SARSA(λ): Learn function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Require:

Sates $\mathcal{S} = \{1, \ldots, n_s\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}$
Reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
$\lambda \in [0, 1]$: Trade-off between TD and MC
procedure SARSA-LAMBDA($\mathcal{S}$, A, R, T, α, γ, λ)
    Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily
    Initialize $e : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ with 0. ▷ eligibility trace
    Start in state $s_0 \in \mathcal{S}, s \leftarrow s_0$
    while Q is not converged **do**
        Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily except set terminal states zero.
        while s is not terminal **do**
            $r \leftarrow R(s, a)$
            $s' \leftarrow T(s, a)$ ▷ Receive the new state
            Calculate π based on Q (e.g. epsilon-greedy)
            $a' \leftarrow \pi(s')$
            $e(s, a) \leftarrow e(s, a) + 1$
            $\delta \leftarrow r + \gamma \cdot Q(s', a') - Q(s, a)$ ▷ Temporal Difference
            for $(\tilde{s}, \tilde{a}) \in \mathcal{S} \times \mathcal{A}$ **do**
                $Q(\tilde{s}, \tilde{a}) \leftarrow Q(\tilde{s}, \tilde{a}) + \alpha \cdot \delta \cdot e(\tilde{s}, \tilde{a})$
                $e(\tilde{s}, \tilde{a}) \leftarrow \gamma \cdot \lambda \cdot e(\tilde{s}, \tilde{a})$ ▷ Update eligibility trace
            $s \leftarrow s'$
            $a \leftarrow a'$
    return Q

Expect-SARSA: Learn function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Require:

Sates $\mathcal{S} = \{1, \ldots, n_s\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}$
Reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
procedure EXPECT-SARSA($\mathcal{S}$, A, R, T, α, γ, λ)
    Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily except set terminal states zero.
    Start in state $s_0 \in \mathcal{S}, s \leftarrow s_0$
    while Q is not converged **do**
        Select $(s_0, a) \in \mathcal{S} \times \mathcal{A}$ arbitrarily
        while s is not terminal **do**
            $r \leftarrow R(s, a)$ ▷ Receive the reward
            $s' \leftarrow T(s, a)$ ▷ Receive the new state
            Calculate π based on Q (e.g. epsilon-greedy)
            $a' \leftarrow \pi(s')$
            $Q(s, a) \leftarrow Q(s, a) + \alpha \cdot (r + \gamma \sum_a \pi(a|s') \cdot Q(s', a) - Q(s, a))$
            $s \leftarrow s'$
            $a \leftarrow a'$
    return Q

SARSA: Learn function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Require:

Sates $\mathcal{S} = \{1, \ldots, n_s\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}$
Reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
procedure SARSA($\mathcal{S}$, A, R, T, α, γ, λ)
    Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily except set terminal states zero.
    Start in state $s_0 \in \mathcal{S}, s \leftarrow s_0$
    while Q is not converged **do**
        Select $(s_0, a) \in \mathcal{S} \times \mathcal{A}$ arbitrarily
        while s is not terminal **do**
            $r \leftarrow R(s, a)$ ▷ Receive the reward
            $s' \leftarrow T(s, a)$ ▷ Receive the new state
            Calculate π based on Q (e.g. epsilon-greedy)
            $a' \leftarrow \pi(s')$
            $Q(s, a) \leftarrow Q(s, a) + \alpha \cdot (r + \gamma Q(s', a') - Q(s, a))$
            $s \leftarrow s'$
            $a \leftarrow a'$
    return Q

Q-learning: Learn function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$

Require:

States $\mathcal{S} = \{1, \ldots, n_s\}$
Actions $\mathcal{A} = \{1, \ldots, n_a\}$
Reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
Black-box (probabilistic) transition function $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
Learning rate $\alpha \in [0, 1]$, typically $\alpha = 0.1$
Discounting factor $\gamma \in [0, 1]$
procedure QLearning($\mathcal{S}$, A, R, T, α, γ)
    Initialize $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arbitrarily except set terminal states zero.
    Start in state $s_0 \in \mathcal{S}, s \leftarrow s_0$
    while Q is not converged **do**
        Start in state $s \in \mathcal{S}$
        while s is not terminal **do**
            Calculate π based on Q (e.g. epsilon-greedy)
            $a \leftarrow \pi(s)$
            $r \leftarrow R(s, a)$ ▷ Receive the reward
            $s' \leftarrow T(s, a)$ ▷ Receive the new state
            $Q(s', a) \leftarrow Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a') - Q(s, a))$
            $s \leftarrow s'$
    return Q