SARSA 值迭代方法：原理、公式、应用与实战

2025 年 9 月 21 日

1 引言

SARSA（State-Action-Reward-State-Action）是典型的在线策略（on-policy）时间差分强化学习算法。与 Q-learning 的“乐观”最大化不同，SARSA 使用策略实际执行的下一动作更新价值函数，因此能够更好地反映探索策略下的期望表现。

2 原理与公式

2.1 在线策略动作价值函数

SARSA 估计当前策略 π 下的动作价值函数 $Q(s,a)$，其贝尔曼方程为：

$$Q^{\pi}(s,a) = \mathbb{E}\big[r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) \mid s_t = s, a_t = a\big], \tag{1}$$

其中 $a_{t+1} \sim \pi(\cdot \mid s_{t+1})$。

2.2 更新规则

完成一次交互 $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ 后，SARSA 更新为：

$$Q_{t+1}(s_t,a_t) \leftarrow Q_t(s_t,a_t) + \alpha_t \Big[r_{t+1} + \gamma Q_t(s_{t+1},a_{t+1}) - Q_t(s_t,a_t)\Big]. \tag{2}$$

通常采用 ε-贪心策略选择动作，上式中的 a_{t+1} 即为该策略在下一状态给出的动作。

2.3 收敛性质

只要学习率满足 $\sum_t \alpha_t = \infty$、$\sum_t \alpha_t^2 < \infty$，且策略确保所有状态-动作对被无限次访问，在有限 MDP 中 SARSA 可收敛到最优 ε-贪心策略。由于更新考虑了探索动作，SARSA 在存在危险区域时通常比 Q-learning 更保守。

3 应用与技巧

- **随机控制：**如“悬崖行走”环境中，探索动作可能导致危险后果。
- **机器人任务：**结合迹（eligibility trace）的 SARSA(λ) 能学习平滑策略，应对传感器噪声。
- **教学/训练模拟：**需要策略显式考虑探索的场景。
- **实用建议：**合理衰减 ε，对比 Q-learning 观察风险偏好差异，监控多次实验的方差。

4 Python 实战

脚本 `gen_sarsa_figures.py` 在含“悬崖”惩罚的随机网格世界中训练 SARSA，绘制回报曲线与最终状态价值图。

Listing 1: 脚本 $\mathrm{gen}_{s}arsa_{f}igures.py$

```
for episode in range(num_episodes):
    state = env.reset()
    action = epsilon_greedy(Q[state], epsilon)
    done = False
    G = 0.0
    while not done:
        next_state, reward, done = env.step(state, action)
        next_action = epsilon_greedy(Q[next_state], epsilon)
        td_target = reward + gamma * Q[next_state, next_action] * (1.0
            - float(done))
        Q[state, action] += alpha * (td_target - Q[state, action])
        state, action = next_state, next_action
        G += reward
    returns.append(G)
```

5 实验结果

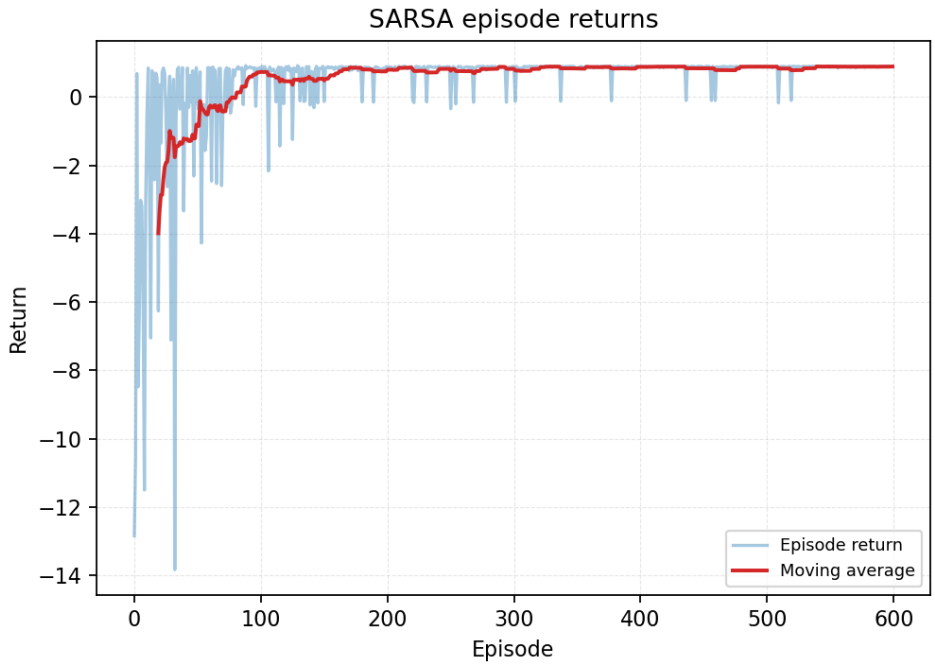

图 1: SARSA 回报曲线，展示 ε-贪心策略下的收敛趋势

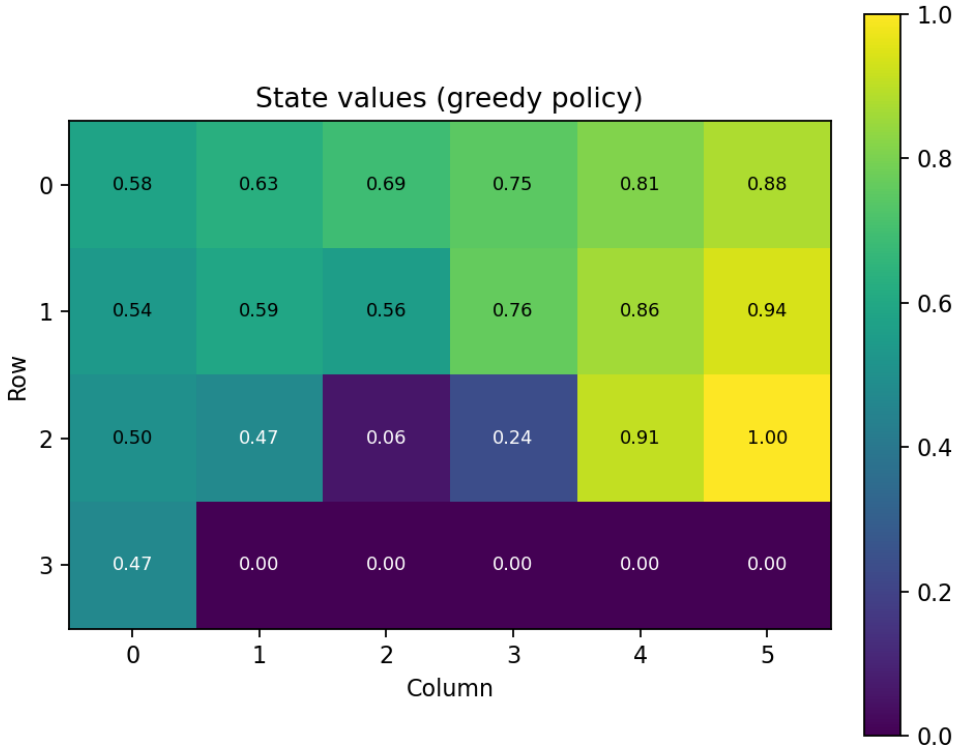

图 2: 最终状态价值热力图，体现算法在危险区域的保守策略

6 总结

SARSA 将探索行为纳入更新目标，适合风险敏感的强化学习任务。通过调节学习率与探索策略，可实现稳定收敛。示例演示了回报随训练稳定提升，以及学习到的价值函数如何回避“悬崖”区域。