@zblu Reinforcement Learning Theory

INF8250AE Course Notes

RL 6: Learning from a Stream of Data 2: Temporal Difference Learning

TD Learning, Empirical Bellman Operator, TD Error, SARSA, Q-Learning, SA Convergence

Key Topics: TD Learning, Empirical Bellman Operator, TD Error, SARSA, Q-Learning, SA Convergence
Reference: `https://amfarahmand.github.io/IntroRL/`

Notes on Reinforcement Learning – Nov 17, 2025

Reinforcement Learning 6: Learning from a Stream of Data 2: Temporal Difference Learning

Zhuobie

2025 年 11 月 18 日

1. Temporal-Difference (TD) Learning

Monte Carlo (MC) learning estimates $V^\pi(x)$ by averaging complete returns

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots.$$

TD Learning replaces the *return* G_t with an *empirical Bellman operator* that uses a **one-step sample** and the **current estimate** V_t:

$$\hat{T}V_t(x) \coloneqq R_t + \gamma V_t(X_{t+1}),$$

which is an unbiased estimator of the Bellman operator

$$(T^\pi V)(x) = r^\pi(x) + \gamma \mathbb{E}[V(X') \mid X = x].$$

Monte Carlo vs. TD (key distinction).

- MC uses the *full return* G_t as a sample of the true value.
- TD uses the *empirical Bellman operator* $\hat{T}V_t$ as a one-step bootstrap target.

MC waits until termination to update; TD updates online at every step.

1.1 TD Learning for Policy Evaluation

The TD(0) update is

$$\boxed{V_{t+1}(X_t) = V_t(X_t) + \alpha_t\big[R_t + \gamma V_t(X_{t+1}) - V_t(X_t)\big].}$$

Define the **TD error**

$$\delta_t = R_t + \gamma V_t(X_{t+1}) - V_t(X_t).$$

Then the TD update becomes the SA recursion

$$V_{t+1}(X_t) = V_t(X_t) + \alpha_t \delta_t.$$

1.2 TD Learning Error: TD Error vs Bellman Residual

The true Bellman residual of a value estimate V is:

$$\mathrm{BR}^{\pi}(V)(x) = (T^{\pi}V)(x) - V(x).$$

TD uses δ_t as an unbiased estimator:

$$\mathbb{E}[\delta_t \mid X_t = x] = \mathrm{BR}^{\pi}(V_t)(x).$$

Remark

TD error is a *sample* of the Bellman residual. TD learning reduces the Bellman residual in expectation, and converges to V^{π} under standard SA assumptions.

2. TD Learning for Control

control problem: policy evaluation + policy improvement.

In value prediction we evaluate a *fixed* policy π. In TD control, however, the goal is to *improve* the policy and eventually obtain the optimal one. Therefore TD control methods (SARSA, Q-learning) operate under the paradigm of **Generalized Policy Iteration (GPI)**:

$$\pi_t \longrightarrow Q_t \longrightarrow \pi_{t+1}.$$

Why does TD control use a time-varying policy π_t?
Although the policy used at each step is typically greedy or ε-greedy, it is defined *with respect to the current estimate* Q_t:

$$\pi_t = \varepsilon\text{-greedy}(Q_t).$$

Since Q_t is updated at every step, the greedy action changes accordingly. Thus the behaviour policy evolves during learning:

$$\pi_1,\ \pi_2,\ \pi_3, \ldots$$

TD control is therefore not evaluating a single fixed policy, but rather a *sequence of increasingly improved policies*.

This leads to the distinction between:

- **On-policy control (SARSA):** evaluates and improves the same behaviour policy π_t used to collect data.
- **Off-policy control (Q-learning):** learns the greedy policy $\pi_g(Q_t)$ while data may come from any exploratory behaviour policy.

2.1 On-Policy TD Control (SARSA)

SARSA is named after its transition tuple

$$(S_t, A_t, R_t, S_{t+1}, A_{t+1}).$$

The update:

$$\boxed{Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha_t\big[R_t + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t)\big].}$$

Algorithm 1: SARSA Algorithm (on-policy TD control)

Input: Step-size schedule $(\alpha_t)_{t\geq 1}$; policy family $(\pi_t)_{t\geq 1}$ (e.g. ε-greedy).
Output: Action-value function $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.

Initialize $Q(x, a)$ arbitrarily, e.g. $Q(x, a) = 0$;
Sample initial state $X_1 \sim \rho$;
Choose initial action $A_1 \sim \pi_1(\cdot \mid X_1)$;
for *each step* $t = 1, 2, \ldots$ **do**
  Take action A_t, observe R_t and X_{t+1};
  Choose next action $A_{t+1} \sim \pi_t(\cdot \mid X_{t+1})$;
  `// Update rule`
  $Q_{t+1}(X_t, A_t) \leftarrow (1 - \alpha_t(X_t, A_t))\, Q_t(X_t, A_t)$;
  $\quad + \alpha_t(X_t, A_t)\big[R_t + \gamma Q_t(X_{t+1}, A_{t+1})\big]$;

On-policy: The update target uses the *same policy* that is used to generate actions. Thus SARSA evaluates and improves the exploration policy (e.g., ε-greedy).

2.2 Off-Policy TD Control (Q-learning)

Q-learning uses the Bellman *optimality* target:

$$\boxed{Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha_t\big[R_t + \gamma \max_{a'} Q_t(S_{t+1}, a') - Q_t(S_t, A_t)\big].}$$

Algorithm 2: Q-Learning Algorithm (off-policy TD control)

Input: Step-size schedule $(\alpha_t)_{t\geq 1}$; behaviour policy mechanism π.
Output: Action-value function $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.

Initialize $Q(x, a)$ arbitrarily, e.g. $Q(x, a) = 0$;
Sample initial state $X_1 \sim \rho$;
for *each step* $t = 1, 2, \ldots$ **do**
  Sample action $A_t \sim \pi(\cdot \mid X_t)$;
  Take action A_t, observe R_t and X_{t+1};
  `// Update rule`
  $Q_{t+1}(X_t, A_t) \leftarrow Q_t(X_t, A_t)$;
  $\quad + \alpha_t(X_t, A_t)\Big[R_t + \gamma \max_{a' \in \mathcal{A}} Q_t(X_{t+1}, a') - Q_t(X_t, A_t)\Big]$;

Off-policy: Actions may come from any behavior policy (exploration), but the update always moves toward the greedy policy. Thus Q-learning learns $\pi^\star$ while behaving non-greedily.

3. Relation Between SARSA and Q-Learning

- SARSA: **Expected return under the current behavior policy**. → converges to Q^π where π is the limiting policy.
- Q-learning: **Expected return under the greedy policy**. → converges to $Q^\star$, independent of behavior policy.

Remark

When ε-greedy exploration persists, SARSA learns the value of the *stochastic* ε-greedy policy, while Q-learning learns the *deterministic optimal* greedy policy.

4. Stochastic Approximation View and Convergence

Both TD and Q-learning are instances of the asynchronous SA recursion:

$$\theta_{t+1}(i) = (1 - \alpha_t(i))\theta_t(i) + \alpha_t(i)\big[(L\theta_t)(i) + \eta_t(i)\big].$$

Where:

$$L = T^\pi \quad \text{(TD)}, \qquad L = T^\star \quad \text{(Q-learning)}.$$

4.1 Noise Assumptions

Let $\mathcal{F}_t$ be the σ-field before observing (R_t, X_{t+1}).

- Zero-mean noise:
$$\mathbb{E}[\eta_t(i) \mid \mathcal{F}_t] = 0.$$
- Bounded conditional variance:
$$\mathbb{E}[\eta_t(i)^2 \mid \mathcal{F}_t] \le c_1 + c_2\|\theta_t\|^2.$$

4.2 Step-Size and Visit Frequency

For each component i:

$$\sum_t \alpha_t(i) = \infty, \qquad \sum_t \alpha_t(i)^2 < \infty,$$

and every state(-action) pair is visited infinitely often.

4.3 Contraction of Bellman Operators

$$\|T^\pi V_1 - T^\pi V_2\|_\infty \le \gamma\|V_1 - V_2\|_\infty,$$
$$\|T^\star Q_1 - T^\star Q_2\|_\infty \le \gamma\|Q_1 - Q_2\|_\infty.$$

Both are γ-contractions and have unique fixed points V^π and $Q^\star$.

4.4 SA Convergence Theorem (Bertsekas–Tsitsiklis)

Under the above assumptions:

$$\theta_{t+1}(i) = (1 - \alpha_t(i))\theta_t(i) + \alpha_t(i)\big[(L\theta_t)(i) + \eta_t(i)\big]$$

converges almost surely to $\theta^\star$, the unique fixed point of L.

Implications for RL.

- TD(0) converges a.s. to V^π.
- SARSA converges a.s. to $Q^{\pi_{\varepsilon\text{-greedy}}}$.
- Q-learning converges a.s. to the optimal $Q^\star$.