Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 2: Dynamic Programming

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2023)

License Information for Theory and Methods for Reinforcement Learning (EE-618)

A refresher on Markov chain

Definition (Markov Chain)

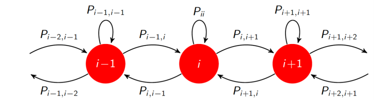
A (time-homogeneous) Markov chain is a stochastic process $\{X_0, X_1, \ldots\}$*, taking values on a countable number of states, satisfying the so-called Markov property, i.e.,*

$$\mathrm{P}(X_{t+1} = j | X_t = i, X_{t-1}, \ldots, X_0) = \mathrm{P}(X_{t+1} = j | X_t = i) = P_{ij}.$$

Markov Process

A Markov process is a tuple $\langle \mathcal{S}, P, \mu \rangle$, where

- $\mathcal{S}$ is the set of all possible states
- $P_{ss'} = \mathrm{P}(s'|s)$: $\mathcal{S} \to \Delta(\mathcal{S})$ is the transition model
- μ is the initial state distribution: $s_0 \sim \mu \in \Delta(\mathcal{S})$



Definition (Stationary distribution)

If a Markov chain is irreducible and aperiodic with finite states (i.e., ergodic), then there exists a unique stationary distribution $\mathrm{d}^\star$ *and* $\{X_t\}$ *converges to it, i.e.,* $\lim_{t\to\infty} P_{ij}^t = \mathrm{d}_j^\star, \forall i, j$*. We can represent this via* $\mathrm{d}^\star = \mathrm{d}^\star P$ *where* $[P]_{ij} = P_{ij}$ *and* $\mathrm{d}^\star$ *is a row vector. Hence,* $\mathrm{d}^\star$ *is the left principal eigenvector of* P*.*

Markov Decision Processes (MDPs)

◦ MDPs are building blocks in RL and form the Markov blanket for the rewards

◦ Also recall the (controlled) Markov property

$$\mathsf{P}(s_{t+1} = s' | s_t = s, a_t = a, \ldots, s_0, a_0) = \mathsf{P}(s_{t+1} = s | s_t = s, a_t = a) = \mathsf{P}(s'|s, a)$$

Markov Decision Process

An MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathsf{P}, r, \mu, \gamma)$, where

- $\mathcal{S}$ is the set of all possible states
- $\mathcal{A}$ is the set of all possible actions
- $\mathsf{P}(s'|s, a)$: $\mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition model
- $r(s, a)$: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function
- μ is the initial state distribution: $s_0 \sim \mu \in \Delta(\mathcal{S})$
- γ is the discount factor: $\gamma \in (0, 1)$

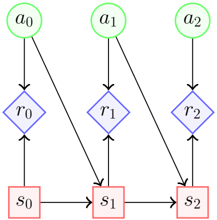
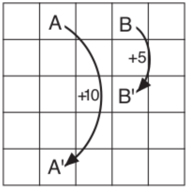


Figure: An MDP graphical model

Example 1: gridworld

- State $\mathcal{S}$: the agent's position
- Action $\mathcal{A}$: moving north/south/east/west
- Reward r:
  - -1 if moving outside the world
  - +10 if moving to A
  - +5 if moving to B
  - 0 otherwise
- Transition model P:
  - move to the adjacent grid according to the direction
  - stay unchanged if moving toward the wall
  - transit to A' if moving into A, transit to B' if moving into B



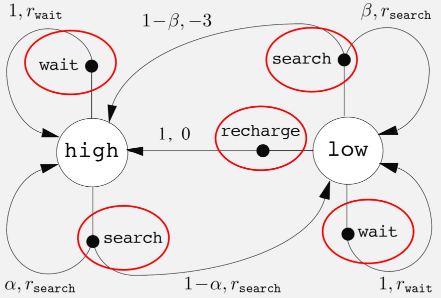
Example 2: recycling robot

◦ Consider a rechargeable mobile robot collecting empty drink cans [19]

- State $\mathcal{S}$: $\{\texttt{high}, \ \texttt{low}\}$, high/low charge level
- Action $\mathcal{A}$:
  - if $\texttt{high}$, action set $\{\texttt{wait}, \ \texttt{search}\}$
  - if $\texttt{low}$, action set $\{\texttt{wait}, \ \texttt{search}, \ \texttt{recharge}\}$
- Reward r:
  - $r(s_t = \texttt{high}, \ a_t = \texttt{search}) =$ number of collected cans
  - $r(s_t = \texttt{low}, \ a_t = \texttt{recharge}) = 0$
  - $\ldots$
- Transition model P:
  - $P(s_{t+1} = \texttt{high} \mid s_t = \texttt{high}, \ a_t = \texttt{search}) = \alpha$
  - $P(s_{t+1} = \texttt{low} \mid s_t = \texttt{low}, \ a_t = \texttt{wait}) = 1$
  - $\ldots$

Example 2: recycling robot (cont'd)

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	r_{search}
high	wait	high	1	r_{wait}
high	wait	low	0	-
low	wait	high	0	-
low	wait	low	1	r_{wait}
low	recharge	high	1	0
low	recharge	low	0	-



Remarks:

- Note that here $r(s, a, s')$ is the reward of the state-action-next-state tuple.
- While strictly not required, we can define $r(s, a) = \mathbb{E}_{s' \sim \mathrm{P}(\cdot|s,a)} \left[r(s, a, s') \right]$.

MDPs: policies

What is our goal?

Find a behaviour or rule to make decisions that maximize the expected return.

- In general, a **policy** selects an action based on the history $h_t := (s_{0:t}, a_{0:t-1}) := (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$
  - A stationary **Markov policy** is a mapping $\pi : \mathcal{S} \to \mathcal{A}$ or $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where Δ is the appropriate probability simplex.

Deterministic Policy

- Stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, $a_t = \pi(s_t)$
- Markov policy $\pi_t : \mathcal{S} \to \mathcal{A}$, $a_t = \pi_t(s_t)$
- History-dependent policy $\pi_t : \mathcal{H}_t \to \mathcal{A}$
  - $\mathcal{H}_t$ is the set of histories up to time t.
  - $a_t = \pi_t(h_t)$

Randomized Policy:

- Stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi(\cdot|s_t)$
- Markov policy $\pi_t : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi_t(\cdot|s_t)$
- History-dependent policy $\pi_t : \mathcal{H}_t \to \Delta(\mathcal{A})$
  - $\mathcal{H}_t$ is the set of histories up to time t.
  - $a_t \sim \pi_t(\cdot|h_t)$

Remarks:
- The **infinite horizon** objective can be maximized by a *stationary deterministic policy*.
- The **finite horizon** objective needs instead a (nonstationary) *deterministic Markov policy*.

From MDPs to performance criteria

Reminder:

- We have described the role of MDPs while establishing a performance criterion.
  - Finite Horizon: Cumulative reward and average reward.
  - Infinite Horizon: Discounted reward and average reward.
- In this course, we mainly focus on **infinite-horizon MDPs**:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 \sim \mu,\ \pi\right].$$

- We use $\gamma \in (0,1)$ to trade off past and present rewards.

Observations:

- If $\gamma = 1$, the total reward may be infinite, e.g., when the Markov process is cyclic.
- With $\gamma \in (0,1)$, assuming bounded rewards, i.e., $r < \infty$, the return will always be finite.

Value functions

Definition (State-Value Function)

$$V^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t, a_t) \mid s_0 = s,\ \pi\right]$$

Value functions

Definition (State-Value Function)

$$V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s,\ \pi\right]$$

Definition (Quality Function / State-Action Value Function)

$$Q^\pi(s,a) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s,\ a_0 = a,\ \pi\right]$$

Observations:

- $V^\pi(s)$ represents the total expected return starting at state s under policy π.
- $Q^\pi(s,a)$ also represents the total expected return when choosing action a in state s and then following policy π.
- For convenience, we may drop the π in RHS when it is clear from the context.

Value functions (cont'd)

Pop quiz: ◦ What is the relation between V^{π} and Q^{π}?

Value functions (cont'd)

Pop quiz: ◦ What is the relation between V^π and Q^π?

Answer: ◦ For any policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, it holds that

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\pi(s') \tag{1}$$

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a\,|\,s)\, Q^\pi(s,a) \tag{2}$$

Proof of equation (1)

Derivation:

$$\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\Big] \\
&= r(s,a) + \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\Big]
\end{aligned}$$

Proof of equation (1)

Derivation:

$$
\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\left[\sum\nolimits_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \mathbb{E}\left[\sum\nolimits_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum\nolimits_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_0 = s,\, s_1 = s',\, a_0 = a,\, \pi\right]
\end{aligned}
$$

Proof of equation (1)

Derivation:

$$
\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_0 = s,\, s_1 = s',\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_1 = s',\, \pi\right] \qquad \text{(Markov assumption)}
\end{aligned}
$$

Proof of equation (1)

Derivation:

$$
\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_0 = s,\, s_1 = s',\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_1 = s',\, \pi\right] \qquad \text{(Markov assumption)} \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t,a_t) \mid s_0 = s',\, \pi\right] \qquad \left(\text{i.e., } V^\pi(s')\right)
\end{aligned}
$$

Proof of equation (1)

Derivation:

$$
\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_0 = s,\, s_1 = s',\, a_0 = a,\, \pi\right] \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t) \mid s_1 = s',\, \pi\right] \qquad \text{(Markov assumption)} \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r(s_t,a_t) \mid s_0 = s',\, \pi\right] \qquad \left(\text{i.e., } V^\pi(s')\right) \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\, V^\pi(s') \square
\end{aligned}
$$

Occupancy measure

Definition (Occupancy measure)

The occupancy measure for a certain μ and π is defined as follows:

$$\lambda_\mu^\pi(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}[s_t = s, a_t = a \mid s_0 \sim \mu,\ \pi],$$

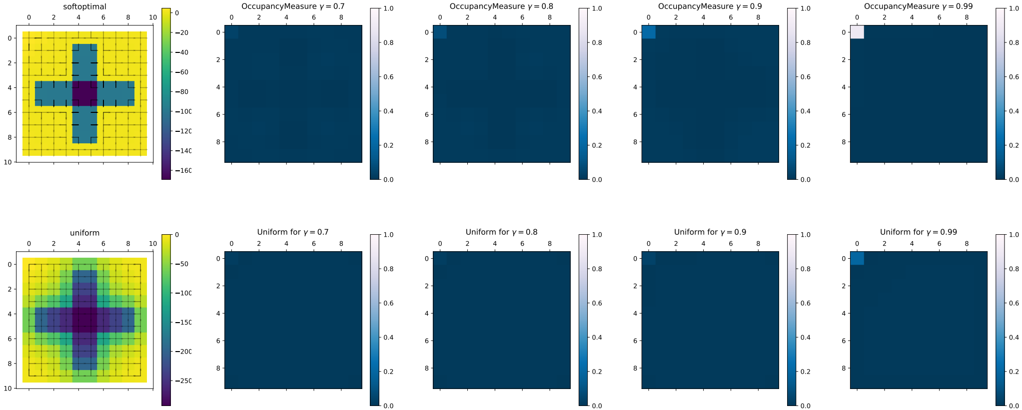
where $\mathbb{P}[\cdot \mid s_0 \sim \mu,\ \pi]$ denotes the probability of an event when following policy π starting from $s_0 \sim \mu$.

Interpretation:

- $\lambda_\mu^\pi(s,a)$ is the normalized discounted visitation frequency of the state-action pair (s,a) when acting according to policy π.

Visualize an occupancy measure

- Let's consider the policies represented by the arrows in the left most column.
- The corresponding occupancy measures varying the discounted factor are depicted just below.
- Notice that increasing γ makes the effect of the initial distribution less and less remarked.



Occupancy measure and value function

Pop quiz: ◦ What is the relation between the occupancy measure and the value function?

Occupancy measure and value function

Pop quiz: ◦ What is the relation between the occupancy measure and the value function?

Answer:

$$(1-\gamma)V^{\pi}(\mu) = \langle \lambda_{\mu}^{\pi}, r \rangle.$$

Occupancy measure and value function

Pop quiz: ◦ What is the relation between the occupancy measure and the value function?

Answer:

$$(1-\gamma)V^{\pi}(\mu) = \langle \lambda_{\mu}^{\pi}, r \rangle.$$

Remark: It holds that

$$V^{\pi}(\mu) = \langle \mu, V^{\pi} \rangle = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \mu,\ \pi\right].$$

Occupancy measure and value function (cont'd)

Derivation:

$$
\begin{aligned}
V^\pi(\mu) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \mu,\ \pi\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{s,a} r(s,a) \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right]
\end{aligned}
$$

Occupancy measure and value function (cont'd)

Derivation:

$$
\begin{aligned}
V^\pi(\mu) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \mu,\ \pi\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{s,a} r(s,a) \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] \\
&= \sum_{s,a} r(s,a)\ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] \qquad \text{(Linearity of expectation)}
\end{aligned}
$$

Occupancy measure and value function (cont'd)

Derivation:

$$
\begin{aligned}
V^\pi(\mu) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \mu,\ \pi\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{s,a} r(s,a) \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] \\
&= \sum_{s,a} r(s,a)\ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] && \text{(Linearity of expectation)} \\
&= \sum_{s,a} r(s,a) \sum_{t=0}^{\infty} \gamma^t\ \mathbb{P}[(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi] && \text{(Dominated convergence theorem)}
\end{aligned}
$$

Occupancy measure and value function (cont'd)

Derivation:

$$
\begin{aligned}
V^\pi(\mu) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \mu,\ \pi\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{s,a} r(s,a)\mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] \\
&= \sum_{s,a} r(s,a)\ \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi\right] && \text{(Linearity of expectation)} \\
&= \sum_{s,a} r(s,a) \sum_{t=0}^{\infty} \gamma^t\ \mathbb{P}[(s_t = s, a_t = a) \mid s_0 \sim \mu,\ \pi] && \text{(Dominated convergence theorem)} \\
&= \frac{\sum_{s,a} r(s,a)\lambda_\mu^\pi(s,a)}{1-\gamma} = \frac{\langle \lambda_\mu^\pi, r\rangle}{1-\gamma}. \ \square
\end{aligned}
$$

Optimal value functions

◦ Let Π be the set of all (possibly non-stationary and randomized) policies.

Definition (Optimal Value Function)

$$V^\star(s) := \max_{\pi \in \Pi} V^\pi(s)$$

Definition (Optimal State Value Function)

$$Q^\star(s,a) := \max_{\pi \in \Pi} Q^\pi(s,a)$$

Pop quiz: ◦ What is the relation between $V^\star$ and $Q^\star$?

Optimal value functions

∘ Let Π be the set of all (possibly non-stationary and randomized) policies.

Definition (Optimal Value Function)

$$V^\star(s) := \max_{\pi \in \Pi} V^\pi(s)$$

Definition (Optimal State Value Function)

$$Q^\star(s,a) := \max_{\pi \in \Pi} Q^\pi(s,a)$$

Pop quiz: ∘ What is the relation between $V^\star$ and $Q^\star$?

Answer:

$$Q^\star(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s') \tag{3}$$

$$V^\star(s) = \max_{a \in \mathcal{A}} Q^\star(s,a) \tag{4}$$

∘ *Self-exercise*: prove equation (4).

Solving MDPs: find the optimal policy

The ultimate goal in RL

To find an **optimal policy** $\pi^\star \in \Pi$ such that

$$V^{\pi^\star}(s) = V^\star(s) := \max_{\pi \in \Pi} V^\pi(s), \forall s \in \mathcal{S}.$$

Remark: ◦ The optimal policy may not be unique, while $V^\star$ is unique.

Key Questions

- **Q1**: Does the optimal policy $\pi^\star$ exist?
- **Q2**: How to evaluate my current policy π, i.e., how to compute $V^\pi(s)$? ***–policy evaluation***
- **Q3**: If $\pi^\star$ exists, how to improve my current policy π, i.e., how to find $\pi^\star$? ***–policy improvement***

Bellman optimality conditions

◦ The optimal value function $V^\star$ is the unique **fixed point** of the following equation:

$$V^\star(s) = \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V^\star(s') \right].$$

Remarks:

◦ This requirement is also known as the **Bellman optimality equation**.

◦ We will show that there exists a **deterministic** optimal policy.

◦ Fixed-point perspective motivates value iteration (VI) and policy iteration (PI) methodologies.

Existence of an optimal policy

Theorem (Existence of an optimal policy [1] [12])

For an infinite horizon MDP $M = (\mathcal{S}, \mathcal{A}, P, t, \mu, \gamma)$, there exists a stationary and deterministic policy π such that for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, we have

$$V^{\pi}(s) = V^{\star}(s), \quad Q^{\pi}(s, a) = Q^{\star}(s, a).$$

Remark:

- Finding $\pi^{\star}$ can be done by first computing $V^{\star}$ or $Q^{\star}$
- Note that we can directly get a (deterministic and stationary) optimal policy from $Q^{\star}$:

$$\pi^{\star}(s) = \arg\max_{a \in \mathcal{A}} Q^{\star}(s, a).$$

- Note: Proof in the supplementary.

Bellman consistency equation

Richard Ernest Bellman
(August 26, 1920 – March 19, 1984)

Theorem (Bellman Consistency Equation)

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V^{\pi}(s')\right]$$

Bellman consistency equation

Richard Ernest Bellman
(August 26, 1920 – March 19, 1984)

Theorem (Bellman Consistency Equation)

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)V^{\pi}(s') \right]$$

Matrix Form

$$\mathbf{V}^{\pi} = \mathbf{R}^{\pi} + \gamma \mathsf{P}^{\pi}\mathbf{V}^{\pi}$$

- Can be derived from equations (1) and (2):

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V^{\pi}(s') \quad (1)$$

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}} \pi(a\,|\,s)\, Q^{\pi}(s,a) \quad (2)$$

- We can write, with $|\mathcal{S}|$ being the cardinality of $\mathcal{S}$:

$$\mathbf{V}^{\pi} \in \mathbb{R}^{|\mathcal{S}|} : \mathbf{V}^{\pi}_s = V^{\pi}(s);$$

$$\mathbf{R}^{\pi} \in \mathbb{R}^{|\mathcal{S}|}, \mathbf{R}^{\pi}_s := \sum_{a\in\mathcal{A}} \pi(a|s)r(s,a);$$

$$\mathbf{P}^{\pi} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|} : \mathbf{P}^{\pi}_{s,s'} := \sum_{a\in\mathcal{A}} \pi(a|s)P(s'\,|\,s,a).$$

Closed-form solution for policy evaluation

Closed-Form Solution of $\mathbf{V}^\pi$

$$\mathbf{V}^\pi = (\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1} \mathbf{R}^\pi.$$

Remarks:

- This is one of **exact solution methods** for policy evaluation.
- Note that the matrix $\mathbf{I} - \gamma \mathbf{P}^\pi$ is always invertible for $\gamma \in (0,1)$.
- The solution of Bellman equation is always unique.
- Computation cost: $\mathcal{O}(|\mathcal{S}|^3 + |\mathcal{S}|^2|\mathcal{A}|)$, which can be expensive for large state spaces.

Bellman expectation operator and fixed-point perspective

Definition (Bellman Expectation Operator)

Let $\mathcal{T}^\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ *be that*

$$\mathcal{T}^\pi \mathbf{V} := \mathbf{R}^\pi + \gamma \mathbf{P}^\pi \mathbf{V} \tag{5}$$

Remarks:

- The Bellman equation implies that $\mathbf{V}^\pi$ is the **fixed point** of $\mathcal{T}^\pi$: $\mathcal{T}^\pi \mathbf{V}^\pi = \mathbf{V}^\pi$.
- $\mathcal{T}^\pi$ is a linear operator and is a γ-contraction mapping.
- The solution of Bellman equation is always unique.
- Fixed point iteration: $\mathbf{V}_{t+1} = \mathcal{T}^\pi \mathbf{V}_t$, $t = 0, 1, \ldots$.
- $\lim_{t\to\infty} (\mathcal{T}^\pi)^t \mathbf{V}_0 = \mathbf{V}^\pi$.

Bellman optimality equations

Theorem (Bellman Optimality Equation)

$$V^\star(s) = \max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) V^\star(s') \right]$$

$$Q^\star(s,a) = r(s,a) + \gamma \left[\sum_{s' \in \mathcal{S}} P(s'|s,a) \left(\max_{a' \in \mathcal{A}} Q^\star(s',a') \right) \right]$$

Remarks:

- These requirement are also known as **Bellman optimality conditions**.
- Obtained by combining equations (3) and (4).
- Fixed-point perspective motivates value iteration (VI) and policy iteration (PI) methodologies.

Bellman optimality perator

Definition (Bellman Optimality Operator)

$$(\mathcal{T}\mathbf{V})(s) := \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)\mathbf{V}(s') \right]$$

Remarks:

- The optimal value function $\mathbf{V}^\star$ is the **fixed point** of $\mathcal{T}$, i.e.,

$$\mathcal{T}\mathbf{V}^\star = \mathbf{V}^\star$$

- The Bellman optimality operator is a γ-contraction mapping w.r.t. ℓ_∞-norm.
- The Bellman operator is also monotonic (component-wise): $\mathbf{V}_1 \le \mathbf{V}_2 \;\Rightarrow\; \mathcal{T}\mathbf{V}_1 \le \mathcal{T}\mathbf{V}_2$.
- We can define a similar Bellman operator on the Q-function and show similar properties.

Contraction of bellman optimality operator

Theorem (Contraction Property of $\mathcal{T}$)

The Bellman optimality operator $\mathcal{T}$ *defined above is a* γ*-contraction mapping under* ℓ_∞*-norm, i.e., for any* $\mathbf{V}', \mathbf{V} \in \mathbb{R}^{|\mathcal{S}|}$*, we have*

$$\left\|\mathcal{T}\mathbf{V}' - \mathcal{T}\mathbf{V}\right\|_\infty \leq \gamma \left\|\mathbf{V}' - \mathbf{V}\right\|_\infty,$$

where $\|\mathbf{x}\|_\infty := \max_i |x_i|$.

Contraction of bellman optimality operator (proof)

Proof.

For any $\mathbf{V}', \mathbf{V} \in \mathbb{R}^{|\mathcal{S}|}$ and $s \in \mathcal{S}$, we have

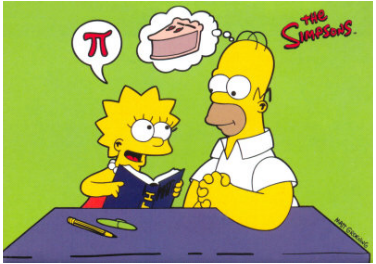
$$
\begin{aligned}
&\left|\left(\mathcal{T}\mathbf{V}'\right)(s) - (\mathcal{T}\mathbf{V})(s)\right| \\
&= \left|\max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)\mathbf{V}'(s')\right] - \max_{a'\in\mathcal{A}} \left[r(s,a') + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a')\mathbf{V}(s')\right]\right| \\
&\leq \max_{a\in\mathcal{A}} \left|\left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)\mathbf{V}'(s')\right) - \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)\mathbf{V}(s')\right)\right| \\
&\leq \max_{a\in\mathcal{A}} \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)\left|\mathbf{V}'(s') - \mathbf{V}(s')\right| \\
&\leq \left\|\mathbf{V}' - \mathbf{V}\right\|_\infty \max_{a\in\mathcal{A}} \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a) = \gamma \left\|\mathbf{V}' - \mathbf{V}\right\|_\infty,
\end{aligned}
$$

which concludes the proof. □

Pause and reflect

◦ Before we move on, take a minute to reflect on these important notations:

▷ π, $\pi^\star$, $V^\pi(s)$, $V^\star(s)$, $Q^\pi(s,a)$, $Q^\star(s,a)$, $\mathcal{T}^\pi$, $\mathcal{T}$



Solving MDPs

- What we talked about:
  - Optimal state-value Function ($V^\star(s)$) and optimal action-value Function ($Q^\star(s,a)$).
  - Bellman consistency equation ($\mathbf{V}^\pi = \mathbf{R}^\pi + \gamma \mathbf{P}^\pi \mathbf{V}^\pi$).
  - Bellman expectation operator and fixed-point perspective ($\mathcal{T}^\pi \mathbf{V} := \mathbf{R}^\pi + \gamma \mathbf{P}^\pi \mathbf{V}$)
  - Bellman optimality equations and Bellman optimality operator.
- How do we use this to do "planning," i.e., finding an optimal policy via MDPs (our goal)?

Algorithm	**Component**	**Output**
Value Iteration (VI)	Bellman Optimality Operator $\mathcal{T}$	V_T such that $\|V_T - V^\star\| \leq \epsilon$
Policy Iteration (PI)	Bellman Operator $\mathcal{T}^\pi$ + Greedy Policy	$V^\star$ and $\pi^\star$

Observation: ◦ These solutions require, and we assume throughout, that the transitions dynamics are known.

Value iteration (VI)

Algorithm: Value Iteration (VI) for solving MDPs

Start with an arbitrary guess $\mathbf{V}_0$ (e.g., $\mathbf{V}_0(s) = 0$ for any s)
for each iteration t **do**
Apply the Bellman operator $\mathcal{T}$ at each iteration

$$\mathbf{V}_{t+1} = \mathcal{T}\mathbf{V}_t.$$

end for

Remarks:
- Finding $V^\star$ or $\pi^\star$ is equivalent to finding a fixed point of $\mathcal{T}$.
- **Value iteration** can be therefore viewed as a fixed-point iteration.

Discussion on value iteration

◦ After obtaining $\mathbf{V}^\star$ via VI, we can obtain an optimal policy from the **greedy policy**:

$$\pi^\star(s) = \arg\max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s') \right].$$

Discussion on value iteration

- After obtaining $\mathbf{V}^\star$ via VI, we can obtain an optimal policy from the **greedy policy**:

$$\pi^\star(s) = \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s') \right].$$

- Alternatively, we can run Q-**value iteration** and compute $\pi^\star$ via

$$\pi^\star(s) = \arg\max_{a \in \mathcal{A}} \; Q^\star(s,a)$$

Remark: ◦ Can be derived from equations (1) and (2):

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\pi(s') \tag{1}$$

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \, Q^\pi(s,a) \tag{2}$$

Convergence of value iteration

Theorem (Linear Convergence of Value Iteration)

The value iteration algorithm attains a linear convergence rate, i.e.,

$$\|\mathbf{V}_t - \mathbf{V}^\star\|_\infty \leq \gamma^t \|\mathbf{V}_0 - \mathbf{V}^\star\|_\infty$$

Convergence of value iteration

Theorem (Linear Convergence of Value Iteration)

The value iteration algorithm attains a linear convergence rate, i.e.,

$$\|\mathbf{V}_t - \mathbf{V}^\star\|_\infty \leq \gamma^t \|\mathbf{V}_0 - \mathbf{V}^\star\|_\infty$$

Proof.

$$\|\mathbf{V}_t - \mathbf{V}^\star\|_\infty = \|\mathcal{T}\mathbf{V}_{t-1} - \mathcal{T}\mathbf{V}^\star\|_\infty \leq \gamma\|\mathbf{V}_{t-1} - \mathbf{V}^\star\|_\infty \leq \cdots \leq \gamma^t\|\mathbf{V}_0 - \mathbf{V}^\star\|_\infty$$

□

Directly update the policy

◦ Value iteration first finds $\mathbf{V}^\star$, then computes the optimal policy $\pi^\star$ by the greedy policy.

Directly update the policy

◦ Value iteration first finds $\mathbf{V}^\star$, then computes the optimal policy $\pi^\star$ by the greedy policy.

◦ Now we **directly search for the optimal policy** $\pi^\star$.

Some intuition: ◦ Start from an initial guess π, iteratively perform:

1. Evaluate policy: compute the value function $\mathbf{V}^\pi$ of the current policy
   $\Rightarrow$ Policy evaluation
2. Improve policy: update the guess by the greedy policy w.r.t. $\mathbf{V}^\pi$
   $\Rightarrow$ Policy improvement

Policy improvement theorem

Theorem (Policy Improvement)

If a (deterministic) policy π' satisfies the following

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s) \quad \forall\ s \in \mathcal{S}, \tag{6}$$

then we have $V^{\pi'}(s) \geq V^{\pi}(s)$ for any $s \in \mathcal{S}$.

Remarks:

- Improving the current policy by one step everywhere, we can improve the whole policy.
- It suggests a natural way of improving the current policy via

$$\pi_{t+1}(s) \leftarrow \arg\max_{a \in \mathcal{A}} Q^{\pi_t}(s, a).$$

- Indeed, $V^{\pi_{t+1}}(s) \geq V^{\pi_t}(s), \forall\ s \in \mathcal{S}$, and the inequality is strict if π_t is suboptimal.

Policy iteration

Algorithm: Policy Iteration (PI) for solving MDPs

Start with an arbitrary policy guess π_0

for each iteration t **do**

(Step 1: Policy evaluation) Compute $\mathbf{V}^{\pi_t}$:

(Option 1) Iteratively apply policy value iteration, $\mathbf{V}_t \leftarrow \mathcal{T}^{\pi_t}\mathbf{V}_t$, until convergence

(Option 2) Use the closed-form solution: $\mathbf{V}^{\pi_t} = (\mathbf{I} - \gamma \mathbf{P}^{\pi_t})^{-1}\mathbf{R}^{\pi_t}$

(Step 2: Policy improvement) Update the current policy π_t by the greedy policy

$$\pi_{t+1}(s) = \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^{\pi_t}(s') \right]. \tag{7}$$

end for

Remarks:

- Recall that we assume that there exists a **deterministic** optimal policy.
- Greedy policy achieves the optimal deterministic policy.

Comparison

Algorithm	Value Update	Policy Update
Value Iteration (VI)	$\mathbf{V}_{t+1} = \mathcal{T}\mathbf{V}_t.$	None
Policy Iteration (PI)	$V_{t+1} = \mathbb{E}\left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'\vert s,a)V(s')\vert\pi_t\right]$	Greedy Policy

Algorithm	Per iteration cost	Number of iterations	Output
Value Iteration (VI)	$\mathcal{O}\left(\vert\mathcal{S}\vert^2\vert\mathcal{A}\vert\right)$	$T = \mathcal{O}\left(\frac{\log(\epsilon^{-1}(1-\gamma))}{\log\gamma}\right)$	V_T such that $\Vert V_T - V^\star\Vert \le \epsilon$
Policy Iteration (PI)	$\mathcal{O}\left(\vert\mathcal{S}\vert^3 + \vert\mathcal{S}\vert^2\vert\mathcal{A}\vert\right)$	$T = \mathcal{O}\left(\frac{\vert\mathcal{S}\vert(\vert\mathcal{A}\vert-1)}{1-\gamma}\right)$	$V^\star$ and $\pi^\star$

Observations:

- VI and PI are broadly dynamic programming approaches.
- PI converges in finite number of iterations [14] whereas VI does not [13].
- These solution mythologies are broadly known as model-based RL.
- *Additional reading*: Modified Policy Iteration [15]

Convergence of policy iteration

Theorem (Linear Convergence of Policy Iteration)

The policy iteration attains a linear convergence rate,

$$\|\mathbf{V}^{\pi_t} - \mathbf{V}^{\star}\|_{\infty} \leq \gamma^t \|\mathbf{V}^{\pi_0} - \mathbf{V}^{\star}\|_{\infty}$$

Convergence of policy iteration

Theorem (Linear Convergence of Policy Iteration)

The policy iteration attains a linear convergence rate,

$$\|\mathbf{V}^{\pi_t} - \mathbf{V}^\star\|_\infty \leq \gamma^t \|\mathbf{V}^{\pi_0} - \mathbf{V}^\star\|_\infty$$

Proof.

$$\|\mathbf{V}^{\pi_t} - \mathbf{V}^\star\|_\infty \leq \|\mathcal{T}\mathbf{V}^{\pi_{t-1}} - \mathcal{T}\mathbf{V}^\star\|_\infty \leq \gamma \|\mathbf{V}^{\pi_{t-1}} - \mathbf{V}^\star\|_\infty \leq \cdots \leq \gamma^t \|\mathbf{V}_0 - \mathbf{V}^\star\|_\infty$$

□

Convergence of policy iteration

Theorem (Linear Convergence of Policy Iteration)

The policy iteration attains a linear convergence rate,

$$\|\mathbf{V}^{\pi_t} - \mathbf{V}^\star\|_\infty \le \gamma^t \|\mathbf{V}^{\pi_0} - \mathbf{V}^\star\|_\infty$$

Proof.

$$\|\mathbf{V}^{\pi_t} - \mathbf{V}^\star\|_\infty \le \|\mathcal{T}\mathbf{V}^{\pi_{t-1}} - \mathcal{T}\mathbf{V}^\star\|_\infty \le \gamma\|\mathbf{V}^{\pi_{t-1}} - \mathbf{V}^\star\|_\infty \le \cdots \le \gamma^t\|\mathbf{V}_0 - \mathbf{V}^\star\|_\infty$$

□

Remarks:

- In fact, with some extra work, it is possible to show a stronger result.
- Policy Iteration converges to the optimum in at most $\mathcal{O}\left(\frac{|S|(|A|-1)}{1-\gamma}\right)$ [14].
- Due to the discrete nature of actions, the proof is conceptually simple.

Summary I

- Basic concepts of Markov decision process (MDP)
  - Policy, value functions, optimal value functions
  - Bellman equations and Bellman operators
  - Fixed point viewpoints
  - Existence and construction of optimal policy

Summary I

- Basic concepts of Markov decision process (MDP)
  - Policy, value functions, optimal value functions
  - Bellman equations and Bellman operators
  - Fixed point viewpoints
  - Existence and construction of optimal policy
- Exact solution methods for policy evaluation

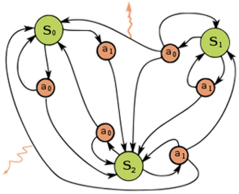
Summary I

- Basic concepts of Markov decision process (MDP)
  - Policy, value functions, optimal value functions
  - Bellman equations and Bellman operators
  - Fixed point viewpoints
  - Existence and construction of optimal policy
- Exact solution methods for policy evaluation
- Exact solution methods for solving MDPs
  - Value iteration: iteratively apply Bellman operator
  - Policy iteration: alternatively execute policy evaluation and policy improvement

From planning to reinforcement learning

Fundamental Challenge 1

The dynamic programming approaches (VI and PI) as well as the linear programming approach all require the full knowledge of the transition model P and the reward.

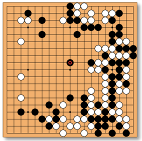


Fundamental Challenge 2

The computation and memory cost can be very expensive for large scale MDP problems.

Chess: 10^{120}

Go: 3^{361}

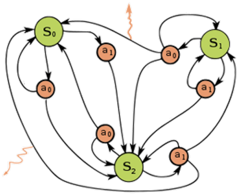


From planning to reinforcement learning

Fundamental Challenge 1

The dynamic programming approaches (VI and PI) as well as the linear programming approach all require the full knowledge of the transition model P and the reward.

$\Rightarrow$Need sampling approaches

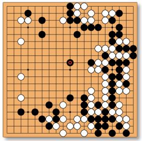


Fundamental Challenge 2

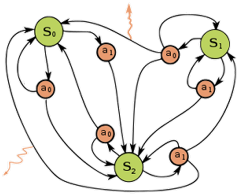
The computation and memory cost can be very expensive for large scale MDP problems.

Chess: 10^{120}

Go: 3^{361}



From planning to reinforcement learning

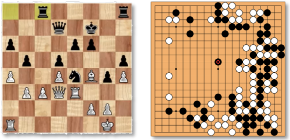


Fundamental Challenge 1

The dynamic programming approaches (VI and PI) as well as the linear programming approach all require the full knowledge of the transition model P and the reward.

$\Rightarrow$Need sampling approaches

Chess: 10^{120} Go: 3^{361}

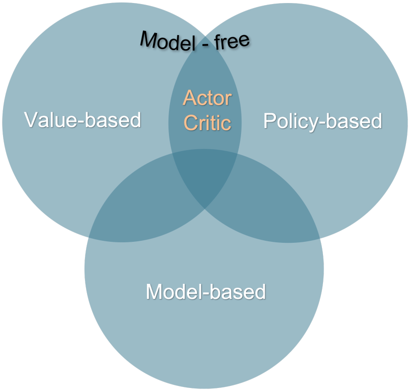


Fundamental Challenge 2

The computation and memory cost can be very expensive for large scale MDP problems.

$\Rightarrow$Need new representations

Overview of reinforcement learning approaches



- **Value-based RL**
  - Learn the optimal value functions $V^\star, Q^\star$
- **Policy-based RL**
  - Learn the optimal policy $\pi^\star$
- **Model-based RL**
  - Learn the model P, R and then do planning

Model-based vs model-free methods

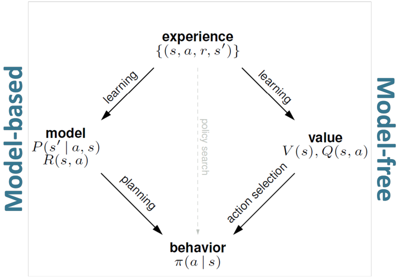


Figure: [8]

- Make full use of "experiences"
- Can reason about model uncertainty
- Sample efficient for easy dynamics

- Direct and simple
- Not affected by poor model estimation
- Not sample efficient

Online vs. offline reinforcement learning

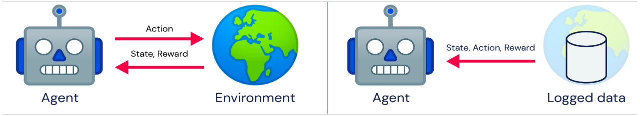


Figure: [4]

Online RL

- Collect data by interacting with environment
- Exploitation-exploration tradeoff

Offline/Batch RL

- Use previously collected data
- Data is static, no online data collection

On-policy vs. Off-policy reinforcement learning

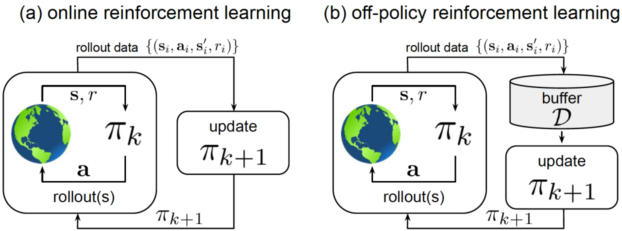


Figure: [11]

On-policy RL

◦ Learn based on data from current policy

◦ Always online

Off-policy RL

◦ Learn based on data from other policies

◦ Can be online or offline

Representation learning

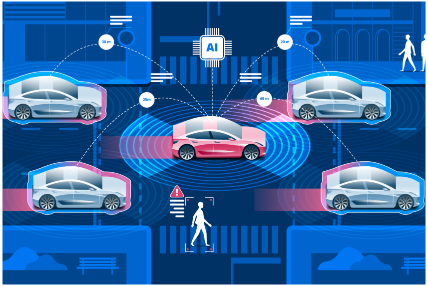


Figure: http://selfdrivingcars.space/?p=68

$\Longrightarrow$

Function approximation

$$V(s) \approx V_\theta(s)$$
$$Q(s,a) \approx Q_\theta(s,a)$$
$$\pi(a|s) \approx \pi_\theta(a|s)$$
$$\mathsf{P}(s'|s,a) \approx \mathsf{P}_\theta(s'|s,a)$$

Large or continuous state and action spaces

Representation learning

Linear Function Approximation

- Linear combination of basis functions

$$V_\theta(s) = [\phi_1(s), \dots, \phi_d(s)] \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix}$$

- Reproducing kernel Hilbert space (RKHS) [16]
- Neural tangent kernel [7]

Nonlinear Function Approximation

- Fully connected neural networks [10]
- Convolutional neural networks [9]
- Residual networks [5]
- Recurrent networks [6]
- Self-attention [21]
- Generative adversarial networks [3]

Mean estimation

- Given a sequence of samples $X_1, X_2, \dots, X_n$, we want to estimate the mean $\mu = \mathbb{E}[X]$.
- **Sample average approximation**:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Equivalently,

$$\hat{\mu}_n = \frac{1}{n}\left(X_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} X_i\right) = \hat{\mu}_{n-1} + \frac{1}{n}(X_n - \hat{\mu}_{n-1})$$

- **Stochastic approximation**:

$$\mu_{n+1} = \mu_n + \alpha_n(X_{n+1} - \mu_n), n = 1, 2, \dots$$

Remark: ◦ $\mu_n \to \mu$ as $n \to \infty$ under Robbins-Monro stepsize, i.e., $\sum_n \alpha_n = \infty, \sum_n \alpha_n^2 < \infty$.

Model-free prediction

Goal:

Given policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, estimate $V^\pi(s)$ or $Q^\pi(s,a)$ from episodes of experience under π

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

Monte Carlo method

Idea: ◦ Estimate $V^\pi(s)$ by the average of returns following all visits to s.

Monte Carlo Method

for each episode **do**
  Generate an episode $\tau = \{s_0, a_0, r_0, s_1, \ldots\}$ following π
  for each state s_t **do**
    Compute return $G_t = r_t + \gamma r_{t+1} + \cdots$
    Update counter $n_{s_t} \leftarrow n_{s_t} + 1$
    Update $V(s_t) \leftarrow V(s_t) + \frac{1}{n_{s_t}}(G_t - V(s_t))$
  end for
end for

Observations: ◦ The value estimates are independent and do not build on that of other state.

◦ Learning can be slow when the episodes are long.

◦ Convergence: MC converges to V^π if each state is visited infinitely often.

Temporal difference learning

◦ Recall the Bellman consistency equation: $V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \mathbb{E}_{s'\sim \mathsf{P}(s'|s,a)} V^\pi(s')\right]$

Idea: ◦ Incrementally estimate $V^\pi(s)$ by the intermediate return plus estimated return at next state.

$$V(s_t) \leftarrow V(s_t) + \alpha_t \Big(\underbrace{r_t + \gamma V(s_{t+1}) - V(s_t)}_{\text{TD error}:=\delta_t} \Big)$$

TD Learning / TD(0)

for each step of an episode τ **do**
  Observe (s_t, a_t, r_t, s_{t+1}) following π
  Update $V(s_t) \leftarrow V(s_t) + \alpha_t(r_t + \gamma V(s_{t+1}) - V(s_t))$
end for

Observations:

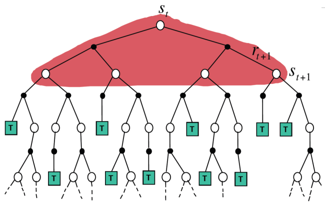
◦ Similar to mean estimation but now we have biased estimates!

◦ Similar to MC: learn directly from episodes of experiences without the MDP knowledge.

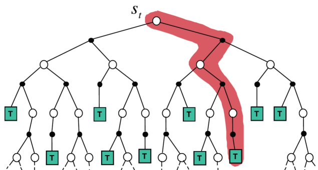
◦ Unlike MC: learn from incomplete episodes, and applicable to non-terminating environment.

◦ Convergence: $V \to V^\pi$ if each state is visited infinitely often and $\alpha_t \to 0$ at suitable rate.

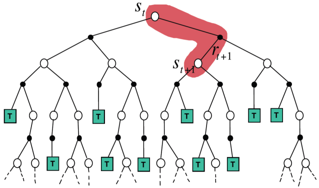
DP vs MC vs TD



DP update



MC update



TD update

- DP: no sampling, exploits Markov property
- MC: sampling, model-free, does not exploit Markov property
- TD: sampling, model-free, online, exploits Markov property

(Figure from Hasselt, UCL 2021)

Numerical example: Random walk

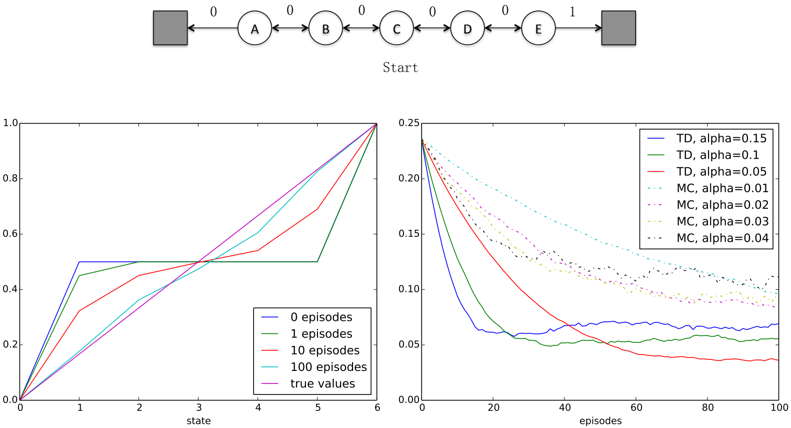


Figure: Left: values learned after various number of updates in a single run of TD(0). Right: the root mean-squared (RMS) error between the value functions learned and the true values. [18]

Bias-variance trade-off

◦ MC return is **unbiased**, but has **higher variance** since it relies on many random steps

◦ TD target is **biased**, but has **lower variance** since it only relies on the next step

◦ The MC error can be written as a sum of TD errors:

$$\begin{aligned} G_t - V(s_t) &= r_{t+1} + \gamma G_{t+1} - V(s_t) + \gamma V(s_{t+1}) - \gamma V(s_{t+1}) \\ &= \delta_t + \gamma (G_{t+1} - V(s_{t+1})) \\ &= \delta_t + \gamma \delta_{t+1} + \gamma^2 (G_{t+2} - V(s_{t+2})) \\ &= \cdots \\ &= \sum_{k=t}^{\infty} \gamma^{k-t} \delta_k \end{aligned}$$

Multiple-step TD learning

Definition (n-step return)

Let T be the termination time step in a given episode, $\gamma \in [0,1]$.

$$
\begin{aligned}
G_t^{(1)} &= r_{t+1} + \gamma V(s_{t+1}) && \textit{TD(0)} \\
G_t^{(2)} &= r_{t+1} + \gamma r_{t+2} + \gamma^2 V(s_{t+2}) && \textit{(two-step return)} \\
G_t^{(n)} &= r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n V(s_{t+n}) && \textit{(n-step return)} \\
G_t^{(\infty)} &= r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{T-t-1} r_T && \textit{MC}
\end{aligned}
$$

Note that $G_t^{(n)} = G_t^{(\infty)}$ if $t + n \geq T$.

Multiple-step TD learning

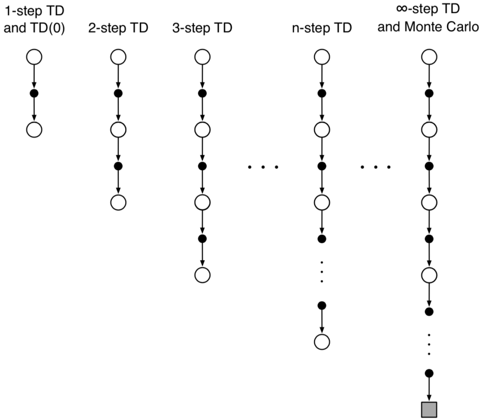


Figure: [19]

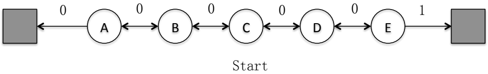
Multiple-step TD learning

Multi-step TD learning:

$$V(s_t) \leftarrow V(s_t) + \alpha_t \Big(\underbrace{G_t^{(n)} - V(s_t)}_{n\text{-step TD error}} \Big)$$

Observations:
- Unifies and combines TD(0) and MC: $n = 1$ recovers TD(0) and $n = \infty$ recovers MC.
- Trades-off bias and variance.
- However, we need to observe $r_{t+1}, \cdots, r_{t+n}$.

Numerical example: Longer random walk



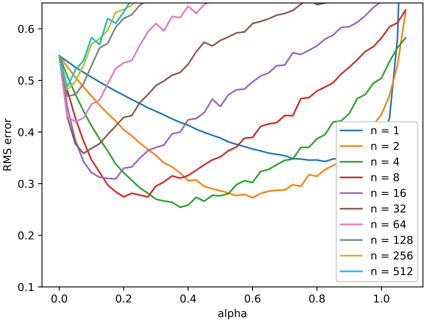


Figure: Performance of n-step TD methods as a function of α, for various values of n, on a 19-state random walk task. [18]

Further extension: TD(λ) with eligibility trace

λ-return (weighted average of all n-step returns)

$$G_t^{\lambda} = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

TD(λ)

$$V(s_t) \leftarrow V(s_t) + \alpha\left[G_t^{\lambda} - V(s_t)\right]$$

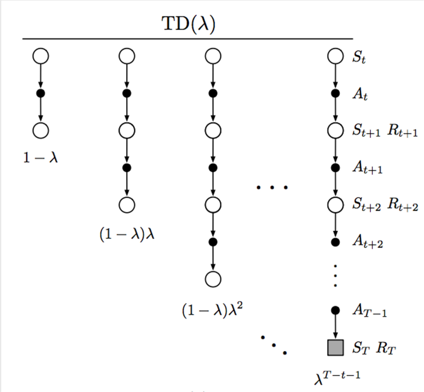


Figure: [19]

Further extension: TD(λ) with eligibility trace

TD(λ)

$$V(s_t) \;\leftarrow\; V(s_t) + \alpha\left[G_t^{\lambda} - V(s_t)\right]$$

Observations:

- $\lambda = 0$ reduces to TD(0); $\lambda = 1$ reduces to MC.
- Can be efficiently implemented:

$$\begin{aligned} V(s) &\leftarrow V(s) + \alpha\delta_t e_t(s) \\ e_t(s) &= \gamma\lambda e_{t-1}(s) + \mathbf{1}\{s_t = s\} \end{aligned}$$

- The term $e_t(s) = \sum_{k=0}^{t} \gamma^{t-k}\mathbf{1}\{s_t = s\}$ is called the eligibility trace.
- Converge faster than TD(0) when λ is appropriately chosen.

State-Action-Reward-State-Action for Q-value estimation

- In VI or PI, we often require the evaluation of Q-function to compute the greedy policy or optimal policy.

- How do we estimate Q^π?

SARSA [17]

for each step **do**
  Observe $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ following π
  Update $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(r_t + \gamma Q(s_{t+1}, a_{+1}) - Q(s_t, a_t))$
end for

Summary: Model-free prediction

Methods	**DP**	**MC**	**TD(0)**
Model knowledge	Need	No need	No need
Bootstrap	Yes	No	Yes
When do updates	After next step	After whole episode	After next step
Use Markov property	Yes	No	Yes
Bias	-	Unbiased	Biased
Variance	-	Big	Small
Convergence rate	Linear rate	-	$\mathcal{O}(1/\sqrt{t})$ [20]

Reference: [19]

Wrap Up

- PI and VI are dynamic programming methods applicable when the transition matrix is known.
- The occupancy measure is a useful quantity that will be used throughout the course.
- Monte Carlo methods are used to estimate value function when the transition matrix is unknown.
- Monte Carlo methods are an instance of stochastic approximation.
- TD is an application of dynamic programming when the transition matrix is unknown.
- **Next week** is about **Linear Programming for RL !**

Supplementary material

Existence of an optimal policy (proof)

Proof Sketch

Assume starting from $(s_0, a_0, r_0, s_1) = (s, a, r, s')$,

1. *Define "offset" policy* $\tilde{\pi}(a_t = a \,|\, h_t) := \pi(a_{t+1} = a \,|\, (s_0, a_0) = (s, a), h_t)$, *Markov property implies*

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t) \,|\, (s_0, a_0, r_0, s_1) = (s, a, r, s'),\ \pi\right] = \gamma V^{\tilde{\pi}}(s')$$

2. *With all* $(s_0, a_0, r_0) = (s, a, r)$, *the set* $\{\tilde{\pi} \,|\, \Pi\}$ *will just be* Π *itself*
3. *Show that the optimal value from* s_1 *onward is independent of* $(s_0, a_0, r_0) = (s, a, r)$,

$$\max_{\pi \in \Pi} \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t, a_t) \,|\, (s_0, a_0, r_0, s_1) = (s, a, r, s'),\ \pi\right] = \gamma \max_{\pi \in \Pi} V^{\tilde{\pi}}(s') = \gamma \max_{\pi \in \Pi} V^{\pi}(s') = \gamma V^{\star}(s')$$

Existence of an optimal policy (proof)

Proof Sketch (cont.)

4. *Let* $\pi(s) = \arg\max_{a\in\mathcal{A}} \max_{\pi'\in\Pi} Q^{\pi'}(s,a)$, *show this deterministic and randomized policy is optimal*

$$
\begin{aligned}
V^*(s_0) = & \max_{\pi'\in\Pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s,\ \pi\right] = \max_{\pi'\in\Pi} \mathbb{E}\left[r(s_0,a_0) + \sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid \pi\right] \\
= & \max_{\pi'\in\Pi} \mathbb{E}\left[r(s_0,a_0) + \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t r(s_t,a_t) \mid (s_0,a_0,r_0,s_1),\pi\right]\right] \\
\leq & \max_{\pi'\in\Pi} \mathbb{E}\left[r(s_0,a_0) + V^\star(s_1)\right] \quad \Longleftarrow \quad \textit{Step 3 above} \\
= & \mathbb{E}\left[r(s_0,a_0) + V^\star(s_1) \mid \pi\right] \quad \Longleftarrow \quad \textit{Definition of } \pi \textit{ above}
\end{aligned}
\tag{8}
$$

5. $V^\star(s_0) \leq \mathbb{E}\left[r(s_0,a_0) + V^\star(s_1) \mid \pi\right] \leq \mathbb{E}\left[r(s_0,a_0) + \gamma r(s_1,a_1) + \gamma^2 V^\star(s_1) \mid \pi\right] \leq \cdots \leq V^\pi(s_0)$, *so* $\mathbf{V}^\pi = \mathbf{V}^\star$, *i.e., the proposed* π *is optimal.*

□

Policy improvement theorem (proof)

Theorem (Policy Improvement)

If a (deterministic) policy π' satisfies that,

$$Q^\pi(s, \pi'(s)) \geq V^\pi(s) \quad \forall\ s \in \mathcal{S}, \tag{9}$$

then $V^{\pi'}(s) \geq V^\pi(s)$ for any $s \in \mathcal{S}$.

Proof.

Follow the property, for any $s \in \mathcal{S}$, (denote $s' \sim P(\cdot|s, \pi'(s))$ as $s' \sim \pi'$)

$$\begin{aligned} V^\pi(s) \leq\ Q^\pi(s, \pi'(s)) &= \mathbb{E}_{\pi'}\left[r(s_0, \pi'(s_0)) + \gamma V^\pi(s_1) |\, s_0 = s\right] \\ &\leq \mathbb{E}_{\pi'}\left[r_0 + \gamma Q^\pi(s_1, \pi'(s_1)) \,|\, s_0 = s\right] \\ &\leq \mathbb{E}_{\pi'}[r_0 + \gamma r_1 + \gamma V^\pi(s_1) \,|\, s_0 = s] \\ &\leq \cdots \\ &\leq \mathbb{E}_{\pi'}\left[r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots \,|\, s_0 = s\right] = V^{\pi'}(s) \end{aligned} \tag{10}$$

□

POMDPs

Partial observable Markov decision processes (POMDPs)

- $\mathcal{S}$ is the set of all possible states
- $\mathcal{A}$ is the set of all possible actions
- $\mathsf{P}(s'|s,a)$: $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition model
- Ω is the set of observations: $o \in \Omega$.
- O is a set of conditional observation probabilities: $\mathsf{O}(o|s',a)$.
- $r(s,a)$: $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function
- μ is the initial state distribution: $s_0 \sim \mu \in \Delta(\mathcal{S})$
- γ is the discount factor: $\gamma \in [0,1]$

MDP vs POMDP:

- POMDPs are flexible: *We do not have to have perfect information about the states.*
- POMDPs are closer to the real world.

  Example: see a baby crying but do not know the true state (hungry, sleepy, etc).
- MDPs assume perfect knowledge of the states.

POMDPs

◦ When we do not observe the actual states, we construct the so-called *belief states* vector.

Definition (Belief states)

A belief state vector b_t is a distribution over states at time t that estimates the state distribution given the observation and the action history $h_t = \{o_0, a_0, \ldots, a_{t-1}, o_t\}$, i.e., $P(s_t = s|h_t)$:

$$b_t(s) := P(s_t = s|h_t).$$

Remarks:

◦ Via the Bayes rule, the belief states must satisfy:

$$\begin{aligned}
\mathsf{P}(s_t = s|h_t) &= \frac{\mathsf{O}(o_t|s_t, a_{t-1}, h_{t-1})\mathsf{P}(s_t|a_{t-1}, h_{t-1})}{\mathsf{P}(o_t|a_{t-1}, h_{t-1})} \\
&= \frac{\mathsf{O}(o_t|s_t, a_{t-1}, h_{t-1}) \sum_{s_{t-1}} \mathsf{P}(s_t|s_{t-1}, a_{t-1})\mathsf{P}(s_{t-1}|h_{t-1})}{\sum_{s_t} \mathsf{O}(o_t|s_t, a_{t-1}, h_{t-1}) \sum_{s_{t-1}} \mathsf{P}(s_t|s_{t-1}, a_{t-1})\mathsf{P}(s_{t-1}|h_{t-1})}
\end{aligned}$$

◦ As a result, we have a recursion for the conditional probability $\mathsf{P}(s_t = s|h_t)$.

◦ We will represent this recursion via a "belief operator."

The belief operator

- We can concisely represent the recursion on $b_t(s)$ using the belief operator $U : \Delta(\mathcal{S}) \times \Omega \times \mathcal{A} \to \Delta(\mathcal{S})$:

$$b_{t+1}(s') = U(b_t; a, o)(s') = \frac{\mathsf{O}(o|s', a) \sum_{s \in S} \mathsf{P}(s'|s, a) b_t(s)}{\sum_{s'} \mathsf{O}(o|s', a) \sum_{s \in S} \mathsf{P}(s'|s, a) b_t(s)}$$

Remarks:

- The expected (non-stationary) **reward** now also depends on our current belief state:

$$r_t(a) = \sum_{s \in S} r(a, s) b_t(s).$$

- We will focus more on MDPs and how to solve them optimally.
- Tools for MDPs translate readily to POMDPs once we have an estimate of $b_t(s)$.

Numerical example: Hex World

- Traverse a tile map to reach a goal state
- Each cell in the tile map represents a state; action is a move in any of the 6 directions
- Taking any action in certain cells gives a specified reward and transports to a **terminal state**

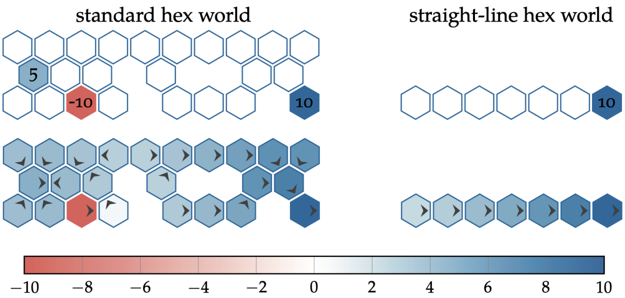


Figure: Top row shows the base problem setup and colors hexes with terminal rewards. Bottom row shows an optimal policy for each problem and colors the expected value. Arrows indicate the action to take in each state. [8]

Numerical example: Value iteration

◦ Initialized with the **east-moving** policy

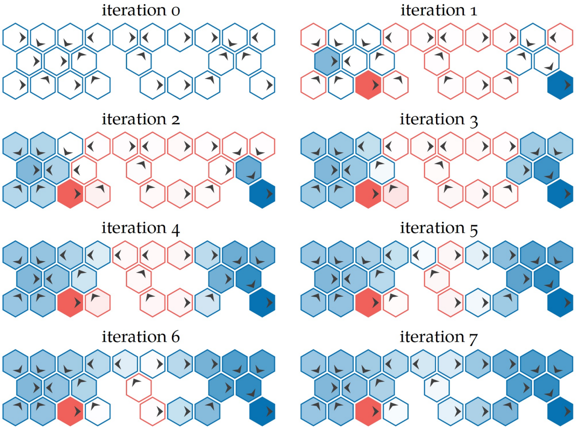


Figure: Value iteration for Hex World. [8]

Numerical example: Policy iteration

- Initialized with the **east-moving** policy
- An optimal policy is obtained (the algorithm converges) in four iterations

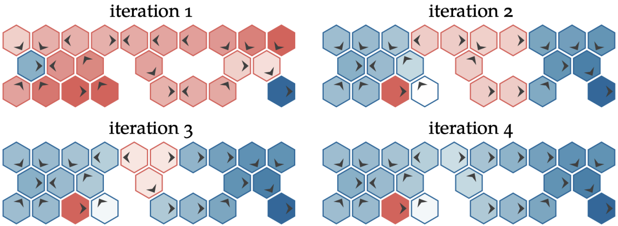


Figure: Policy iteration for Hex World. [8]

References I

[1] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun.
Reinforcement learning: Theory and algorithms.
CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep, 2019.
32

[2] T. Cormen, C. Leiserson, R. Rivest, C. Stein, et al.
Introduction to algorithms, volume 2.
MIT press Cambridge, 2001.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative Adversarial Networks.
ArXiv e-prints, June 2014.
67

[4] Caglar Gulcehre, Sergio Gómez Colmenarejo, Jakub Sygnowski, Thomas Paine, Konrad Zolna, Yutian Chen, Matthew Hoffman, Razvan Pascanu, Nando de Freitas, et al.
Addressing extrapolation error in deep offline reinforcement learning.
2020.
64

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
pages 770–778, 2016.
67

References II

[6] Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.
67

[7] Arthur Jacot, Franck Gabriel, and Clément Hongler.
Neural tangent kernel: Convergence and generalization in neural networks.
In *Advances in neural information processing systems*, pages 8571–8580, 2018.
67

[8] Mykel J. Kochenderfer, Tim A. Wheeler, and Kyle H. Wray.
Algorithms for Decision Making.
MIT press, 2022.
63, 90, 91, 92

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.
67

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton.
Deep learning.
Nature, 521(7553):436–444, 2015.
67

References III

[11] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu.
Offline reinforcement learning: Tutorial, review, and perspectives on open problems.
arXiv preprint arXiv:2005.01643, 2020.
65

[12] Martin L Puterman.
Markov decision processes: discrete stochastic dynamic programming.
John Wiley & Sons, 2014.
32

[13] Satinder Singh Richard and Richard C. Yee.
An upper bound on the loss from approximate optimal-value functions.
In *Machine Learning*, pages 227–233, 1994.
52

[14] Bruno Scherrer.
Improved and generalized upper bounds on the complexity of policy iteration.
Mathematics of Operations Research, 41(3):758–774, 2016.
52, 53, 54, 55

[15] Bruno Scherrer, Victor Gabillon, Mohammad Ghavamzadeh, and Matthieu Geist.
Approximate modified policy iteration.
arXiv preprint arXiv:1205.3054, 2012.
52

References IV

[16] Bernhard Schölkopf and Alexander J. Smola.
Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.
MIT Press, Cambridge, MA, USA, 2001.
67

[17] Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári.
Convergence results for single-step on-policy reinforcement-learning algorithms.
Machine learning, 38(3):287–308, 2000.
81

[18] Richard S. Sutton and A. G. Barto.
Reinforcement Learning: An Introduction.
MIT Press, Cambridge, MA, 1998.
73, 78

[19] Richard S Sutton and Andrew G Barto.
Reinforcement learning: An introduction.
MIT press, 2018.
6, 76, 79, 82

[20] V.B. Tadic.
On the almost sure rate of convergence of linear stochastic approximation algorithms.
IEEE Transactions on Information Theory, 50(2):401–409, 2004.
82

References V

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin.
Attention is all you need.
2017.
67