Lecture 3: Bellman Equation and Value Function Iteration

Yasuyuki Sawada, Yaolang Zhong

University of Tokyo

yaolang.zhong@e.u-tokyo.ac.jp

October 15, 2025

Recap: Markov Decision Process (MDP)

- An MDP is defined by the tuple

$$(\mathcal{S},\ \mathcal{A},\ P,\ r,\ \beta)$$

  where:

  - State space $\mathcal{S}$: Possible system states.
  - Action space $\mathcal{A}(s)$: Feasible actions when in state s.
  - Transition kernel $P_t(s' \mid s, a)$: Probability of moving from s to s' given action a.
  - Reward function $r_t(s, a)$: Instantaneous payoff from taking action a in state s. (Sometimes expressed as a *cost* $-r(s, a)$.)
  - Discount factor $\beta \in (0, 1)$: Weights future rewards relative to current ones.

- Objective: Choose a policy $\pi : \mathcal{S} \to \mathcal{A}$ to maximize expected discounted rewards:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{T} \beta^t r_t(s_t, a_t) + \beta^T R_T(s_T) \right].$$

Value Function

- $V_t^\pi(s)$: The value function corresponding to a policy π

$$V_t^\pi(s) \equiv \mathbb{E}_\pi\left[\sum_{\tau=t}^{T-1} \beta^{\tau-t} r_\tau\big(s_\tau, a_\tau)\big) + \beta^{T-t} R_T(s_T) \,\middle|\, s_t = s\right].$$

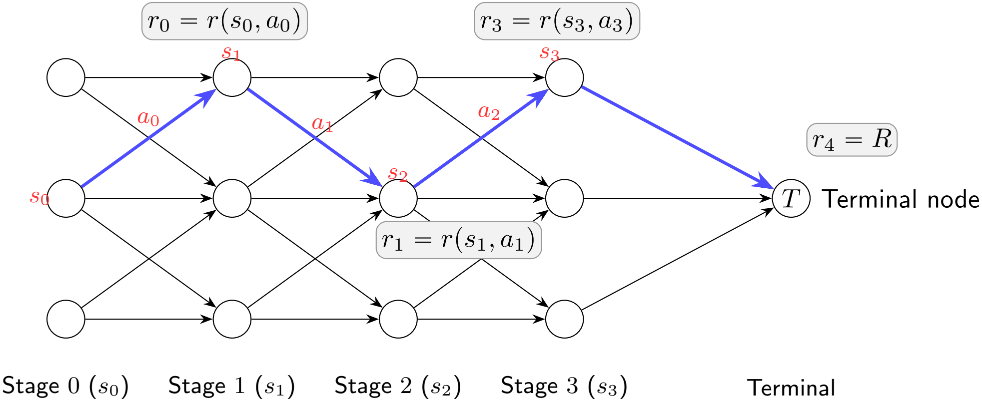
  Note that how the policy $a = \pi(s)$ impact the value value:

  - directly: reward $r_t(s_t, a_t) = r_t(s_t, \pi(s_t))$
  - indirectly: state transition $P_t(s'|s, a) = P_t(s'|s, \pi(s))$

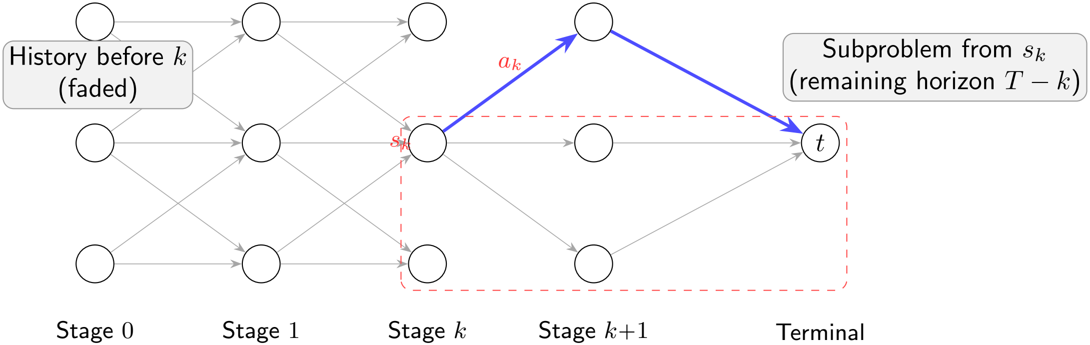
- $V_t^*(s)$: The optimal value function

$$V_t^*(s) = \sup_\pi V_t^\pi(s)$$

Principle of optimality and Backward Induction



Principle of optimality and Backward Induction



Principle of Optimality

- Intuition: the tail of an optimal sequence is optimal for the subproblem that begins where you restart.
- Consider a policy tail $\pi^*_{t:} = (\pi^*_t, \pi^*_{t+1}, \ldots, \pi^*_{T-1})$ that is optimal from state s at time t. Its first decision rule satisfies

$$\pi^*_t(s) \in \arg\max_{a \in \mathcal{A}_t(s)} \mathbb{E}_{\pi^*_{t+1:}}\left[\sum_{\tau=t}^{T-1} \beta^{\tau-t} r_\tau(s_\tau, a_\tau) + \beta^{T-t} R_T(s_T) \,\middle|\, s_t = s,\ a_t = a \right],$$

  where for $\tau > t$ we take $a_\tau = \pi^*_\tau(s_\tau)$.
- Then for any $t' \in \{t, \ldots, T-1\}$ and any state s' reachable at t' under $\pi^*_{t:}$, the continuation rule is optimal for the subproblem:

$$\pi^*_{t'}(s') \in \arg\max_{a \in \mathcal{A}_{t'}(s')} \mathbb{E}_{\pi^*_{t'+1:}}\left[\sum_{\tau=t'}^{T-1} \beta^{\tau-t'} r_\tau(s_\tau, a_\tau) + \beta^{T-t'} R_T(s_T) \,\middle|\, s_{t'} = s',\ a_{t'} = a \right].$$

Bellman optimality (finite horizon, time-varying)

- Setup: periods $t = 0, \dots, T$; discount $\beta \in (0,1)$; time-varying $\mathcal{A}_t(s)$, $r_t(s,a)$, transition $P_t(\cdot \mid s,a)$; terminal payoff R_T.
- Optimal value as a plan problem:

$$V_t(s) \;=\; \max_{\pi_{t:}} \mathbb{E}_{\pi_{t:}}\left[\sum_{\tau=t}^{T-1} \beta^{\tau-t} r_\tau(s_\tau, a_\tau) + \beta^{T-t} R_T(s_T) \,\middle|\, s_t = s \right], \qquad V_T(s) = R_T(s).$$

- By the principle of optimality, one-step decomposition:

$$V_t(s) \;=\; \max_{a \in \mathcal{A}_t(s)} \left\{ r_t(s,a) + \beta\, \mathbb{E}[V_{t+1}(S_{t+1}) \mid s,a] \right\}.$$

- Optimal decision rule at date t:

$$\pi_t^*(s) \in \arg\max_{a \in \mathcal{A}_t(s)} \left\{ r_t(s,a) + \beta\, \mathbb{E}[V_{t+1}(S_{t+1}) \mid s,a] \right\}.$$

Backward induction (finite horizon): pseudo-code

- Step 0 (inputs): horizon T, discount $\beta \in (0,1)$; primitives $\{\mathcal{A}_t(s), r_t(s,a), P_t(\cdot \mid s,a)\}_{t=0}^{T-1}$; terminal payoff R_T.
- Step 1 (initialize terminal value): for all states s, set $V_T(s) \leftarrow R_T(s)$.
- Step 2 (backward sweep): for $t = T-1, T-2, \dots, 0$ and each state s,

$$Q_t(s,a) \;\leftarrow\; r_t(s,a) + \beta\,\mathbb{E}[\,V_{t+1}(S_{t+1}) \mid s,a] \quad \text{for all } a \in \mathcal{A}_t(s),$$

$$V_t(s) \;\leftarrow\; \max_{a\in\mathcal{A}_t(s)} Q_t(s,a), \qquad \pi_t^*(s) \;\in\; \arg\max_{a\in\mathcal{A}_t(s)} Q_t(s,a).$$

- Step 3 (outputs): optimal value sequence $(V_t)_{t=0}^{T}$ and policy sequence $(\pi_t^*)_{t=0}^{T-1}$.
- Remarks
  - dynamic programming (DP) decomposes the multi-stage problem via the principle of optimality
  - the $\arg\max$ at each t is greedy w.r.t. Q_t (globally optimal because V_{t+1} is optimal)

From finite-horizon recursion to the infinite-horizon equation

- Finite-horizon, time-varying recursion ($t = T-1, \dots, 0$):

$$V_T(s) = R_T(s), \qquad V_t(s) = \max_{a \in \mathcal{A}_t(s)} \{\, r_t(s,a) + \beta\, \mathbb{E}[V_{t+1}(S_{t+1}) \mid s, a]\,\}.$$

- Time-homogeneous primitives and infinite horizon:

$$\mathcal{A}_t \equiv \mathcal{A}, \quad r_t \equiv r, \quad P_t \equiv P, \quad \beta \in (0,1), \;\; r \text{ bounded.}$$

- Stationary infinite-horizon objective:

$$V(s) = \sup_{\pi} \mathbb{E}_\pi \Big[\sum_{t=0}^{\infty} \beta^t\, r(S_t, \pi(S_t)) \;\Big|\; S_0 = s \Big].$$

- Bellman equation (time-homogeneous form):

$$V(s) = \max_{a \in \mathcal{A}(s)} \{\, r(s,a) + \beta\, \mathbb{E}[V(S') \mid s, a]\,\}.$$

Bellman operators and fixed points

- Optimal Bellman operator $T : \mathcal{B}(\mathcal{S}) \to \mathcal{B}(\mathcal{S})$:

$$(TV)(s) = \max_{a \in \mathcal{A}(s)} \{ r(s,a) + \beta\, \mathbb{E}[V(S') \mid s,a] \}.$$

- Policy Bellman operator for a fixed stationary policy π:

$$(T^\pi V)(s) = r(s,\pi(s)) + \beta\, \mathbb{E}[V(S') \mid s,\pi(s)].$$

- Fixed points:

$$V^* = TV^* \quad \text{(optimal value)}, \qquad V^\pi = T^\pi V^\pi \quad \text{(value under policy } \pi\text{)}.$$

- Greedy policy with respect to a value V (via the one-step value Q_V):

$$Q_V(s,a) = r(s,a) + \beta\, \mathbb{E}[V(S') \mid s,a], \qquad \pi_V(s) \in \arg\max_a Q_V(s,a).$$

  The choice is greedy with respect to Q_V; when $V = V^*$ this yields an optimal policy $\pi^* = \pi_{V^*}$.

Contraction mapping theorem and its consequences

- Sup norm $\|V\|_\infty = \sup_s |V(s)|$. With bounded rewards and $\beta \in (0,1)$:

$$\|TV - TW\|_\infty \leqslant \beta \|V - W\|_\infty, \qquad \|T^\pi V - T^\pi W\|_\infty \leqslant \beta \|V - W\|_\infty.$$

- Hence T and T^π are β-contractions on $(\mathcal{B}(\mathcal{S}), \|\cdot\|_\infty)$.
- Consequences:
  - Unique fixed points V^* and V^π.
  - Iteration converges geometrically from any start $V^{(0)}$:

$$V^{(k+1)} \leftarrow TV^{(k)} \to V^*, \qquad V^{(k+1)} \leftarrow T^\pi V^{(k)} \to V^\pi.$$

  - Error bound after k value-iteration steps:

$$\|V^{(k)} - V^*\|_\infty \leqslant \frac{\beta^k}{1-\beta} \|TV^{(0)} - V^{(0)}\|_\infty.$$

Value Function Iteration (time-homogeneous, infinite horizon)

- Step 0 (inputs): state space $\mathcal{S}$; actions $\mathcal{A}(s)$; reward $r(s,a)$; transition $P(\cdot \mid s,a)$; discount $\beta \in (0,1)$; tolerance $\varepsilon > 0$.
- Step 1 (Bellman operator): for any $V : \mathcal{S} \to \mathbb{R}$,

$$(TV)(s) \;=\; \max_{a \in \mathcal{A}(s)} \Big\{ r(s,a) + \beta\, \mathbb{E}\big[V(S') \mid s,a\big] \Big\}.$$

- Step 2 (initialize value): choose $V^{(0)}$ (e.g., $V^{(0)} \equiv 0$); set $k \leftarrow 0$.
- Step 3 (value update): for each state s,

$$Q_{V^{(k)}}(s,a) \;=\; r(s,a) + \beta\, \mathbb{E}\Big[V^{(k)}(S') \mid s,a\Big] \quad \text{for all } a \in \mathcal{A}(s),$$

$$V^{(k+1)}(s) \;\leftarrow\; \max_{a \in \mathcal{A}(s)} Q_{V^{(k)}}(s,a).$$

- Step 4 (convergence check): compute $\Delta_{k+1} = \|V^{(k+1)} - V^{(k)}\|_\infty$. If $\Delta_{k+1} \leqslant \varepsilon$, go to Step 5; else set $k \leftarrow k+1$ and return to Step 3.
- Step 5 (policy extraction): greedy w.r.t. $Q_{V^{(k+1)}}$,

$$\pi^*(s) \;\in\; \arg\max_{a \in \mathcal{A}(s)} \Big\{ r(s,a) + \beta\, \mathbb{E}\Big[V^{(k+1)}(S') \mid s,a\Big] \Big\}.$$

Approximation in Dynamic Programming

approximate $\max$
Discretization /
Selective Minimization /
Gradient

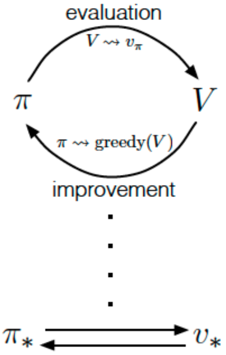
at s_t

$$\max_{a_t \in \mathcal{A}_t(s_t)} \mathbb{E}\Big[r_t(s_t, a_t, W_t) \; + \; \beta \, \tilde{V}_{t+1}(S_{t+1}) \; \Big| \; s_t, a_t \Big]$$

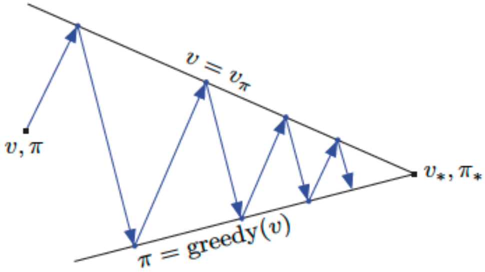
approximate $\mathbb{E}$
Monte Carlo / Quasi-MC / Quadrature / Sparse sampling

approximate $\tilde{V}_{t+1}$
Parametric approximation /
Neural networks /
Problem approximation /
Rollout

Generalized Policy Iteration: two views



(a) View 1



(b) View 2

Figure 1: Generalized Policy Iteration (two views), adapted from Sutton and Barto (2018), Chapter 4.

Decomposing value iteration into evaluation + improvement

Value iteration (one step):

$$V^{(k+1)} \;=\; (TV^{(k)})(s) \;=\; \max_{a\in\mathcal{A}(s)} \left\{ r(s,a) + \beta\,\mathbb{E}[\,V^{(k)}(S') \mid s,a] \right\}.$$

Equivalent two-step decomposition:

1. improvement (greedy w.r.t. $V^{(k)}$):

$$\pi^{(k)}(s) \in \arg\max_{a\in\mathcal{A}(s)} \left\{ r(s,a) + \beta\,\mathbb{E}[\,V^{(k)}(S') \mid s,a] \right\};$$

2. one-step policy evaluation:

$$V^{(k+1)}(s) \;=\; (T^{\pi^{(k)}} V^{(k)})(s) \;=\; r\big(s, \pi^{(k)}(s)\big) + \beta\,\mathbb{E}\Big[V^{(k)}(S') \mid s, \pi^{(k)}(s)\Big].$$

Notes: Full evaluation to $V^{\pi^{(k)}}$ yields classical policy iteration; a single backup yields value iteration; in-between gives modified policy iteration. All are instances of GPI.

Generalized Policy Iteration (GPI): idea

- Maintain a current policy π and a value approximation V (or Q).
- Two interacting processes:
  - policy evaluation: push V toward V^π using T^π (exact, iterative, or sample-based);
  - policy improvement: make π greedier w.r.t. the current V or Q.
- These can be interleaved, partial, and asynchronous; the combination drives (V, π) toward (V^*, π^*).
- With finite MDPs and $\beta \in (0, 1)$, repeated improvement and sufficient evaluation converge to an optimal policy; value iteration and policy iteration are special cases.

Why the decomposition works (sketch)

- T and T^π are β-contractions in $\|\cdot\|_\infty$ (bounded rewards, $\beta \in (0,1)$) $\Rightarrow$ unique fixed points V^* and V^π, geometric convergence.
- Greedy improvement yields π' with $V^{\pi'} \geqslant V^\pi$ componentwise.
- Alternating evaluation and improvement produces a monotone sequence bounded by V^*; sufficient evaluation/exploration $\Rightarrow$ optimality.

References & Further Reading

- Dimitri P. Bertsekas, **Reinforcement learning and optimal control (2025 Spring course at ASU)**, Lecture 1 and 2: https://web.mit.edu/dimitrib/www/RLbook.html
- Sutton and Barto, **Reinforcement Learning: An Introduction**, Chapter 4 https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

Sutton, Richard S. and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.