Algorithmic and Theoretical Foundations of RL

Value Iteration and Policy Iteration

Recap: Bellman Operator and Bellman Optimality Operator

Bellman Operator

Elementwise form: $[\mathcal{T}_\pi v](s) = \underbrace{\mathbb{E}_{a\sim\pi(\cdot|s)}\mathbb{E}_{s'}}_{\mathbb{E}_\pi}\left[r(s,a,s') + \gamma v(s')\right]$

Matrix form: $\mathcal{T}_\pi v = r_\pi + \gamma P^\pi v$

$\mathcal{T}_\pi$ is a contraction and v_π a fixed point of $\mathcal{T}_\pi$: $\mathcal{T}_\pi v_\pi = v_\pi$.

Bellman Optimality Operator

Elementwise form: $[\mathcal{T} v](s) = \max_a \mathbb{E}_{s'}\left[r(s,a,s') + \gamma v(s')\right]$

Matrix form: $\mathcal{T} v = \max_\pi \mathcal{T}_\pi v = \max_\pi \{r_\pi + \gamma P^\pi v\}$

$\mathcal{T}$ is a contraction and v^* a fixed point of $\mathcal{T}$: $\mathcal{T} v^* = v^*$.

Table of Contents

Value Iteration

Value Iteration (VI): Solve Bellman optimality equation by fixed point iteration,

$$v_{k+1}(s) \leftarrow \max_{a} \sum_{s' \in \mathcal{S}} P\left(s'|s,a\right)\left(r(s,a,s') + \gamma v_k\left(s'\right)\right),$$

- To retrieve a policy after value iteration:

$$\pi_k(a|s) = \begin{cases} 1 & \arg\max_{a} \sum_{s' \in \mathcal{S}} P\left(s'|s,a\right)\left(r(s,a,s') + \gamma v_k\left(s'\right)\right) \\ 0 & \text{otherwise.} \end{cases}$$

Convergence of Value Iteration

Theorem 1

Let $\{v_k\}$ be the sequence of value functions produced by value iteration. Then for any $k \geq 0$,

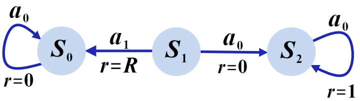
$$\|v_k - v^*\|_\infty \leq \gamma^k \|v_0 - v^*\|_\infty ,$$

which implies that $\lim_{k\to\infty} v_k = v^*$.

- The per iteration computational cost of value iteration is $O\left(|\mathcal{S}|^2|\mathcal{A}|\right)$.
- After at most $k = O\left(\frac{\log(1/\varepsilon)}{\log(1/\gamma)}\right)$ iterations, one has $\|v_k - v^*\|_\infty \leq \varepsilon$.

We may also write $k = O\left(\frac{1}{1-\gamma}\log(1/\varepsilon)\right)$, where $\frac{1}{1-\gamma}$ is referred to as the planning horizon that can relate a infinite horizon discounted problem to a finite horizon problem.

Illustrative Example



- three states: $\mathcal{S} = \{s_0, s_1, s_2\}$
- two actions: $\mathcal{A} = \{a_0, a_1\}$

Each edge is associated with a deterministic transition and a reward.

Suppose we start from $v_0 = 0$. Then

$$v_k(s_0) = r(s_0, a_0, s_0) + \gamma v_{k-1}(s_0) = \gamma v_{k-1}(s_0) = \gamma^k v_0(s_0) = 0,$$

$$v_k(s_2) = r(s_2, a_0, s_2) + \gamma v_{k-1}(s_2) = 1 + \gamma v_{k-1}(s_2) = \frac{1-\gamma^k}{1-\gamma} + \gamma^k v_0(s_2) = \frac{1-\gamma^k}{1-\gamma},$$

$$\begin{aligned} v_k(s_1) &= \max\left\{r(s_1, a_0, s_2) + \gamma v_{k-1}(s_2), r(s_1, a_1, s_0) + \gamma v_{k-1}(s_0)\right\} \\ &= \max\left\{\frac{\gamma}{1-\gamma}\left(1-\gamma^{k-1}\right), R\right\}. \end{aligned}$$

Thus (assuming $R < \frac{\gamma}{1-\gamma}$),

$$v^*(s_0) = \lim_{k\to\infty} v_k(s_0) = 0, v^*(s_1) = \lim_{k\to\infty} v_k(s_1) = \frac{\gamma}{1-\gamma}, v^*(s_2) = \lim_{k\to\infty} v_k(s_2) = \frac{1}{1-\gamma}.$$

Asynchronous Value Iteration

The state values in VI are updated synchronously. An alternative is **asynchronous value iteration**: Rather than sweeping through all states to create a new value vector, only updates one state (an entry of vector) at a time.

Gauss-Seidel Value Iteration:

$$\text{for } s = 1, 2, 3, \ldots :$$
$$v(s) \leftarrow \max_a \sum_{s'} p(s'|s, a)\left(r(s, a, s') + \gamma v(s')\right)$$

Table of Contents

Policy Iteration

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi^*$$

Policy Iteration (PI) has two ingredients: Given π_0,

- Policy Evaluation:

$$v_{\pi_k} = r_{\pi_k} + \gamma P^{\pi_k} v_{\pi_k},$$

- Policy Improvement (one-step value iteration):

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = \arg\max\limits_a \left\{ \underbrace{\sum_{s'} p(s'|s,a)\left(r(s,a,s') + \gamma v_{\pi_k}(s')\right)}_{q_{\pi_k}(s,a)} \right\} \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathcal{T}_{\pi_{k+1}} v_{\pi_k} = \mathcal{T} v_{\pi_k} = r_{\pi_{k+1}} + \gamma P^{\pi_{k+1}} v_{\pi_k}$.

Policy improvement in PI is one-step lookahead plus exploitation of the experience from π_k.

Theorem 2 (Policy Improvement)

For any policy π, if π' is a deterministic policy such that for every state s,

$$q_\pi\left(s, \pi'(s)\right) \geq v_\pi(s),$$

then we have $\pi' \geq \pi$.

Corollary 1

For any given initial policy π_0, policy iteration generates an improving sequence of policies $\{\pi_k\}$, i.e.,

$$v_{\pi_{k+1}}(s) \geq v_{\pi_k}(s), \forall s \in \mathcal{S}.$$

Proof. It is clear that

$$q_{\pi_k}(s, \pi_{k+1}(s)) = \mathcal{T}_{\pi_{k+1}} v_{\pi_k}(s) = \mathcal{T} v_{\pi_k}(s) \geq \mathcal{T}_{\pi_k} v_{\pi_k}(s) = v_{\pi_k}(s).$$

Here $\pi'(s)$ denotes the action π' chooses.

Convergence of Policy Iteration

Theorem 3

Let $\{\pi_k\}$ be the policy sequence produced by policy iteration. Then for any $k \geq 0$,

$$\|v_{\pi_k} - v^*\|_\infty \leq \gamma^k \|v_{\pi_0} - v^*\|_\infty$$

which implies that $\lim_{k\to\infty} v_{\pi_k} = v^*$.

- The per iteration computational cost of policy iteration is $O\left(|\mathcal{S}|^3\right)$ to evaluate v_{π_k} plus $O\left(|\mathcal{S}|^2|\mathcal{A}|\right)$ to produce a new policy.
- After at most $k = O\left(\frac{\log(1/\varepsilon)}{\log(1/\gamma)}\right)$ iterations, one has $\|v_{\pi_k} - v^*\|_\infty \leq \varepsilon$.

Proof of Theorem 3

First it holds that

$$\begin{aligned} v_{\pi_k} &= r_\pi + \gamma P^{\pi_k} v_{\pi_k} \\ &\geq r_\pi + \gamma P^{\pi_k} v_{\pi_{k-1}} \\ &= \mathcal{T} v_{\pi_{k-1}} \geq \cdots \geq \mathcal{T}^k v_{\pi_0}. \end{aligned}$$

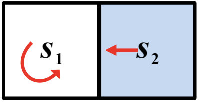
It follows that

$$v^* - v_{\pi_k} \leq v^* - \mathcal{T}^k v_{\pi_0} = \mathcal{T}^k (v^* - v_{\pi_0}).$$

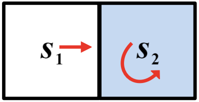
The assertion follows immediately by taking infinite norm on both sides.

Illustrative Example

Consider the example in following figure, where each state is associated with three possible actions: a_l, a_0, a_r (move leftwards, stay unchanged, and move rightwards). The reward is r_{s_1} = -1 and r_{s_2} = 1. The discount rate is γ = 0.9.



(a)



(b)

Assume the initial policy π_0 is given in (a). This policy satisfies $\pi_0(a_l|s_1) = 1$ and $\pi_0(a_l|s_2) = 1$. This policy is not good because it does not move toward s_2. We next apply policy iteration algorithm to this setting.

“Mathematical Foundation of Reinforcement Learning” by Shiyu Zhao, 2022.

Illustrative Example

- Policy Evaluation:

$$\begin{cases} v_{\pi_0}(s_1) = -1 + \gamma v_{\pi_0}(s_1) \\ v_{\pi_0}(s_2) = -1 + \gamma v_{\pi_0}(s_1) \end{cases} \Rightarrow \begin{cases} v_{\pi_0}(s_1) = -10 \\ v_{\pi_0}(s_2) = -10 \end{cases}$$

- Policy Improvement:

$q_{\pi_0}(s,a)$	a_ℓ	a_0	a_r
s_1	$-$	-10	-8
s_2	-10	-8	$-$

Since π_1 choose the action that maximize $q_{\pi_0}(s,a)$, one has (see (b)):

$$\pi_1(a_r|s_1) = 1, \pi_1(a_0|s_2) = 1$$

It is evident that this is an optimal policy.

Table of Contents

δ-Optimal Policy and Error Amplification

Definition 1 (δ-optimal policy)
A policy π is called δ-optimal policy if

$$v_\pi \geq v^* - \delta \mathbf{1}.$$

Theorem 4 (Error-Amplification)

For any vector $v \in \mathbb{R}^{|\mathcal{S}|}$, let π_v be the greedy policy with respect to v, i.e,

$$\pi_v(a|s) = \begin{cases} 1 & a = \arg\max_a \sum_{s'} p(s'|s,a)\left(r(s,a,s') + \gamma v(s')\right) \\ 0 & \text{otherwise.} \end{cases}$$

Then $v_{\pi_v} \geq v^* - \frac{2\gamma}{1-\gamma} \|v - v^*\|_\infty \mathbf{1}$.

Proof of Theorem 4

A Useful Lemma

Lemma 1

For any policy π and a vector $v \in \mathbb{R}^{|\mathcal{S}|}$, there holds

$$v_\pi \geq v - \frac{1}{1-\gamma} \max_s \{v(s) - T_\pi v(s)\} \mathbf{1}.$$

Proof. Note that $v_\pi = \mathcal{T}_\pi^k v, k \to \infty$. We may first consider $\mathcal{T}_\pi^k v$,

$$\begin{aligned}
\mathcal{T}_\pi^k v = \mathcal{T}_\pi^{k-1}(\mathcal{T}_\pi v) &\geq \mathcal{T}_\pi^{k-1}\left(v - \max_s \{v(s) - \mathcal{T}_\pi v(s)\}\mathbf{1}\right) \\
&= \mathcal{T}_\pi^{k-1} v - \gamma^{k-1} \max_s \{v(s) - \mathcal{T}_\pi v(s)\}\mathbf{1} \\
&\geq \cdots\cdots \\
&\geq v - (1 + \cdots + \gamma^{k-1}) \max_s \{v(s) - \mathcal{T}_\pi v(s)\}\mathbf{1} \\
&= v - \frac{1-\gamma^k}{1-\gamma} \max_s \{v(s) - \mathcal{T}_\pi v(s)\}\mathbf{1}.
\end{aligned}$$

Taking a limit on both sides yield the result.

Proof of Theorem 4

Proof

For ease of notation, we simplify π_v to π. One has

$$\mathcal{T}_\pi v - \mathcal{T}_\pi^2 v = r_\pi + \gamma P^\pi v - r_\pi - \gamma P^\pi (\mathcal{T}_\pi v) = \gamma P^\pi (v - \mathcal{T}_\pi v).$$

Thus, it follows that

$$\begin{aligned}\max_s\{\mathcal{T}_\pi v(s) - \mathcal{T}_\pi^2 v(s)\} &\leq \gamma \max_s\{P^\pi (v - \mathcal{T}_\pi v)(s)\} \leq \gamma \max_s\{(v - \mathcal{T}_\pi v)(s)\} \\ &= \gamma \max_s\{(v - \mathcal{T} v)(s)\} \leq \gamma(1+\gamma)\|v - v^*\|_\infty,\end{aligned}$$

where the inequality follows from the fact $\mathcal{T}_\pi v = \mathcal{T} v$ by the definition of π. Thus, the application of Lemma 1 yields that

$$\begin{aligned}v_\pi &\geq \mathcal{T}_\pi v - \frac{1}{1-\gamma}\max_s\{\mathcal{T}_\pi v(s) - \mathcal{T}_\pi^2 v(s)\}\mathbf{1} \\ &\geq \mathcal{T} v - \frac{\gamma(1+\gamma)}{1-\gamma}\|v - v^*\|_\infty = \mathcal{T} v - \mathcal{T} v^* + v^* - \frac{1}{1-\gamma}\max_s\{\mathcal{T}_\pi v(s) - \mathcal{T}_\pi^2 v(s)\}\mathbf{1},\end{aligned}$$

from which the assertion follows directly.

δ-Optimal Policy and Error Amplification

Theorem 5 (Q-Error-Amplification)
For any vector $q \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$, let π_q be the greedy policy with respect to q, i.e.,

$$\pi_q(a|s) = \begin{cases} 1 & a = \arg\max_{a \in \mathcal{A}} q(s,a) \\ 0 & \textit{otherwise}. \end{cases}$$

Then $v^{\pi_q} \geq v^* - \frac{2}{1-\gamma} \|q - q^*\|_\infty \mathbf{1}$.

Proof. The theorem can be proved in a pretty straightforward way.

Computational Complexity for δ-Optimal Policy

Theorem 6 (Computational Complexity of Value Iteration)
Fix a target accuracy δ. Then after

$$O\left(\frac{|\mathcal{S}|^2 |\mathcal{A}|}{1-\gamma} \log\left(\frac{1}{(1-\gamma)\delta}\right)\right)$$

elementary arithmetic operations, value iteration produces a δ-optimal π.

Theorem 7 (Computational Complexity of Policy Iteration)
Fix a target accuracy δ. Then after

$$O\left(\frac{|\mathcal{S}|^3 + |\mathcal{S}|^2 |\mathcal{A}|}{1-\gamma} \log\left(\frac{1}{\delta}\right)\right)$$

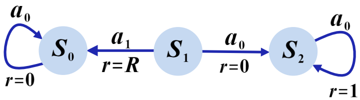
elementary arithmetic operations, policy iteration produces a δ-optimal π.

Computational Complexity for Optimal Policy

Definition 2 (Strongly Polynomial)
An algorithm is strongly polynomial if it is guaranteed to find an optimal policy with computation complexity ***only*** *being polynomial in* $|\mathcal{S}|$, $|\mathcal{A}|$, *and the planning horizon* $\frac{1}{1-\gamma}$.

- VI is not strongly polynomial, but PI is strongly polynomial.

VI is Not Strongly Polynomial: Example



- three states: $\mathcal{S} = \{s_1, s_1, s_3\}$
- two actions: $\mathcal{A} = \{a_0, a_1\}$

Each edge is associated with a deterministic transition and a reward.

Recall that at k-th iteration, if starting from $v_0 = 0$ then one has

$$v_k(s_0) = 0, v_k(s_1) = \max\left\{\frac{\gamma}{1-\gamma}\left(1-\gamma^{k-1}\right), R\right\}, v_k(s_2) = \frac{1-\gamma^k}{1-\gamma}.$$

The greedy policy with respect to v_k at state s_1 satisfies:

$$\pi_{v_k}(s_1) = \begin{cases} a_0 & \text{if } \frac{\gamma}{1-\gamma}\left(1-\gamma^{k-1}\right) > R \\ a_1 & \text{otherwise.} \end{cases}$$

"Modified policy iteration algorithms are not strongly polynomial for discounted dynamic programming" by Eugene A. Feinberg, Jefferson Huang and Bruno Scherrer, 2014.

VI is Not Strongly Polynomial: Example

Assume $R < \frac{\gamma}{1-\gamma}$. Then $v^*(s_1) = \frac{\gamma}{1-\gamma}$ and the optimal action at s_1 is a_0. Thus the greedy policy is optimal only if:

$$\frac{\gamma}{1-\gamma}\left(1-\gamma^{k-1}\right) > R \Leftrightarrow \gamma^{k-1} < 1 - R\left(\frac{1-\gamma}{\gamma}\right) \Rightarrow k > 1 + \frac{\log\left(1 - R\left(\frac{1-\gamma}{\gamma}\right)\right)}{\log\gamma}.$$

Since $k \to \infty$ when $R \to \frac{\gamma}{1-\gamma}$, (nearly) infinite iterations are needed to produce an optimal policy.

Lemma 2

[Strict Progress Lemma] Fix an arbitrary suboptimal policy π_0 and let $\{\pi_k\}$ be the sequence of policies produced by policy iteration. Then there exists a state s_0 such that for any $k \geq \frac{1}{1-\gamma} \log\left(\frac{1}{1-\gamma}\right)$, one has

$$\pi_k\left(s_0\right) \neq \pi_0\left(s_0\right).$$

The lemma shows that after every k iterations, policy iteration eliminates one action choice at one state until there remains no suboptimal action to be eliminated. This can only be continued for at most $|\mathcal{S}||\mathcal{A}| - |\mathcal{S}|$ times: for every state, at least one action must be optimal.

"Improved and generalized upper bounds on the complexity of policy iteration" by Bruno Scherrer, 2016.

Proof of Lemma 2

The first key question is about how to measure the progress of policies. To this end, consider

$$g(\pi', \pi) = \mathcal{T}_{\pi'} v_\pi - v_\pi,$$

which can be viewed as *advantage* of π' relative to π in one-step lookahead. It is worth noting that if $g(\pi', \pi) \geq 0$, then

$$v_{\pi'} - v_\pi = (I - \gamma P^{\pi'})^{-1}(r_{\pi'} - (I - \gamma P^{\pi'})v_\pi) = (I - \gamma P^{\pi'})^{-1} g(\pi', \pi) \geq 0.$$

Moreover, it can be shown that π^* is the optimal policy if and only if

$$g(\pi, \pi^*) \leq 0 \quad \forall\, \pi.$$

Thus, we can use $-g(\pi_k, \pi^*)$ to measure the progress of π_k, which is expected to decrease to zero. It is easy to see that if

$$-g(\pi_k, \pi^*)(s) < -g(\pi_0, \pi^*)(s),$$

then $\pi_k(s) \neq \pi_0(s)$.

Proof of Lemma 2 (Cont'd)

Moreover, we have

$$-g(\pi_k, \pi^*) = (I - \gamma P^{\pi_k})(v_{\pi^*} - v_{\pi_k}) = v_{\pi^*} - v_{\pi_k} - \gamma P^{\pi_k}(v_{\pi^*} - v_{\pi_k}) \le v_{\pi^*} - v_{\pi_k}.$$

If follows that

$$\begin{aligned}
\|g(\pi_k, \pi^*)\|_\infty &\le \|v_{\pi_k} - v_{\pi^*}\|_\infty \le \gamma^k \|v_{\pi_0} - v_{\pi^*}\|_\infty \\
&= \gamma^k \|(I - \gamma P^{\pi_0})^{-1} g(\pi_0, \pi^*)\|_\infty \\
&\le \frac{\gamma^k}{1-\gamma} \|g(\pi_0, \pi^*)\|_\infty
\end{aligned}$$

Thus, there exists an s such that

$$-g(\pi_k, \pi^*)(s) < -g(\pi_0, \pi^*)(s)$$

for sufficiently large k.

Runtime Bound for Policy Iteration

Theorem 8
Let $\{\pi_k\}$ *be the sequence of policies obtained by policy iteration starting from an arbitrary initial policy* π_0*. Then, after at most*

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

iterations, the policy produced by policy iteration is optimal. In particular, policy iteration can compute an optimal policy with at most

$$O\left(\frac{|\mathcal{S}|^4|\mathcal{A}|+|\mathcal{S}|^3|\mathcal{A}|^2}{1-\gamma}\log\left(\frac{1}{1-\gamma}\right)\right)$$

arithmetic and logic operations.

Another Strongly Polynomial Approach: Linear Programing (LP)

The linear programming approach is based on an interesting fact: If a vector v satisfies $\mathcal{T}v \leq v$ then $v^* \leq v$. This means that for all $s \in \mathcal{S}$,

$$v^*(s) = \min\{v(s) : \mathcal{T}v \leq v\}.$$

Thus v^* is the unique solution of following optimization problem:

$$\begin{aligned} \min \quad & \sum_{s \in \mathcal{S}} v(s) \\ \text{s.t.} \quad & \mathcal{T}v(s) = \max_{a \in \mathcal{A}} \sum_{s'} p(s'|s,a)\left(r(s,a,s') + \gamma v(s')\right) \leq v(s),\ \forall s \in \mathcal{S}. \end{aligned}$$

This is further equivalent to LP with $|\mathcal{S}|$ unknown variables and $|\mathcal{S}| \times |\mathcal{A}|$ inequality constraints:

$$\begin{aligned} \min \quad & \sum_{s \in \mathcal{S}} v(s) \\ \text{s.t.} \quad & \sum_{s' \in \mathcal{S}} p(s'|s,a)\left(r(s,a,s') + \gamma v(s')\right) \leq v(s),\ \forall s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned}$$

"The Simplex and Policy-Iteration Methods are Strongly Polynomial for the Markov Decision Problem with a Fixed Discount Rate" by Yinyu Ye, 2011.

Table of Contents

Truncated Policy Iteration

Truncated policy iteration (TPI) is the same as policy iteration except that it merely runs a finite number of iterations in the policy evaluation step.

- **Truncated Policy Evaluation**: Set $v_{k,0} = v_{k-1}$ and estimate v_{π_k} by applying the following iteration m times:

$$v_{k,j} = r_{\pi_k} + \gamma P^{\pi_k} v_{k,j-1},$$

  where $1 \leq j \leq m_k$. Set $v_k = v_{k,m_k}$, or equivalently, $v_k = \mathcal{T}_{\pi_k}^{m_k} v_{k-1}$.

- **Policy Improvement**:

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = \arg\max_a \left\{ \sum_{s'} p(s'|s,a) \left(r(s,a,s') + \gamma v_k(s') \right) \right\} \\ 0 & \text{otherwise.} \end{cases}$$

Remark 1

If we set $m = \infty$, then $v_k = v_{\pi_k}$ and TPI is exactly PI. On the other hand, if we set $m = 1$, then $v_k = \mathcal{T} v_{k-1}$ and TPI is exactly VI.

Theorem 9

For any $m \in \mathbb{N}^+ \cup \{+\infty\}$ in the policy evaluation step and any initial condition v_{-1} (for evaluation of v_{π_0}), the sequence $\{v_k\}, \{\pi_k\}$ produced by truncated policy iteration satisfies:

$$\lim_{k\to\infty} v_k = v^* \quad \textit{and} \quad \lim_{k\to\infty} v_{\pi_k} = v^*$$

Proof of Theorem 9

The goal is to see whether v_{k+1} is comparable with $\mathcal{T}v_k$. Without loss of generality, assume $m_k = m$ for any k. First consider the case $\mathcal{T}v_{-1} \geq v_{-1}$. Then we have

$$\mathcal{T}v_{k+1} = \mathcal{T}(\mathcal{T}_{\pi_{k+1}}^{m} v_k) \geq \mathcal{T}_{\pi_{k+1}}^{m+1} v_k = \mathcal{T}_{\pi_{k+1}}^{m}(\mathcal{T}_{\pi_{k+1}} v_k) = \mathcal{T}_{\pi_{k+1}}^{m}(\mathcal{T}v_k) \geq \mathcal{T}_{\pi_{k+1}}^{m} v_k = v_{k+1},$$

where the second inequality follows from the induction hypothesis. Moreover,

$$v_{k+1} = \mathcal{T}_{\pi_{k+1}}^{m} v_k = \mathcal{T}_{\pi_{k+1}}^{m-1}(\mathcal{T}v_k) \geq \mathcal{T}_{\pi_{k+1}}^{m-1} v_k \geq \cdots \geq \mathcal{T}_{\pi_{k+1}} v_k = \mathcal{T}v_k.$$

It follows that $v_{k+1} \geq \mathcal{T}^{k+1} v_{-1}$. In addition, since $\mathcal{T}v_{k+1} \geq v_{k+1}$, one has $v_{k+1} \leq v^*$. Thus, letting $k \to \infty$ yields that $v_k \to v^*$ and $v_{\pi_k} \to v^*$ (use Theorem 4).

When $\mathcal{T}v_{-1} < v_{-1}$, we can add $c \cdot \mathbf{1}$ to v_{-1} such that $\mathcal{T}(v_{-1} + c \cdot \mathbf{1}) \geq v_{-1} + c \cdot \mathbf{1}$ for some c. Moreover, it can be shown that starting from $v_{-1} + c \cdot \mathbf{1}$ yields the same policy as starting from v_{-1}.

Approximate Policy Iteration

Approximate Policy Iteration (API) is an even more general framework than truncated policy iteration, where each policy π_k is evaluated approximately and the new policy π_{k+1} may also be generated by (approximate) policy improvement.

- **Approximate Policy Evaluation**: Given π_k, estimate v_{π_k} by v_k that satisfies

$$\left\| v_k - v^{\pi_k} \right\|_\infty \leq \delta.$$

- **Approximate Policy Improvement**: Produces a polcicy π_{k+1} that satisfies

$$\left\| r_{\pi_{k+1}} + \gamma P^{\pi_{k+1}} v_k - \mathcal{T} v_k \right\|_\infty \leq \varepsilon.$$

Convergence of Approximate Policy Iteration

Theorem 10

Let $\{\pi_k\}$ *be the sequence generated by approximate policy iteration. Then we have the following asymptotic result:*

$$\limsup_{k\to\infty} \|v_{\pi_k} - v^*\|_\infty \le \frac{\varepsilon + 2\gamma\delta}{(1-\gamma)^2}.$$

"Reinforcement Learning and Optimal Control" by Dimitri P. Bertsekas, 2019.

Proof of Theorem 10

We will make use of Lemma 1 in this proof. First note that by the algorithm,

$$\mathcal{T}_{\pi_k} v_{k-1} \geq \mathcal{T} v_{k-1} - \varepsilon \mathbf{1} \geq v\mathbf{1}.$$

Thus,

$$\begin{aligned}
v_{\pi_k} &\geq \mathcal{T}_{\pi_k} v_{k-1} - \frac{\max_s\{\mathcal{T}_{\pi_k} v_{k-1}(s) - \mathcal{T}_{\pi_k}^2 v_{k-1}(s)\}}{1-\gamma} \\
&\geq \mathcal{T}_{\pi_k} v_{k-1} - \frac{\gamma}{1-\gamma} \max_s\{v_{k-1}(s) - \mathcal{T}_{\pi_k} v_{k-1}(s)\} \\
&\geq \mathcal{T} v_{\pi_{k-1}} - (\gamma\delta + \varepsilon)\mathbf{1} - \frac{\gamma}{1-\gamma} \max_s\{v_{k-1}(s) - \mathcal{T} v_{\pi_{k-1}}(s) + (\gamma\delta + \varepsilon)\} \\
&\geq \mathcal{T} v_{\pi_{k-1}} - \frac{\varepsilon + 2\gamma\delta}{1-\gamma}\mathbf{1} \\
&\geq \cdots\cdots \\
&\geq \mathcal{T}^k v_{\pi_0} - \frac{(1-\gamma^k)(\varepsilon + 2\gamma\delta)}{1-\gamma}\mathbf{1}.
\end{aligned}$$

Taking a limit yields the result.

Questions?