Model-free Control

The model-free control problem is about learning a close-to optimal policy π given experience samples $(s, a, r, s'))$
Good thing about $Q(s, a)$ is that to do policy improvement, we don't need the model:

$$\pi^*(a|s) = argmax_{a \in A} Q(s, a)$$

Bellman Optimality for Q

$$Q^{\pi^*(s,a)} = R(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) max_{a \in A} Q^{\pi^*}(s', a')$$

Greedy policy improvement has some problems:

1. Non-optimal initial choices may disorient exploration
2. Areas of the state space remain unexplored
   Simple idea to balance exploration and achieving rewards it to use ϵ-greedy policy with respect to a state-action value $Q(s, a)$. In words, select argmax action with probability $1 - \epsilon$, else select action uniformly at random.

Policy Improvement theorem showed that policy iteration given dynamics and reward models was guaranteed to monotonically improve. This proof assumed that policy improvement output a deterministic policy. The same property holds for ϵ-greedy policies.

Monotonic ϵ- greedy Policy Improvement

For any ϵ-greedy policy π_i the ϵ-greedy policy with respect to Q^{π_i}, π_{i+1} is a monotonic improvement $V^{\pi_{i+1}} \geq V^{\pi_i}$

Greedy in the Limit of Infinite Exploration (GLIE)

All state-action pairs are visited an infinite number of times:

$$lim_{i \to \infty} N_i(s, a) \to \infty$$

Behaviour policy converges to greedy policy. GLIE MC control converges to the optimal state-action value function $Q(s, a) \to Q^*(s, a)$

TD control

We have two variants of TD control: SARSA and Q-learning
SARSA(on-policy):

1. Initialize arbitrary $Q(s, a)$ $\forall a \in S$, $\forall s \in S$, except the terminal state $Q(terminal, *) = 0$

2. Repeat (for each episode):
   2.1 Choose $a = argmax Q_\pi(s,a)$ $\epsilon-$ greedy
   2.2 Repeat (for each step):
   2.2.1. Take action $a_{t+1} \sim \pi(s_{t+1})$
   2.2.2 Observe r_{t+1}, s_{t+2}
   2.2.3 $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t))$
   2.2.4 $s \leftarrow s_{t+1}$, $a \leftarrow a_{t+1}$
   until S is terminal

On-policy vs Off-policy

On-policy: learn while doing the job. Learn about policy π from experience sampled with π
Off-policy: look over someone's shoulder. Learn about policy π from experience sampled with μ .
Why is off-policy learning a good idea?

- can learn from demonstrations
- can re-use all experience
- learn about **optimal** policy while following **exploratory** policy
  SARSA is on-policy learning algorithm that learns to estimate and evaluate a policy from experience obtained from following that policy.
  On the other hand, Q-values are updated assuming a greedy-policy were followed despite the fact that the current policy was not greedy.

Q-learning:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha((r_t + \gamma max_{a'} Q(s_{t+1},a')) - Q(s_t,a_t))$$

SARSA chooses the safe route because SARSA incorporates the current policy ($\epsilon-$greedy)
Q-learning chooses the optimal path and falls of the cliff on CliffWalk environment.

Convergence

SARSA converges to the optimal action-value function, under the following conditions:

- GLIE sequence of policies $\pi_t(a|s)$ (decay of exploration rate)
- Robbins-Monro sequence of step-sizes α_t:

$$\sum_{t=1}^{\infty} \alpha_t = \infty$$

$$\sum_{t=1}^{\infty} \alpha_t^2 < \infty$$