) = \arg\max_{a \in A} Q(s,a)$
    For all state-action-reward-state tuple $(s,a,r,s') \in \tau$,
      $Q'(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha \max_{a' \in \mathcal{A}} [r + \gamma Q(s',a')]$
$Q^* \leftarrow Q$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
$\pi^* \leftarrow \arg\max_{a \in \mathcal{A}} Q(s,a)$
return $Q^*(s,a)$, $\pi^*(s)$ for all $s, a$
```

Q-Learning

Q Learning in Grid World.



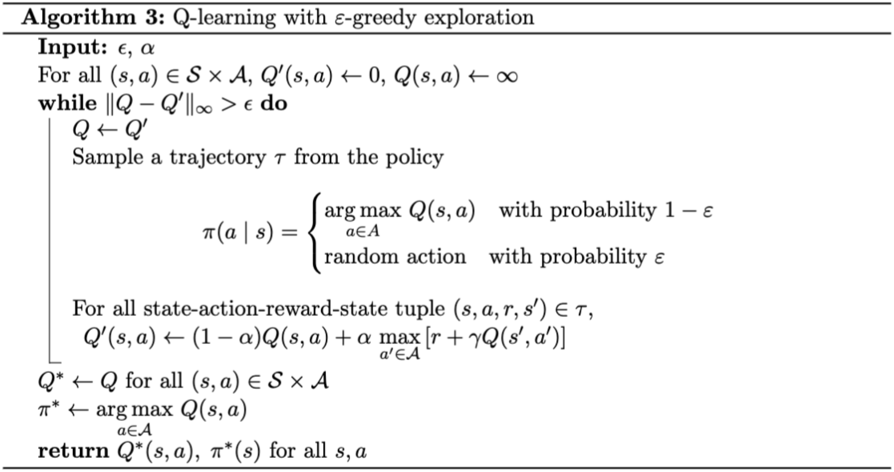
Exploration and ε-greedy Q-learning

In Q-learning, the trajectory sampled is subject to the current policy and thereof the current value estimation. However,

- It is possible that the algorithm is stuck at a suboptimal action value estimate and does not update itself.
- It is possible that some states are never explored with some initialization of the policy and value functions.

A simple way of involving exploration is to force the algorithm to select a random action with probability ε. This ε could delay over the iterations, as is in the ε-greedy algorithm for multi-armed bandits.

Exploration and ε-greedy Q-learning



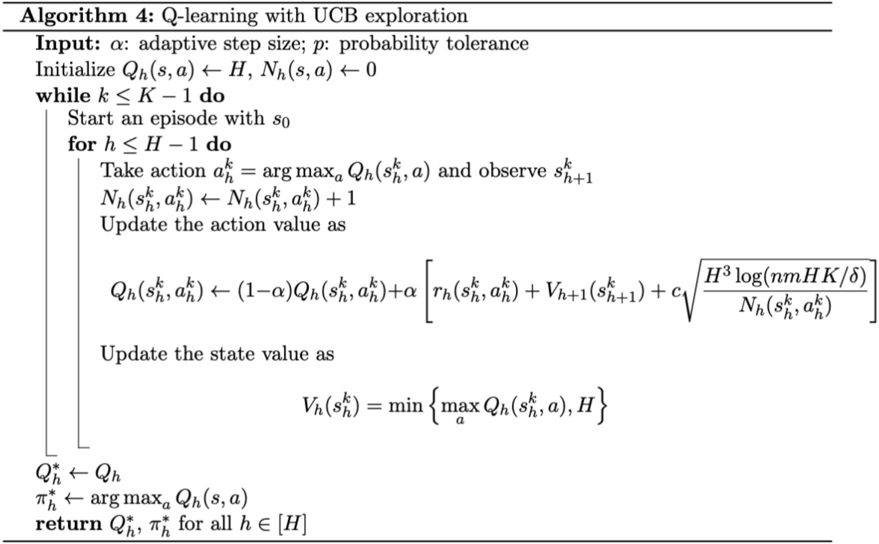
```
Algorithm 3: Q-learning with $\varepsilon$-greedy exploration
Input: $\epsilon$, $\alpha$
For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q'(s,a) \leftarrow 0$, $Q(s,a) \leftarrow \infty$
while $\|Q - Q'\|_\infty > \epsilon$ do
    $Q \leftarrow Q'$
    Sample a trajectory $\tau$ from the policy
        $$\pi(a \mid s) = \begin{cases} \arg\max_{a \in A} Q(s,a) & \text{with probability } 1 - \varepsilon \\ \text{random action} & \text{with probability } \varepsilon \end{cases}$$
    For all state-action-reward-state tuple $(s,a,r,s') \in \tau$,
     $Q'(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha \max_{a' \in \mathcal{A}} [r + \gamma Q(s',a')]$
$Q^* \leftarrow Q$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
$\pi^* \leftarrow \arg\max_{a \in \mathcal{A}} Q(s,a)$
return $Q^*(s,a)$, $\pi^*(s)$ for all $s, a$
```

Q-learning with UCB

In spite of simplicity, ε-greedy Q-learning does not have a rigorous regret guarantee.

- We present another variant of Q-learning with UCB exploration. This algorithm is the first Q-learning variant that has a rigorous regret guarantee of $\sqrt{K}$.
- We again use $Q_h(s,a)$ as the time-dependent action-value function, which is necessary when the horizon of each episode is constant.

Q-learning with UCB



Algorithm 4: Q-learning with UCB exploration

```
Input: $\alpha$: adaptive step size; $p$: probability tolerance
Initialize $Q_h(s,a) \leftarrow H$, $N_h(s,a) \leftarrow 0$
while $k \leq K-1$ do
    Start an episode with $s_0$
    for $h \leq H-1$ do
        Take action $a_h^k = \arg\max_a Q_h(s_h^k, a)$ and observe $s_{h+1}^k$
        $N_h(s_h^k, a_h^k) \leftarrow N_h(s_h^k, a_h^k) + 1$
        Update the action value as
            $Q_h(s_h^k,a_h^k) \leftarrow (1-\alpha)Q_h(s_h^k,a_h^k)+\alpha\left[r_h(s_h^k,a_h^k)+V_{h+1}(s_{h+1}^k)+c\sqrt{\frac{H^3\log(nmHK/\delta)}{N_h(s_h^k,a_h^k)}}\right]$
        Update the state value as
            $V_h(s_h^k) = \min\left\{\max_a Q_h(s_h^k,a), H\right\}$
$Q_h^* \leftarrow Q_h$
$\pi_h^* \leftarrow \arg\max_a Q_h(s,a)$
return $Q_h^*$, $\pi_h^*$ for all $h \in [H]$
```

Q-learning with UCB

Theorem

By choosing $\alpha = \frac{H+1}{H+N}$ *with the visitation count* $N = N_h(s_h^k, a_h^k)$, *there exists an absolute constant c such that with probability at least* $1-\delta$ *the regret of Q-learning with UCB exploration is at most* $O(\sqrt{nmH^5K\log(nmHK/\delta)})$.

The proof relies on the cast of the variables into a filtration and therefore the use of the Azuma-Hoeffding inequality (introduced in LN3). For those students that are interested in the proof we could host you with a presentation of it.

Question and Answering (Q&A)