# From Discrete Bellman to Continuous-Time HJB and Reinforcement Learning

Bellman $\rightarrow$ HJB $\leftrightarrow$ RL


Fenghui Yu

TU Delft


First Meeting of the Dutch Sequential Decision-Making Community

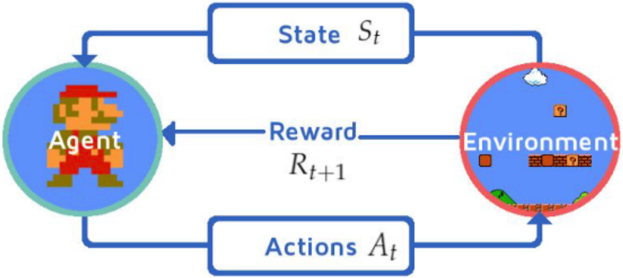
Eindhoven, August 28th, 2025

# Contents

# Introduction to reinforcement learning

# The basics of reinforcement learning

- **Goal:** automate goal-directed learning and decision-making.
- **Setup:** an agent interacts with an environment via *states* $s_t$, *actions* $a_t$, and *rewards* $r_{t+1}$.
- **Objective:** learn a policy $\pi(a \mid s)$ that maximizes long-term return.



# Value function

- If we consider infinite time horizon with discounted reward where $\gamma \in [0, 1)$, and $\mathbb{E}^{\Pi}$ denotes the expectation under the policy $\Pi$, $V^*$ for each $s \in \mathcal{S}$ to be

$$V^*(s) = \sup_{\Pi} V^{\Pi}(s) := \sup_{\Pi} \mathbb{E}^{\Pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s\right],$$

subject to

$$s_{t+1} \sim P(s_t, a_t), \quad a_t \sim \pi_t(s_t).$$

- The problem with finite time horizon can be expressed as

$$V^{\star}(s) = \sup_{\Pi} V^{\Pi}(s) := \sup_{\Pi} \mathbb{E}^{\Pi}\left[\sum_{t=0}^{T-1} r_t(s_t, a_t) + r_T(s_T) \middle| s_0 = s\right], \ \forall s \in \mathcal{S},$$

subject to

$$s_{t+1} \sim P_t(s_t, a_t), \quad a_t \sim \pi_t(s_t), \ \ 0 \le t \le T-1.$$

# Bellman equation for the $Q$-function

- DPP $\Rightarrow$ Bellman optimality:

$$V^*(s) = \max_{a \in \mathcal{A}} \mathbb{E}\left[ r(s,a) + \gamma V^*(s') \,\middle|\, s, a \right], \quad s' \sim P(\cdot \mid s, a).$$

- $Q$-function:

$$Q^*(s,a) = \mathbb{E}\left[ r(s,a) + \gamma V^*(s') \,\middle|\, s, a \right], \qquad V^*(s) = \max_a Q^*(s,a).$$

- Interpretation: $Q^*(s,a)$ = one-step reward + discounted next-state value; $\pi^*(s) = \arg\max_a Q^*(s,a)$.

# Example: $Q$-learning

- Value-based RL to learn $Q^*$ by bootstrapping Bellman optimality from samples $(s, a, r, s')$.
- Update at iteration $n$:

$$Q_{n+1}(s,a) \leftarrow (1-\alpha_n) \underbrace{Q^n(s,a)}_{\text{current estimate}} + \alpha_n \underbrace{\left[ r(s,a) + \gamma \max_{a'} Q^n(s',a') \right]}_{\text{new estimate}},$$

  where $\alpha_n$ is the learning rate.
- Policy: $\pi_{n+1}(s) = \arg\max_a Q_{n+1}(s,a)$ (use $\varepsilon$-greedy for exploration).

# Bellman $\rightarrow$ HJB $\leftrightarrow$ RL

# Overview

- **Goal:** Show how the discrete Bellman equation limits to the continuous-time HJB, and how core RL updates are sample-based solvers of that PDE.
- **Three links:**
  1. Discrete Bellman (semi-Lagrangian) $\Rightarrow$ HJB via Itô–Taylor expansion.
  2. TD/Q-learning errors $\Rightarrow$ HJB residual / Hamiltonian.
  3. Policy iteration / actor–critic $\Rightarrow$ policy improvement for HJB (incl. soft variants).

# Bellman $\to$ HJB

# Controlled diffusion and objective

Continuous-time controlled SDE:

$$dX_t = \mu(X_t, a_t)\,dt + \sigma(X_t, a_t)\,dW_t, \quad X_t \in \mathbb{R}^d,\ a_t \in \mathcal{A}.$$

Discounted infinite-horizon return:

$$J^\pi(x) = \mathbb{E}_x^\pi\left[\int_0^\infty e^{-\rho t}\, r(X_t, a_t)\,dt\right], \quad \rho > 0.$$

Value function: $V(x) = \sup_\pi J^\pi(x)$.

Controlled generator for smooth $f$:

$$(\mathcal{L}^a f)(x) = \mu(x,a)\cdot\nabla f(x) + \tfrac{1}{2}\,\mathrm{Tr}\big(\sigma\sigma^\top(x,a)\,\nabla^2 f(x)\big).$$

# Discrete one-step Bellman (semi-Lagrangian form)

Time step $h > 0$, per-step discount $\gamma_h = e^{-\rho h} = 1 - \rho h + o(h)$. The discrete Bellman equation (semi-Lagrangian form) is

$$V(t,x) = \sup_{a \in \mathcal{A}} \mathbb{E}\big[\, r(x,a)\, h + \gamma_h\, V\big(t+h, X_{t+h}^{x,a}\big)\big]\,.$$

One Euler step:

$$X_{t+h}^{x,a} = x + \mu(x,a)\, h + \sigma(x,a)\sqrt{h}\,\xi, \qquad \xi \sim \mathcal{N}(0, I).$$

# Itô–Taylor expansion and cancellation

Expand $V$ to $O(h)$ and take expectation:

$$\begin{aligned}\mathbb{E}\big[V(t+h, X^{x,a}_{t+h})\big] &= V(t,x) + h\,\partial_t V(t,x) + h\,\nabla V(t,x)^\top \mu(x,a) \\ &\quad + \frac{h}{2}\,\operatorname{Tr}\big(\sigma\sigma^\top(x,a)\,\nabla^2 V(t,x)\big) + o(h).\end{aligned}$$

Plug into Bellman, subtract $V(t,x)$, divide by $h$, then let $h \downarrow 0$:

$$0 = \sup_{a\in\mathcal{A}} \Big\{ r(x,a) + \partial_t V + \nabla V \cdot \mu(x,a) + \tfrac{1}{2}\operatorname{Tr}\big(\sigma\sigma^\top(x,a)\,\nabla^2 V\big) - \rho\, V \Big\}.$$

# HJB in generator/Hamiltonian form

Generator form:

$$\boxed{\partial_t V(t,x) + \sup_a \{r(x,a) + (\mathcal{L}^a V)(t,x)\} - \rho V(t,x) = 0.}$$

Hamiltonian $H(x,p,M) = \sup_a \{r(x,a) + \mu(x,a)\cdot p + \frac{1}{2}\operatorname{Tr}(\sigma\sigma^\top(x,a)M)\}$:

$$\partial_t V + H(x, \nabla V, \nabla^2 V) - \rho V = 0.$$

Infinite-horizon stationary case (no $t$-dependence):

$$\boxed{\sup_a \{r(x,a) + (\mathcal{L}^a V)(x)\} - \rho V(x) = 0.}$$

# HJB $\leftrightarrow$ RL

# TD error $\approx$ HJB residual

Temporal Difference (TD) error with step $h$:

$$\delta_h := r(x,a)\,h + \gamma_h\,V(X_{t+h}) - V(x).$$

Taking expectation and using the expansion:

$$\mathbb{E}[\delta_h \mid x,a] = h\Big(r(x,a) + (\mathcal{L}^a V)(x) - \rho V(x)\Big) + o(h).$$

Hence

$$\boxed{\frac{1}{h}\,\delta_h \xrightarrow[h\to 0]{\text{in mean}} r + \mathcal{L}^a V - \rho V} \quad \text{(the HJB residual at } (x,a)\text{).}$$

# $Q$-learning $\approx$ stochastic value iteration for HJB

One-step action-value:

$$Q_h(x,a) := r(x,a)\,h + \gamma_h\,\mathbb{E}\Big[V(X_{t+h}^{x,a})\Big]\,, \qquad V(x) = \sup_a Q_h(x,a).$$

Then

$$\frac{Q_h(x,a) - V(x)}{h} \;\to\; \underbrace{r(x,a) + \mathcal{L}^a V(x) - \rho V(x)}_{\mathcal{H}(x,a;V)}.$$

Off-policy $Q$-learning update:

$$Q_h \leftarrow Q_h + \alpha\Big(rh + \gamma_h \max_{a'} Q_h(x',a') - Q_h(x,a)\Big),$$

i.e. stochastic value iteration for the HJB; the scaled limit $Q_h/h$ estimates the Hamiltonian integrand.

# Policy evaluation in continuous time = Poisson/HJB equation

For a fixed policy $\pi$,

$$r^\pi(x) := \mathbb{E}_{a\sim\pi}[r(x,a)], \quad \mathcal{L}^\pi V := \mathbb{E}_{a\sim\pi}[\mathcal{L}^a V].$$

Evaluation PDE (linear):

$$r^\pi + \mathcal{L}^\pi V^\pi - \rho V^\pi = 0 \iff (\rho I - \mathcal{L}^\pi)V^\pi = r^\pi.$$

TD/Least Square TD with features solves a projected version of this PDE.

# Actor–Critic in continuous time (policy gradients via HJB)

Define the differential advantage:

$$A^\pi(x,a) := r(x,a) + \mathcal{L}^a V^\pi(x) - \rho V^\pi(x).$$

A continuous-time policy gradient theorem (discounted case) yields policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi\big[\nabla_\theta \log \pi_\theta(a|x)\, A^\pi(x,a)\big],$$

with $A^\pi$ estimated by $\delta_h/h$ from the critic, and update $\theta \leftarrow \theta + \eta\, \nabla_\theta J(\theta)$.

# Soft HJB (entropy regularization)

Max-entropy objective adds $\alpha\,\mathsf{H}(\pi(\cdot|x))$:

$$0 = \sup_{\pi} \Big\{ \mathbb{E}_{a\sim\pi}\big[ r(x,a) + \mathcal{L}^a V(x) - \rho V(x) \big] + \alpha\,\mathsf{H}(\pi(\cdot|x)) \Big\}.$$

Optimal policy (Boltzmann in continuous-time $Q$-integrand):

$$\pi^*(a|x) \propto \exp\Big( \tfrac{1}{\alpha}\,[\, r + \mathcal{L}^a V - \rho V \,] \Big).$$

Soft HJB replaces $\max_a$ by log-sum-exp; SAC-style updates solve it sample-wise.

# Practical recipe (how to "do RL" for an HJB)

1. **Time-discretize** with small $h$.
2. **Critic:** regress $V_\phi$ (or $Q_\psi$) to minimize the squared TD error
$$\mathbb{E}\Big[\big(r\,h + \gamma_h V_\phi(x') - V_\phi(x)\big)^2\Big]\,;$$
equivalently, fit $\delta_h/h$ to zero.
3. **Actor:** improve by greedy (deterministic)
$$a^*(x) = \arg\max_a \widehat{\mathcal{H}}(x, a; V_\phi)\,,$$
or by a stochastic policy updated with the gradient using $\widehat{A}_t$ estimated from critic.
4. **Shrink** $h$ (or refine the state interpolant) to reduce discretization error; this converges to the viscosity solution of the HJB.

# Discount mapping and scaling

Discrete $\leftrightarrow$ continuous discount:

$$\gamma_h = e^{-\rho h} \iff \rho = -\frac{1}{h}\log\gamma_h \approx \frac{1-\gamma_h}{h}\ (h \to 0).$$

Reward scaling:

$$r_h(x,a) \approx r(x,a)\,h.$$

**Rule of thumb:** If your TD/Q update uses $(rh,\ \gamma_h)$ at step $h$, then $\delta_h/h$ estimates the HJB residual.