@zblu Reinforcement Learning Theory

INF8250AE Course Notes

RL 4:Planning with a Known Model

Policy Evaluation · Value Iteration · Policy Iteration · Linear Programming Formulation

Key Topics: Policy Evaluation, Value Iteration, Policy Iteration, Policy Improvement Theorem, Convergence Analysis, Linear Programming for Optimal Value Function
Reference: `https://amfarahmand.github.io/IntroRL/`

Notes on Reinforcement Learning – Oct 15, 2025

Reinforcement Learning 3: Bellman Operators: Properties and Consequences

Zhuobie

2025 年 10 月 15 日

1. Problem Background

In this chapter we study *planning with a known MDP* $(\mathcal{X}, \mathcal{A}, P, R, \gamma)$: our objective is to compute the optimal value function (and thus an optimal policy) using exact model information. The focus is on three fundamental methods that we will keep referring to in later RL analysis: *Value Iteration (VI)*, *Policy Iteration (PI)*, and an *LP formulation*.

1.1 Goal and Setting

- **Goal:** find π^* (or V^*, Q^*) for a given, known MDP.
- **Assumption:** the model is known (we know P and R). This assumption is stronger than the usual RL setting, but the algorithms and proofs here form the basis for understanding and designing RL methods later.
- **Two standard viewpoints:**
  1. *Value-based:* first compute V^* or Q^*, then act greedily to obtain π^*.
  2. *Policy-based (direct search):* improve policies without explicitly computing V^*, possibly guided by value information (hybrid).

1.2 Policy Evaluation vs. Control

We distinguish:

- **Policy Evaluation (PE):** for a given π, compute V^π or Q^π.
- **Control:** compute V^*, Q^* or directly obtain π^*.

Dynamic Programming (DP) methods exploit Bellman structure to solve both tasks efficiently.

2. Policy Evaluation

2.1 Problem Statement

Given an MDP $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ and a policy π, we would like to compute V^π or Q^π:

$$V^\pi(x) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \,\middle|\, X_1 = x\right].$$

2.2 Direct Computation

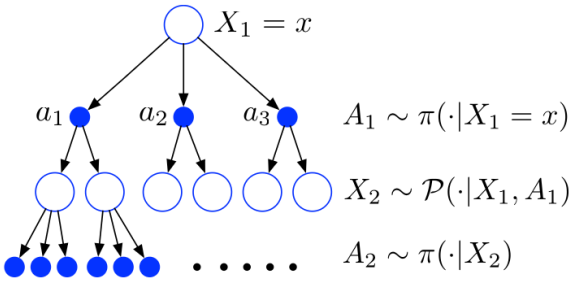

Figure 1: Policy Evaluation: compute V^π for a given policy π.

This is inefficient. The size of the tree grows very fast. The sample-based approximate version of this idea works. The structure is really like the one in multistage stochastic programming.

2.3 Linear-System View of Policy Evaluation

There is a way using the bellman operator to do the policy evaluation more efficiently. Using the Bellman equation for any fixed π,

$$V^\pi = T^\pi V^\pi, \qquad (T^\pi V)(x) = r^\pi(x) + \gamma \sum_{x'} P^\pi(x'|x)\, V(x'),$$

the PE problem on a finite MDP reduces to a linear system

$$(I - \gamma P^\pi)\, V = r^\pi.$$

This can be solved by standard linear solvers (forming the inverse explicitly is unnecessary). In contrast, the *control* Bellman optimality equation

$$V = T^* V, \qquad (T^* V)(x) = \max_a \left\{ r(x,a) + \gamma \sum_{x'} P(x'|x,a) V(x') \right\}$$

is nonlinear due to the max operator, so a different approach is needed.

2.4 Numerical Example: Policy Evaluation (Two States)

We consider a fixed policy π on a tiny MDP with two states $\{1, 2\}$:

$$r^\pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad P^\pi = \begin{bmatrix} 0.5 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}, \qquad \gamma = 0.9.$$

We want to compute $V^\pi = [V_1, V_2]^\top$.

(A) Direct computation (infinite series). Using the Neumann expansion

$$V^\pi = \sum_{k=0}^{\infty} \gamma^k (P^\pi)^k r^\pi,$$

the first few partial sums are

$$S_0 = r^\pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$S_1 = r^\pi + \gamma P^\pi r^\pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0.9 \begin{bmatrix} 0.5 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 1.45 \\ 0.18 \end{bmatrix},$$

$$S_2 = S_1 + \gamma^2 (P^\pi)^2 r^\pi = \begin{bmatrix} 1.7335 \\ 0.3906 \end{bmatrix},$$

$$S_3 = S_2 + \gamma^3 (P^\pi)^3 r^\pi \approx \begin{bmatrix} 1.956 \\ 0.593 \end{bmatrix},$$

$$S_4 = S_3 + \gamma^4 (P^\pi)^4 r^\pi \approx \begin{bmatrix} 2.147 \\ 0.779 \end{bmatrix}.$$

As k
to
$infty$, the sum converges to

$$V^\pi \approx \begin{bmatrix} 3.8356 \\ 2.4658 \end{bmatrix}.$$

(B) Linear-system approach. Solve the Bellman equation

$$V^\pi = r^\pi + \gamma P^\pi V^\pi \quad \Leftrightarrow \quad (I - \gamma P^\pi) V^\pi = r^\pi.$$

Compute

$$I - \gamma P^\pi = \begin{bmatrix} 0.55 & -0.45 \\ -0.18 & 0.28 \end{bmatrix}, \qquad r^\pi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Hence

$$\begin{cases} 0.55 V_1 - 0.45 V_2 = 1, \\ \\ -0.18 V_1 + 0.28 V_2 = 0, \end{cases} \quad \Rightarrow \quad V_2 = \tfrac{0.18}{0.28} V_1 = \tfrac{9}{14} V_1.$$

Substitute into the first equation:

$$(0.55 - 0.45 \times \tfrac{9}{14}) V_1 = 1 \;\Rightarrow\; V_1 \approx 3.8356, \qquad V_2 = \tfrac{9}{14} V_1 \approx 2.4658.$$

Check. Both approaches give the same result: $V^\pi(1) \approx 3.8356$, $V^\pi(2) \approx 2.4658$. The series expansion and the linear-system formulation are equivalent since

$$(I - \gamma P^\pi)^{-1} = \sum_{k=0}^{\infty} (\gamma P^\pi)^k.$$

In large-scale problems, we often use iterative solvers or approximate evaluation instead of computing the matrix inverse explicitly.

Remark

Observation. When solving for V^π, the Bellman equation

$$V^\pi = r^\pi + \gamma P^\pi V^\pi$$

is a *linear* system because P^π is fixed under a given policy. However, for the optimal control problem,

$$V = T^* V, \qquad (T^*V)(x) = \max_{a \in \mathcal{A}} \Big\{ r(x,a) + \gamma \sum_{x'} P(x'|x,a) V(x') \Big\},$$

the presence of the $\max_a$ operator couples the equations in a nonlinear way. Hence, finding V^* is *not* a linear system anymore; we must use iterative methods such as Value Iteration or Policy Iteration.

3. Value Iteration

Value Iteration (VI) can be viewed as an approximate form of policy evaluation, where instead of solving the linear system $(I - \gamma P^\pi)V = r^\pi$ exactly, we iteratively apply the *Bellman optimality operator*

$$(T^*V)(x) = \max_{a \in \mathcal{A}} \Big\{ r(x,a) + \gamma \sum_{x'} P(x'|x,a) V(x') \Big\}.$$

3.1 Iterative Scheme

Starting from any V_0, we update

$$V_{k+1} = T^* V_k = \max_a \Big\{ r(x,a) + \gamma \sum_{x'} P(x'|x,a) V_k(x') \Big\}.$$

Each step applies a one-sweep improvement similar to policy evaluation, but the $\max_a$ operation automatically performs a greedy improvement at every state.

3.2 Convergence

Theorem 3.1 *The operator T^* is a γ-contraction in $\|\cdot\|_\infty$. Hence $V_k \to V^*$ for any bounded V_0.*

Interpretation. Policy evaluation solves $(I - \gamma P^\pi)V = r^\pi$ exactly for fixed π. Value Iteration replaces the exact solve with successive applications of T^*, thereby approximating the solution to the nonlinear optimality equation $V = T^*V$.

4. Policy Iteration

Policy Iteration (PI) alternates between two main steps:

1. **Policy Evaluation:** for the current policy π_k, compute its value function

$$V^{\pi_k} = T^{\pi_k} V^{\pi_k}, \qquad (T^{\pi_k}V)(x) = r^{\pi_k}(x) + \gamma \sum_{x'} P^{\pi_k}(x'|x) V(x').$$

2. **Policy Improvement:** build a new greedy policy We define the policy improvement step as follows:

$$\pi_{k+1}(x) \leftarrow \pi_g(x; Q^{\pi_k}) = \arg\max_{a \in \mathcal{A}} Q^{\pi_k}(x, a), \qquad \forall x \in \mathcal{X}.$$

This is equivalent to enforcing

$$T^{\pi_{k+1}} Q^{\pi_k} = T^* Q^{\pi_k}.$$

Intuitively, Step 1 evaluates the current policy; Step 2 updates it greedily w.r.t. the estimated values.

Proof

By definition of the Bellman operator,

$$(T^{\pi_{k+1}} Q^{\pi_k})(x, a) = r(x, a) + \gamma \int_{\mathcal{X}} P(dx'|x, a) \int_{\mathcal{A}} \pi_{k+1}(da'|x') Q^{\pi_k}(x', a').$$

Since π_{k+1} is greedy, it places all probability mass on $a' = a^*(x') = \arg\max_{a'} Q^{\pi_k}(x', a')$. Hence

$$\pi_{k+1}(da'|x') = \delta_{a^*(x')}(da'),$$

and for any function $f(a')$,

$$\int_{\mathcal{A}} f(a')\, \delta_{a^*(x')}(da') = f(a^*(x')).$$

Applying this to the inner integral gives

$$\int_{\mathcal{A}} \pi_{k+1}(da'|x')\, Q^{\pi_k}(x', a') = Q^{\pi_k}(x', a^*(x')) = \max_{a'} Q^{\pi_k}(x', a').$$

Therefore,

$$\begin{aligned}(T^{\pi_{k+1}} Q^{\pi_k})(x, a) &= r(x, a) + \gamma \int_{\mathcal{X}} P(dx'|x, a) \max_{a'} Q^{\pi_k}(x', a') \\ &= (T^* Q^{\pi_k})(x, a).\end{aligned}$$

4.1 Policy Improvement Theorem

Theorem 4.1 (Policy Improvement) *If for policies π and π' it holds that*

$$T^{\pi'} Q^{\pi} = T^* Q^{\pi},$$

then

$$Q^{\pi'} \geq Q^{\pi} \quad \textit{(pointwise)}.$$

In other words, the greedy policy is a proper policy improvement step.

Proof

Step 1: Show that $T^{\pi'} Q^{\pi} \geq Q^{\pi}$.
By the assumption $T^{\pi'} Q^{\pi} = T^* Q^{\pi}$ and the fact that $T^* Q^{\pi} \geq T^{\pi} Q^{\pi} = Q^{\pi}$, we have

$$T^{\pi'} Q^{\pi} = T^* Q^{\pi} \geq Q^{\pi}.$$

Equivalently, for any (x, a),

$$\underbrace{r(x,a) + \gamma \int P(dx'|x,a) \max_{a' \in \mathcal{A}} Q^{\pi}(x', a')}_{(T^* Q^{\pi})(x,a)} \;\geq\; \underbrace{r(x,a) + \gamma \int P(dx'|x,a) Q^{\pi}\big(x', \pi(x')\big)}_{(T^{\pi} Q^{\pi})(x,a)}. \tag{1}$$

Step 2: Use monotonicity of $T^{\pi'}$ and iterate.
Apply $T^{\pi'}$ to both sides of $T^{\pi'} Q^{\pi} \geq Q^{\pi}$ and use that $T^{\pi'}$ is monotone:

$$T^{\pi'}(T^{\pi'} Q^{\pi}) \;\geq\; T^{\pi'} Q^{\pi} \;=\; T^* Q^{\pi} \;\geq\; Q^{\pi}.$$

Thus $(T^{\pi'})^2 Q^{\pi} \geq Q^{\pi}$. Repeating this argument, for any $m \geq 1$,

$$(T^{\pi'})^m Q^{\pi} \;\geq\; Q^{\pi}. \tag{2}$$

Step 3: Take limits and use contraction of $T^{\pi'}$.
Since $T^{\pi'}$ is a γ-contraction, $(T^{\pi'})^m Q^{\pi}$ converges to the unique fixed point of $T^{\pi'}$, namely $Q^{\pi'}$:

$$\lim_{m \to \infty} (T^{\pi'})^m Q^{\pi} \;=\; Q^{\pi'}. \tag{3}$$

we get that

$$Q^{\pi'} = \lim_{m \to \infty} (T^{\pi'})^m Q^{\pi} \geq T^* Q^{\pi} \geq Q^{\pi}. \tag{3}$$

4.2 Convergence of Policy Iteration

Theorem 4.2 (Convergence of the Policy Iteration Algorithm) *Let $(\pi_k)_{k \geq 0}$ be the sequence generated by Policy Iteration. Then:*

- *For all k, we have $V^{\pi_{k+1}} \geq V^{\pi_k}$, with equality if and only if $V^{\pi_k} = V^*$.*
- $\lim_{k \to \infty} \| V^{\pi_k} - V^* \|_{\infty} = 0.$

Proof

Proof of the first point $V^{\pi_{k+1}} \geq V^{\pi_k}$ and equality condition. By the policy improvement step, π_{k+1} is greedy w.r.t. Q^{π_k}, hence

$$T^{\pi_{k+1}} Q^{\pi_k} = T^* Q^{\pi_k} \;\geq\; T^{\pi_k} Q^{\pi_k} = Q^{\pi_k}.$$

[Equivalence: $V^{\pi_{k+1}} = V^{\pi_k} \iff V^{\pi_k} = V^*$]

($\Rightarrow$) If $V^{\pi_{k+1}} = V^{\pi_k}$, then $V^{\pi_k} = V^*$. Equality of values gives

$$T^{\pi_k} V^{\pi_k} = V^{\pi_k} \quad \text{and} \quad T^{\pi_{k+1}} V^{\pi_k} = V^{\pi_{k+1}} = V^{\pi_k}.$$

But π_{k+1} is greedy w.r.t. Q^{π_k}, so

$$T^{\pi_{k+1}} V^{\pi_k} = T^* V^{\pi_k}.$$

Therefore V^{π_k} is a fixed point of T^*. The fixed point of T^* is unique and equals V^*, hence $V^{\pi_k} = V^*$.

($\Leftarrow$) If $V^{\pi_k} = V^*$, then $V^{\pi_{k+1}} = V^{\pi_k}$. From $V^{\pi_k} = V^*$ we have $Q^{\pi_k} = Q^*$. Any greedy policy w.r.t. Q^* is

optimal; in particular π_{k+1} (greedy w.r.t. $Q^{\pi_k} = Q^*$) is optimal, so

$$V^{\pi_{k+1}} = V^* = V^{\pi_k}.$$

Combining the two directions,

$$\boxed{V^{\pi_{k+1}} = V^{\pi_k} \iff V^{\pi_k} = V^*.}$$

Proof

Proof of the second point $\lim_{k\to\infty} \|V^{\pi_k} - V^*\|_\infty = 0$**.** From $T^{\pi_{k+1}} Q^{\pi_k} = T^* Q^{\pi_k} \geq Q^{\pi_k}$ we obtain

$$Q^{\pi_{k+1}} \geq T^* Q^{\pi_k} \geq Q^{\pi_k}.$$

By induction,

$$Q^{\pi_{k+1}} \geq (T^*)^k Q^{\pi_0}.$$

For any policy π, $Q^\pi \leq Q^*$, hence

$$Q^* \geq Q^{\pi_{k+1}} \geq (T^*)^k Q^{\pi_0}.$$

Therefore,

$$\| Q^{\pi_{k+1}} - Q^* \|_\infty \leq \| (T^*)^k Q^{\pi_0} - Q^* \|_\infty \xrightarrow[k\to\infty]{} 0$$

by the γ-contraction of T^*. Thus $Q^{\pi_k} \to Q^*$ and consequently $V^{\pi_k} \to V^*$ in $\| \cdot \|_\infty$.

4.3 Complexity

4.4 Complexity of Policy Iteration

It can be shown that the PI algorithm converges in

$$\mathcal{O}\left(\frac{|\mathcal{X}|\,|\mathcal{A}|}{1-\gamma} \log \frac{1}{1-\gamma} \right)$$

iterations. The proof appears in *Foundations of Reinforcement Learning* (Farahmand, 2025). This is a significant quantitative improvement over earlier bounds.

Remark

This is a relatively recent result, with related forms proved by Ye (2011), Hansen–Miltersen–Zwick (2013), and Scherrer (2016).

5. Linear Programming for Finding V^*

Consider the upper-bound set

$$\mathcal{C} = \{ V : V \geq T^* V \}.$$

Interesting property. If $V \in \mathcal{C}$, then

$$V \geq T^*V \;\Rightarrow\; T^*V \geq T^*(T^*V) = (T^*)^2V \;\Rightarrow\; \cdots \;\Rightarrow\; V \geq (T^*)^m V, \quad \forall\, m \geq 1.$$

Hence every $V \in \mathcal{C}$ dominates V^*, so V^* is the *pointwise minimal* element of $\mathcal{C}$.

5.1 LP Formulation

Choose a strictly positive weight vector $\mu > 0$ (dimension $|\mathcal{X}|$) and solve

$$\min_{V \in \mathcal{C}} \mu^\top V \qquad \Longleftrightarrow \qquad \min_{V} \mu^\top V \;\text{ s.t. }\; V(x) \;\geq\; (T^*V)(x), \;\; \forall x \in \mathcal{X}.$$

Each nonlinear constraint

$$V(x) \;\geq\; \max_{a \in \mathcal{A}} \Big\{ r(x,a) + \gamma \sum_y P(y|x,a)\, V(y) \Big\}$$

is equivalent to the family of linear constraints

$$V(x) \;\geq\; r(x,a) + \gamma \sum_y P(y|x,a)\, V(y), \qquad \forall (x,a) \in \mathcal{X} \times \mathcal{A}.$$

Therefore, the problem becomes the linear program

$$\min_V \mu^\top V \quad \text{s.t.} \quad V(x) \;\geq\; r(x,a) + \gamma \sum_y P(y|x,a)\, V(y), \;\; \forall (x,a) \in \mathcal{X} \times \mathcal{A},$$

which has $|\mathcal{X}| \cdot |\mathcal{A}|$ linear constraints and yields V^*.

Remark

No matter whether we approach from above or below, the sequence converges to the unique fixed point V^* of the Bellman optimality operator T^* ; hence, in principle, the problem can also be formulated as a maximization linear program.

$$\mathcal{C} \;=\; \{\, V \;:\; V \leq T^*V \,\}.$$

$$\max_V \mu^\top V \quad \text{s.t.} \quad V(x) \;\leq\; \max_{a \in \mathcal{A}} \Big\{\, r(x,a) + \gamma \sum_y P(y \mid x,a)\, V(y) \,\Big\}, \qquad \forall x \in \mathcal{X}.$$

The formulation with "$\leq \max(\cdot)$" is not used in practice, since it is nonlinear and cannot be expanded into linear constraints. In contrast, writing $V(x) \geq r(x,a) + \gamma \sum_y P(y|x,a)V(y)$ for all a is linear and naturally fits the LP form.