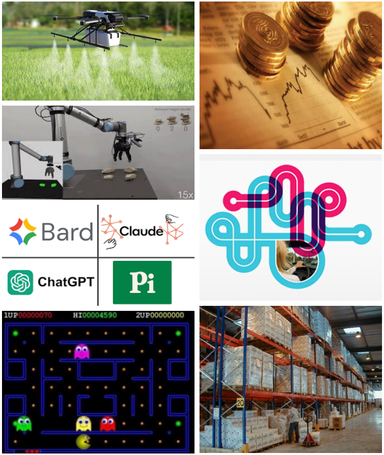
Theory of Discounted Markov Decision Processes

Mohammad Sadegh Talebi
m.shahi@di.ku.dk
Department of Computer Science

Sequential Decision Making

Many tasks in real life are **online sequential decision-making** tasks that fall in the framework of **reinforcement learning**:



- Selling or buying an asset
- Inventory management
- Portfolio optimization
- Robotics
- Playing computer games
- Routing in networks /
- Precision Agriculture and Farming
- LLMs

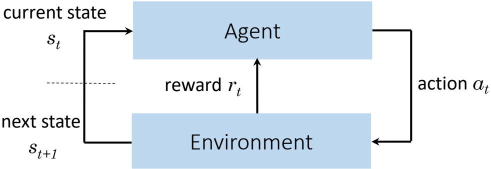
Sequential Decision Making: General Setting

Consider a discrete time system.

Minimal ingredients of decision making:

- A notion of **state** capturing different situations
- **Actions** capturing options available at any situation
- A **reward signal** indicating the quality of the action taken

Goal: To maximize an objective function, often defined in terms of rewards.



Sequential Decision Making: General Setting

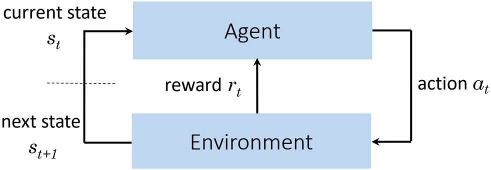
At each step $t = 1, 2, \ldots, N$, an agent interacts with an unknown environment

- observes state s_t,
- chooses an action a_t from a given action set, using a control policy φ

$$a_t = \varphi(s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}),$$

- receives (random) reward r_t.

- **Goal:** To maximize $f(r_1, \ldots, r_N)$.
  - E.g., $f(r_1, \ldots, r_N) = \sum_{t=1}^N r_t$, $f(r_1, \ldots, r_N) = \sum_{t=1}^T \log(1 + r_t)$
- Observations and rewards are generated by an **uncertain** and (potentially) **unknown** environment.



Markov Decision Processes

Markov Decision Process

A Markov Decision Process (MDP) is a tuple $M = (\mathcal{S}, \mathcal{A}, P, R)$:

- State-space $\mathcal{S}$ (finite, countably infinite, or continuous)
- Action-space $\mathcal{A} = \cup_{s\in\mathcal{S}}\mathcal{A}_s$ (finite, countably infinite, or continuous)
  - $\mathcal{A}_s$ is the set of actions available in state s
- Transition function P: Selecting $a \in \mathcal{A}_s$ in $s \in \mathcal{S}$ leads to a transition to s' with probability $P(s'|s,a)$. $P(\cdot|s,a)$ is a probability distribution over $\mathcal{S}$, i.e.,

$$\sum_{s'\in\mathcal{S}} P(s'|s,a) = 1$$

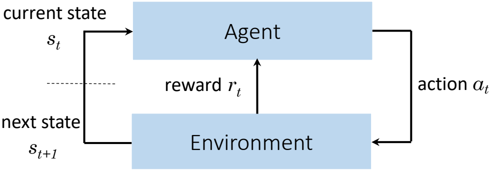
- Reward function R: Selecting $a \in \mathcal{A}_s$ in $s \in \mathcal{S}$ yields a reward $r \sim R(s,a)$.

Interaction with MDP

An **agent** interacts with the MDP for N rounds.

At each time step t:

- The agent observes the current state s_t and takes an action $a_t \in \mathcal{A}_{s_t}$
- The environment (MDP) decides a reward $r_t := r(s_t, a_t) \sim R(s_t, a_t)$ and a next state $s_{t+1} \sim P(\cdot|s_t, a_t)$
- The agent receives r_t (any time in step t before start of $t+1$)



This interaction produces a trajectory (or history)

$$h_t = (s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$$

Markov Property

MDPs adhere to the Markov property.

- At each time t, s_{t+1} and r_t only depend on s_t and a_t.
- More precisely,

$$\mathbb{P}\Big(s_{t+1} = s' \Big| s_1, a_1, \dots, s_{t-1}, a_{t-1}, s_t, a_t\Big) = \underbrace{\mathbb{P}\Big(s_{t+1} = s' \Big| s_t, a_t\Big)}_{=P(s'|s_t, a_t)}$$

$$R\big(s_1, a_1, \dots, s_{t-1}, a_{t-1}, s_t, a_t\big) = R(s_t, a_t)$$

Classification of MDPs based on Horizon N

- **Finite-Horizon MDPs**: $N < \infty$, and the goal is to solve
$$\max_{\text{all strategies}} \mathbb{E}\Big[\sum_{t=1}^{N-1} r(s_t, a_t) + r(s_N)\Big]$$
- **Infinite-Horizon Discounted MDPs:** $N = \infty$, and given discount factor $\gamma \in (0, 1)$, the goal is to solve
$$\max_{\text{all strategies}} \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t)\Big]$$
- **Infinite-Horizon Undiscounted MDPs (Average-Reward MDPs):** $N = \infty$, and the goal is to solve
$$\max_{\text{all strategies}} \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\Big[\sum_{t=1}^{N} r(s_t, a_t)\Big]$$

This lecture: Infinite-Horizon Discounted MDPs.

Reward Function: Some Comments

Two Reward Models:

- $R(s,a)$: Reward distribution in state s when executing action a
- $R(s,a,s')$: Reward distribution in state s when executing action a and the next state is s'

We consider the first model, but the two models are related:

$$R(s,a) = \sum_{s' \in \mathcal{S}} R(s,a,s')P(s'|s,a)$$

Reward Function: Some Comments

- **Bounded Rewards Assumption**: We assume

$$R_{\max} := \sup_{s,a} \left| \mathbb{E}_{r\sim R(s,a)}[r] \right| < \infty$$

- For simplicity, we assume *deterministic rewards*
  - Hence, $r \sim R(s,a)$ means $r = R(s,a)$.
  - Hence, we may use $r(s,a)$ and $R(s,a)$ interchangeably, but tend to keep $r(s,a)$ for generality.
  - The results in this lecture will hold for stochastic rewards under mild assumptions (and often by replacing $R(s,a)$ or $r(s,a)$ with its mean).

This lecture: We consider deterministic and bounded rewards.

Infinite Horizon Discounted MDPs

Infinite-Horizon Discounted MDPs: $N = \infty$, and the goal is to maximize the total expected sum of discounted rewards

$$\max_{\text{all strategies}} \mathbb{E}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t)\Big]$$

Two views on discounting with a discount factor $\gamma \in [0, 1)$:

- Earlier rewards are more important. A unit reward at present will worth γ in the next slot.
- Problems with random horizon N and absorbing states

Example 1

Suppose we receive an order for a given product with probability α. We can either process all the unfilled orders or process no order.

- The cost per unfilled order per period is $c > 0$, and the setup cost to process unfilled order is $K > 0$.
- Assume that the total number of orders that can remain unfilled is n.
- Assume there is a discount factor $\gamma < 1$.

The goal is to find an order processing strategy that has minimal expected cost.

Example 1

Modeling as a discounted MDP:

- **State Space:** Define the state as the number of unfilled orders at the beginning of each period $\Longrightarrow \mathcal{S} = \{0, 1, \dots, n\}$.
- **Action Space:** For $s \neq 0, n$, we have $\mathcal{A}_s = \{J, \overline{J}\}$, where J =processing unfilled orders and $\overline{J}$ = processing no order $\Longrightarrow \mathcal{A}_0 = \{\overline{J}\}$ and $\mathcal{A}_n = \{J\}$.
- **Reward Function:**

$$R(i, J) = -K, \quad R(i, \overline{J}) = -ci, \quad i = 1, \dots, n-1,$$
$$R(0, \overline{J}) = 0, \quad R(n, J) = -K.$$

- **Transition Function:**

$$P(0|i, J) = 1 - \alpha, \quad P(1|i, J) = \alpha, \quad i = 1, 2, \dots, n-1,$$
$$P(i|i, \overline{J}) = 1 - \alpha, \quad P(i+1|i, \overline{J}) = \alpha, \quad i = 1, 2, \dots, n-1,$$
$$P(0|n, J) = 1 - \alpha, \quad P(1|n, J) = \alpha,$$
$$P(0|0, \overline{J}) = 1 - \alpha, \quad P(1|0, \overline{J}) = \alpha.$$

Example 2

A job seeker receives a job offer at each time period, which she may accept or reject. The offered policy takes one of n possible values $w_1, \dots, w_n$ with given probabilities independently of preceding offers. Let q_i be the probability of an offer at salary level w_i at any one period.

- If she accepts the offer, she must keep the job for the rest of her life.
- If she rejects the offer, she receives unemployment compensation c for the current period and is eligible to accept future offers.
- Assume that income is discounted by a factor $\gamma < 1$.

The job seeker is interested in a strategy maximizing her income.

Example 2

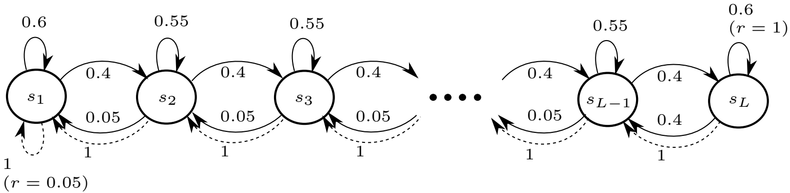
Modeling as a discounted MDP:

- **State Space:** $\mathcal{S} = \{s_1, s_2, \dots, s_n, s'_1, \dots, s'_n\}$, where for each i, s_i corresponds to the case where the job seeker is unemployed and being offered a salary w_i, and s'_i corresponds to the case where she is employed at a salary level w_i.
- **Action Space:** $\mathcal{A}_{s_i} = \{C, \overline{C}\}$ for all i, where C denotes the action corresponding to accepting an offer ($\overline{C}$ rejecting the offer). Furthermore, $\mathcal{A}_{s'_i} = \{X\}$ for all i, where X indicates continuation of the job.
- **Reward Function:** For all i, $R(s_i, C) = w_i$, $R(s_i, \overline{C}) = c$, and $R(s'_i, X) = w_i$.
- **Transition Function:**

$$P(s'_i|s'_i, X) = 1, \quad P(s'_i|s_i, C) = 1, \quad P(s_j|s_i, \overline{C}) = q_j, \quad i = 1, \dots, n.$$

Example 3

The L-state RiverSwim MDP



Exercise: Determine

- **State Space:**
- **Action Space:**
- **Reward Function:**
- **Transition Function:**

Policy and Value Function

Policy

When interacting with an MDP, actions are taken according to some **policy**:

Classification of policies:

- deterministic vs. randomized (stochastic)
- stationary vs. history-dependent

	deterministic	randomized
stationary	$\pi : \mathcal{S} \to \mathcal{A}$	$\pi : \mathcal{S} \to \Delta(\mathcal{A})$
history-dependent	$\pi : \mathcal{H} \to \mathcal{A}$	$\pi : \mathcal{H} \to \Delta(\mathcal{A})$

- $\Delta(\mathcal{A})$ denotes the simplex of probability distributions over $\mathcal{A}$.
- $\mathcal{H}$ the set of all possible histories (trajectories).

Policy

	deterministic	randomized
stationary	$\pi : \mathcal{S} \to \mathcal{A}$	$\pi : \mathcal{S} \to \Delta(\mathcal{A})$
history-dependent	$\pi : \mathcal{H} \to \mathcal{A}$	$\pi : \mathcal{H} \to \Delta(\mathcal{A})$

- Π^{SD}: The set of stationary deterministic policies
- Π^{SR}: The set of stationary randomized policies
- Π^{HD}: The set of history-dependent deterministic policies
- Π^{HR}: The set of history-dependent randomized policies

(i) $\Pi^{\mathsf{SD}} \subset \Pi^{\mathsf{SR}} \subset \Pi^{\mathsf{HR}}$ (ii) $\Pi^{\mathsf{SD}} \subset \Pi^{\mathsf{HD}} \subset \Pi^{\mathsf{HR}}$

Notation:

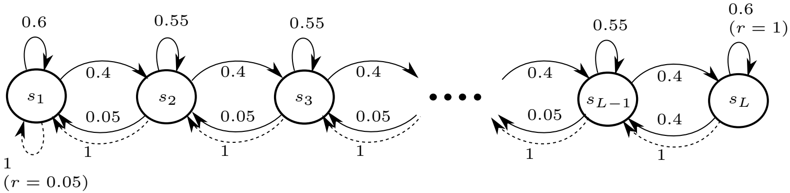
- For $\pi \in \Pi^{\mathsf{SR}}$, we write $a \sim \pi(\cdot|s)$. Also, given $f : \mathcal{A}_s \to \mathbb{R}$,

$$\mathbb{E}_{a\sim\pi(s)}[f(a)] = \sum_{a\in\mathcal{A}_s} f(a)\pi(a|s)$$

- For $\pi \in \Pi^{\mathsf{HR}}$, we write $a \sim \pi(\cdot|h)$.

Policy: Examples

The L-state RiverSwim MDP



Examples:

- π_1 : always go right. $(\pi_1 \in \Pi^{\text{SD}})$
- π_2 : go right w.p. 0.7 and left w.p. 0.3. $(\pi_2 \in \Pi^{\text{SR}})$
- π_3 : go right if $s_t \neq s_{t-1}$, otherwise go left . $(\pi_3 \in \Pi^{\text{HD}})$

Induced Markov Chains

- Every $\pi \in \Pi^{\mathsf{SR}}$ induces a Markov chain on M, with transition probability matrix P^π given by:

$$P^\pi_{s,s'} = \sum_{a \in \mathcal{A}_s} P(s'|s,a)\pi(a|s)\,, \quad s, s' \in \mathcal{S}.$$

- Every $\pi \in \Pi^{\mathsf{SR}}$ induces a reward vector $r^\pi \in \mathbb{R}^S$ on M defined by:

$$r^\pi(s) = \sum_{a \in \mathcal{A}_s} R(s,a)\pi(a|s)\,, \quad s \in \mathcal{S}.$$

- If $\pi \in \Pi^{\mathsf{SD}}$, then $P^\pi_{s,s'} = P(s'|s,\pi(s))$ and $r^\pi(s) = R(s,\pi(s))$.

Every policy $\pi \in \Pi^{\mathsf{SR}}$ induces a **Markov Reward Process (MRP)** on M, specified by r^π and P^π.

Value Function

The value function of policy π (or simply, value of π) is a mapping $V^\pi : \mathcal{S} \to \mathbb{R}$ defined as

$$V^\pi(s) := \mathbb{E}^\pi\left[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t, a_t) \middle| s_1 = s\right].$$

where $\mathbb{E}^\pi$ indicates expectation over trajectories generated by π.

- Intuitively, $V^\pi(s)$ measures the sum of future discounted rewards (in expectation) when the agent starts in s and follows π.
- We have

$$|V^\pi(s)| \leq \frac{R_{\max}}{1-\gamma}, \quad \forall s \in \mathcal{S}$$

Action-Value Function

The action-value function of policy π (or simply, Q-value of π) is a mapping $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ defined as (Under the bounded reward assumption)

$$Q^\pi(s,a) := \mathbb{E}^\pi\Big[\sum_{t=1}^{\infty} \gamma^{t-1} r(s_t,a_t)\Big| s_1 = s, a_1 = a\Big] .$$

- Intuitively, $Q^\pi(s,a)$ measures the sum of future discounted rewards (in expectation) when the agent starts in s and takes action a in the first step (possibly $a \neq \pi(s)$), and then follows π afterwards.
- We have
$$|Q^\pi(s,a)| \le \frac{R_{\max}}{1-\gamma} , \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$$
- For all $s \in \mathcal{S}$, $Q^\pi(s,\pi(s)) = V^\pi(s)$.

Policy Evaluation

Bellman Equation for π

Theorem (Bellman Equation for π)

Let $\pi \in \Pi^{\textsf{SR}}$. For all $s \in \mathcal{S}$,

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{a\sim\pi(s)}[r(s,a)] + \gamma\mathbb{E}_{a\sim\pi(s)}\Big[\sum\nolimits_{x\in\mathcal{S}} P(x|s,a)V^\pi(x)\Big] \\ &= \sum\nolimits_{a\in\mathcal{A}_s} \pi(a|s) r(s,a) + \gamma \sum\nolimits_{a\in\mathcal{A}_s} \pi(a|s) \sum\nolimits_{x\in\mathcal{S}} P(x|s,a)V^\pi(x) \end{aligned}$$

Equivalently, $V^\pi = r^\pi + \gamma P^\pi V^\pi$.

- These relations are called the Bellman equation.
- The theorem tells us that for $\pi \in \Pi^{\textsf{SR}}$, V^π satisfies the Bellman equation.
- For a deterministic policy $\pi \in \Pi^{\textsf{SD}}$, the Bellman equation becomes:

$$V^\pi(s) = r(s,\pi(s)) + \gamma \sum_{x\in\mathcal{S}} P(x|s,\pi(s))V^\pi(x)\,, \quad s \in \mathcal{S}.$$

Bellman Operator for π

The Bellman operator associated to $\pi \in \Pi^{\mathsf{SR}}$ is a mapping $\mathcal{T}^\pi : \mathbb{R}^S \to \mathbb{R}^S$, such that for any function $f : \mathcal{S} \to \mathbb{R}$,

$$\mathcal{T}^\pi f := r^\pi + \gamma P^\pi f \,.$$

- Intuitively, $\mathcal{T}^\pi$ is the value of π for the same one-stage problem.
- $\mathcal{T}^\pi$ applies to (or *operates on*) a function defined on $\mathcal{S}$ and returns another function defined on $\mathcal{S}$.
- The Bellman equation $V^\pi = r^\pi + \gamma P^\pi V^\pi$ reads

$$V^\pi = \mathcal{T}^\pi V^\pi$$

In other words, V^π is the *unique* fixed-point of the operator $\mathcal{T}^\pi$.

Bellman Equation for π

We prove the theorem for $\pi \in \Pi^{\mathsf{SD}}$. (See Lecture Notes for $\pi \in \Pi^{\mathsf{SR}}$.)

Proof. Let $\pi \in \Pi^{\mathsf{SD}}$ and $s \in \mathcal{S}$. We have

$$\begin{aligned}
V^\pi(s) &= \mathbb{E}^\pi\Big[\sum_{t=1}^{\infty}\gamma^{t-1}r(s_t,\pi(s_t))\Big|s_1=s\Big] \\
&= r(s,\pi(s)) + \mathbb{E}^\pi\Big[\sum_{t=2}^{\infty}\gamma^{t-1}r(s_t,\pi(s_t))\Big|s_1=s\Big] \\
&= r(s,\pi(s)) + \gamma\sum_{x\in\mathcal{S}}\mathbb{P}(s_2=x|s_1=s,a_1=\pi(s_1))\,\underbrace{\mathbb{E}^\pi\Big[\sum_{t=2}^{\infty}\gamma^{t-2}r(s_t,\pi(s_t))\Big|s_2=x\Big]}_{=V^\pi(x)} \\
&= r(s,\pi(s)) + \gamma\sum_{x\in\mathcal{S}}\mathbb{P}(s_2=x|s_1=s,a_1=\pi(s_1))V^\pi(x) \\
&= r(s,\pi(s)) + \gamma\sum_{x\in\mathcal{S}}P(x|s,\pi(s))V^\pi(x)\,.
\end{aligned}$$

Bellman Equation for π

The notion of the Bellman operator can be extended to Q-value functions.

The Bellman operator for Q-value of π is a mapping $\mathcal{T}^\pi : \mathbb{R}^{S\times A} \to \mathbb{R}^{S\times A}$ defined as follows: For any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$,

$$(\mathcal{T}^\pi f)(s,a) = r(s,a) + \gamma \mathbb{E}_{a' \sim \pi(s)}\Big[\sum_y P(y|s,a')f(y,a')\Big], \qquad (s,a) \in \mathcal{S} \times \mathcal{A}$$

- Hence, we have $Q^\pi = \mathcal{T}^\pi Q^\pi$
- In other words, Q^π is the fixed point of the operator $\mathcal{T}^\pi$ (for Q-function).

Policy Evaluation

Policy Evaluation: Computing V^π for a given π

- **Direct Computation:** Using Bellman equation,

$$V^\pi = r^\pi + \gamma P^\pi V^\pi \quad \Longrightarrow_{I-\gamma P^\pi \text{ is invertible}} \quad V^\pi = (I - \gamma P^\pi)^{-1} r^\pi$$

- **Iterative Policy Evaluation:** Using $V^\pi = \mathcal{T}^\pi V^\pi$, the sequence

$$V_{n+1} = \mathcal{T}^\pi V_n = \underbrace{\mathcal{T}^\pi \cdots \mathcal{T}^\pi}_{n+1 \text{ times}} V_0$$

  converges to V^π starting from any V_0.

- **Monte-Carlo Method:** Generate a number of trajectories of π and use the sample mean as an estimator to V^π.

So far:

- We defined policies and the value function.
- We characterized the value of stationary policies (via Bellman equations and operator).
- We developed ways to compute the value of a *fixed* stationary policy.

How to find an optimal strategy/policy? Alternatively, how to find policies with good values?

Optimization in Discounted MDPs: Optimal Policy and Value

Optimal Value and Policy

Solving a discounted MDP M amounts to solving the following optimization problem:

$$V^\star(s) = \sup_{\pi \in \Pi^{\mathsf{HR}}} V^\pi(s)\,, \qquad \forall s \in \mathcal{S}.$$

(i) $V^\star : \mathcal{S} \to \mathbb{R}$ is called the optimal value function.

(ii) If there exists $\pi^\star$ such that $V^{\pi^\star}(s) = V^\star(s)$ for all $s \in \mathcal{S}$, then $\pi^\star$ is called an optimal policy.

(iii) π is ε-optimal for $\varepsilon > 0$ if

$$V^\pi(s) \geq V^\star(s) - \varepsilon, \quad \forall s \in \mathcal{S}$$

Bellman Optimality Equation

Theorem

$V^\star$ satisfies the optimal Bellman equation:

$$V^\star(s) = \max_{a\in\mathcal{A}_s}\Big(r(s,a) + \gamma\sum_{x\in\mathcal{S}} P(x|s,a)V^\star(x)\Big), \quad s\in\mathcal{S}$$

The optimal Bellman operator is a mapping $\mathcal{T}:\mathbb{R}^S\to\mathbb{R}^S$, such that for any function $f:\mathcal{S}\to\mathbb{R}$,

$$(\mathcal{T}f)(s) := \max_{a\in\mathcal{A}_s}\Big(r(s,a) + \gamma\sum_{x\in\mathcal{S}} P(x|s,a)f(x)\Big), \quad s\in\mathcal{S}$$

- $V^\star$ satisfies $\mathcal{T}V^\star = V^\star$.
- We can define $\mathcal{T}$ and optimal Bellman equation for the optimal Q function.

Optimality Theorems

Theorem

Suppose the state space $\mathcal{S}$ is finite. Then there exists a policy $\pi^\star \in \Pi^{SD}$.

- Thus, when seeking $\pi^\star$ in a discounted MDP with a finite $\mathcal{S}$, we can restrict our attention to Π^{SD}.
- In other words, for finite $\mathcal{S}$,

$$\sup_{\pi\in\Pi^{\mathsf{HR}}} V^\pi = \sup_{\pi\in\Pi^{\mathsf{SD}}} V^\pi = \max_{\pi\in\Pi^{\mathsf{SD}}} V^\pi$$

Optimality Theorems

A fundamental result in the theory of discounted MDPs:

Theorem

A stationary deterministic policy π is optimal if and only if

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

Equivalently, π is optimal if and only if it attains the maximum in the Bellman optimality equations: For all $s \in \mathcal{S}$,

$$\pi(s) \in \arg\max_{a \in \mathcal{A}_s} \Big(r(s,a) + \sum_{x \in \mathcal{S}} P(x|s,a) V^{\star}(x) \Big) .$$

So far:

- We defined policies and the value function.
- We characterized the value of stationary policies (via Bellman equations and operator).
- We developed ways to compute the value of a *fixed* stationary policy.
- We defined the notion of optimality and showed that there exists $\pi^\star \in \Pi^{\mathsf{SD}}$ when $\mathcal{S}$ is finite.
- We characterized the optimal value function $V^\star$ (via optimal Bellman equation).

How to actually compute $\pi^\star$?

Algorithms for Solving Discounted MDPs

Major Solution Methods

Three major classes of algorithms for solving discounted MDPs:

- Value Iteration
- Policy Iteration
- Linear Programming

Contraction Mapping

An operator (or mapping) $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^n$ is called a *κ-contraction mapping* (with respect to $\| \cdot \|$) if there exists $\kappa \in [0, 1)$ such that for all $v, v' \in \mathbb{R}^n$,

$$\|\mathcal{L}v - \mathcal{L}v'\| \leq \kappa \|v - v'\| \, .$$

Theorem (Banach Fixed-Point Theorem)

Suppose $\mathcal{L}$ is a contraction mapping. Then

(i) *there exists a unique $v^\star \in \mathbb{R}^n$ such that $\mathcal{L}v^\star = v^\star$;*

(ii) *for any $v_0 \in \mathbb{R}^n$, the sequence $(v_n)_{n \geq 0}$ with $v_{n+1} = \mathcal{L}v_n = \mathcal{L}^{n+1}v_0$ for $n \geq 0$ converges to $v^\star$.*

$\mathcal{T}^\pi$ and $\mathcal{T}$ Are Contraction Mapping

Lemma

For any $v, v' \in \mathbb{R}^S$, *and any* π,

$$\|\mathcal{T}^\pi v - \mathcal{T}^\pi v'\|_\infty \le \gamma \|v - v'\|_\infty \,,$$
$$\|\mathcal{T} v - \mathcal{T} v'\|_\infty \le \gamma \|v - v'\|_\infty \,.$$

Hence, $\mathcal{T}^\pi$ *and* $\mathcal{T}$ *are* γ*-contraction mappings w.r.t.* $\|\cdot\|_\infty$.

Proof. First statement is easy to prove. For the second, we have:

$$\begin{aligned}
&\|\mathcal{T} v - \mathcal{T} v'\|_\infty \\
&\qquad = \max_s \Big| \max_{a\in\mathcal{A}_s} \Big(r(s,a) + \gamma \sum_j P(j|s,a) v(j) \Big) - \max_{a\in\mathcal{A}_s} \Big(r(s,a) + \gamma \sum_j P(j|s,a) v'(j) \Big) \Big| \\
&\qquad \le \max_s \max_{a\in\mathcal{A}_s} \Big| \gamma \sum_j P(j|s,a)(v(j) - v'(j)) \Big|
\end{aligned}$$

(Using inequality $|\max_x f(x) - \max_x g(x)| \le \max_x |f(x) - g(x)|$)

$$\le \gamma \max_s \max_{a\in\mathcal{A}_s} \max_j |v(j) - v'(j)| \sum_j P(j|s,a) = \gamma \|v - v'\|_\infty$$

Value Iteration

Value Iteration (`VI`)

- The most well-known, and perhaps the simplest, algorithm for solving discounted MDPs
- Around since the early days of MDPs
- Also known as successive approximation, backward induction, etc.

Idea: The optimal Bellman operator $\mathcal{T}$ is contracting. Iterate $\mathcal{T}$ until convergence:

$$V_{n+1} = \mathcal{T} V_n, \quad n = 0, 1, 2, \dots$$

Indeed, `VI` is an algorithm for approximating the fixed point of $\mathcal{T}$.

Value Iteration (VI)

input: ε

- **initialization:** Select a value function $V_0 \in \mathbb{R}^S$, $V_1 = R_{\max}/(1-\gamma)\mathbf{1}$, and set $n = 0$
- **while** $\left(\|V_{n+1} - V_n\|_\infty \geq \frac{\varepsilon(1-\gamma)}{2\gamma}\right)$
  - (i) Update, for each $s \in \mathcal{S}$,

$$V_{n+1}(s) = \max_{a \in \mathcal{A}_s} \left(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_n(x) \right)$$

  - (ii) Increment n.

output:

$$\pi^{\mathtt{VI}}(s) \in \arg\max_{a \in \mathcal{A}_s} \left(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a) V_n(x) \right), \quad s \in \mathcal{S}$$

VI: Convergence

VI is a *globally convergent* method for finding an ε-optimal policy. Formally:

Theorem

Let $(V_n)_{n\geq 0}$ a sequence of value functions generated by VI with some $\varepsilon > 0$ starting from an arbitrary initial point $V_0 \in \mathbb{R}^S$. Then,

(i) *V_n converges to $V^\star$ in norm;*

(ii) *the algorithm stops after finitely many iterations;*

(iii) *π^{VI} is ε-optimal;*

(iv) *when convergence criterion is satisfied, $\|V_{n+1} - V^\star\|_\infty < \varepsilon/2$.*

Each iteration of VI involves $O(S^2A)$ arithmetic calculations. The iteration complexity of VI depends on both ε and γ. The larger the γ, the more iteration until the algorithm finds an ε-optimal policy.

Policy Iteration

Policy Iteration (PI)

- A popular algorithm for solving discounted MDPs
- Around since early days of MDPs
- Like VI, it is an iterative algorithm but directly searches in the space of policies.

Idea: Starting from an initial policy, at each iterate n,

(i) Find V^{π_n} (policy evaluation)

(ii) Improve π_n to π_{n+1} using V^{π_n} (policy improvement)

Policy Iteration (PI)

- **initialization:** Select π_0 and π_1 arbitrarily ($\pi_0 \neq \pi_1$), and set $n = 0$
- **while**$\big(\pi_{n+1} \neq \pi_n\big)$
  - (i) *Policy Evaluation:* Find V_n, the value of π_n by solving
    $$(I - \gamma P^{\pi_n})V_n = r^{\pi_n}$$
  - (ii) *Policy Improvement:* Choose π_{n+1} such that
    $$\pi_{n+1}(s) \in \arg\max_{a \in \mathcal{A}_s} \Big(r(s,a) + \gamma \sum_{x \in \mathcal{S}} P(x|s,a)V_n(x) \Big)$$
    and if possible, set $\pi_{n+1} = \pi_n$.
  - (iii) Increment n.
- **output:** $\pi^{\mathtt{PI}} = \pi_n$

PI: Convergence

Theorem

Suppose M has a finite state-action space. Then,

(i) *PI terminates in at most*

$$O\left(\max\left\{\frac{SA}{1-\gamma}\log\frac{1}{1-\gamma}, \frac{A^S}{S}\right\}\right) \qquad \textit{iterations;}$$

(ii) $\pi^{PI} = \pi^\star$.

The values of successive stationary policies generated by PI are non-decreasing. I.e., $V_{n+1} \geq V_n$ for any n. Further, the number of policies is finite A^S.

- Each iteration in PI involves solving a linear system with S equations and S unknowns. Hence, per iteration complexity of PI is $O(S^3 + S^2 A)$.
- In practice, PI converges within, at most, a few tens of iterations.