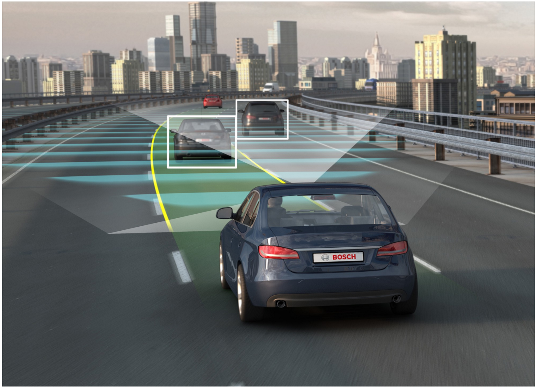
# Markov Decision Processes

**Insoon Yang**

Department of Electrical and Computer Engineering
Seoul National University

# Systems under Uncertainty I: Self-Driving Cars

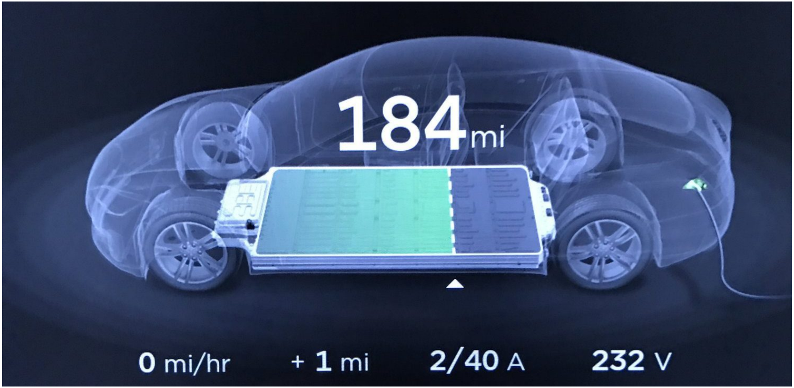



- Uncertainty from weather (rain, sun), sensors, vision, other cars, human (pedestrians, drivers)

# Systems under Uncertainty II: Drones

- Uncertainty from weather (wind), sensors, vision, other drones

# Systems under Uncertainty III: Battery Management Systems




- Uncertainty from electrochemical concentration levels, temperatures, sensors

# Systems under Uncertainty IV: Smart Grids

- Uncertainty from weather (wind, sun), human (demand) sensors, electricity prices

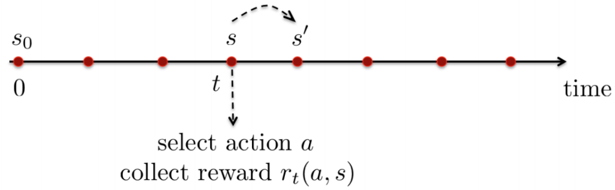
# Systems under Uncertainty V: Financial Markets


- Almost everything is uncertain

# Common Features

- Dynamic: system state changes over time and depends on control action/input
- Unpredictable: we don't know exactly how the system evolves over time
- Power of probability (or data): but we have some information useful in decision-making

# Sequential Decision Making




- Goal: select actions **over time** to maximize the expected cumulative reward
- Actions may have long term effects. *Q) Why?*
- It may be better to sacrifice immediate reward to gain more long-term reward. *Q) How?*

## State

- At each stage (or time), the system occupies a *state*.
- Notation: $s_t$ (state at stage $t$ (or time $t$))
- It quantifies the status of the system.
- Example: position, velocity, temperature, chemical concentration, wealth, population
- $S$: set of states (state space)
  e.g., $S = \{1, \ldots, n\}$ (discrete), $S = \mathbb{R}^n$ (continuous)

# Action

- At each stage, the decision maker observes the system state and choose an *action*.
- Notation: $a_t$ (action at stage $t$)
- It quantifies the adjustable input to the system.
- Example: acceleration, steering, ON/OFF, buy/sell
- $A$: set of actions (action space)
  e.g., $A = \{1, \ldots, m\}$ (discrete), $A = \mathbb{R}^m$ (continuous)
- Actions may be chosen either randomly or deterministically

# Rewards

- As a result of choosing action $a_t$ in state $s_t$ at stage $t$, the decision maker receives a *reward*, $r(s_t, a_t)$.
- Notation: $r : S \times A \to \mathbb{R}$ (reward function)
- It quantifies how well the *immediate* action and state are chosen.
- It does *not* measure the benefits from future actions or states.
- Example: income, score, negative cost

## Transition Probabilities

- If the decision maker chooses action $a_t$ in state $s_t$ at stage $t$, the system state at the next stage is determined by the probability distribution $p(\cdot|s_t, a_t)$, called the *transition probability*.
- It describes how the system evolves over time (modeling stochastic dynamics).
- Notation: $p(s'|s, a) := \mathrm{Prob}(s_{t+1} = s'|s_t = s, a_t = a)$ (transition probability function)
- Example: vehicle dynamics, robot movement, temperature fluctuation, congestion in communication networks, stock markets
- We usually assume that

$$\sum_{s' \in S} p(s'|s, a) = 1 \quad \forall (s, a) \in S \times A.$$

# Decision Rules (s,a)

~ policy(s,a,t)

- A *decision rule* prescribes a procedure for action selection in each state at a specified stage.
- Notation: $\pi_t : S \to A$ ((deterministic Markov) decision rule)
- Markov vs history dependent
- deterministic vs stochastic (randomized)
- A fundamental question in MDP:
  *Under what conditions is it optimal to use a deterministic Markov decision rule at each stage?*

# Policies

finite time vs infinite time ==> optimal decision rule이 매시간마다 다르거나 수렴하거나

- A *policy* or *strategy* specifies the decision rule to be used at all stages.
- Notation: $\pi := (\pi_1, \pi_2, \ldots)$
- A policy is called *stationary* if $\pi_t$'s are identical for all $t$.

  action종류가 동일
- Policy is what we'll optimize
- Often use the term "policy" instead of "decision rule".

# Markov Decision Processes (MDPs)

## Definition

A Markov decision process (MDP) is a tuple $\langle S, A, P, R, \gamma \rangle$, consisting of

- $S$: set of states (state space)
  e.g., $S = \{1, \ldots, n\}$ (discrete), $S = \mathbb{R}^n$ (continuous)
- $A$: set of actions (action space)
  e.g., $A = \{1, \ldots, m\}$ (discrete), $A = \mathbb{R}^m$ (continuous)
- $p$: state transition probability
  $p(s'|s, a) := \mathrm{Prob}(s_{t+1} = s'|s_t = s, a_t = a)$
- $r$: reward function
  $r(s_t, a_t) = r_t$
- $\gamma \in (0, 1)$: discount factor

# Example: Two-State MDP

Setting

- State set: $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$
- Action set: $A_{s_1} = \{\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}\}$, $A_{s_2} = \{\boldsymbol{a}_{2,1}\}$
- Rewards: $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 5$, $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 10$, $r(\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = -1$
- Transition probabilities: $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 0.5$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 1$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 1$

# Example: Two-State MDP

Setting

- State set: $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$
- Action set: $A_{s_1} = \{\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}\}$, $A_{s_2} = \{\boldsymbol{a}_{2,1}\}$
- Rewards: $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 5$, $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 10$, $r(\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = -1$
- Transition probabilities: $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 0.5$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 1$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 1$

Q)

1. Example of deterministic Markov policies?

# Example: Two-State MDP

Setting

- State set: $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$
- Action set: $A_{s_1} = \{\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}\}$, $A_{s_2} = \{\boldsymbol{a}_{2,1}\}$
- Rewards: $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 5$, $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 10$, $r(\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = -1$
- Transition probabilities: $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 0.5$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 1$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 1$

Q)

1. Example of deterministic Markov policies?

$$\pi(\boldsymbol{s}_1) = \boldsymbol{a}_{1,1}, \quad \pi(\boldsymbol{s}_2) = \boldsymbol{a}_{2,1}$$

# Example: Two-State MDP

Setting

- State set: $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$
- Action set: $A_{s_1} = \{\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}\}$, $A_{s_2} = \{\boldsymbol{a}_{2,1}\}$
- Rewards: $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 5$, $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 10$, $r(\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = -1$
- Transition probabilities: $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 0.5$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 1$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 1$

Q)

1. Example of deterministic Markov policies?

   $$\pi(\boldsymbol{s}_1) = \boldsymbol{a}_{1,1}, \quad \pi(\boldsymbol{s}_2) = \boldsymbol{a}_{2,1}$$

2. Example of randomized Markov policies?

# Example: Two-State MDP

Setting

- State set: $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2\}$
- Action set: $A_{s_1} = \{\boldsymbol{a}_{1,1}, \boldsymbol{a}_{1,2}\}$, $A_{s_2} = \{\boldsymbol{a}_{2,1}\}$
- Rewards: $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 5$, $r(\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 10$, $r(\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = -1$
- Transition probabilities: $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,1}) = 0.5$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_1, \boldsymbol{a}_{1,2}) = 1$
  $p(\boldsymbol{s}_1|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 0$, $p(\boldsymbol{s}_2|\boldsymbol{s}_2, \boldsymbol{a}_{2,1}) = 1$

Q)

1. Example of deterministic Markov policies?

   $$\pi(\boldsymbol{s}_1) = \boldsymbol{a}_{1,1}, \quad \pi(\boldsymbol{s}_2) = \boldsymbol{a}_{2,1}$$

2. Example of randomized Markov policies?

   $$\pi(\boldsymbol{a}_{1,1}|\boldsymbol{s}_1) = 0.7, \quad \pi(\boldsymbol{a}_{1,2}|\boldsymbol{s}_1) = 0.3, \quad \pi(\boldsymbol{a}_{2,1}|\boldsymbol{s}_2) = 1$$

# Infinite-Horizon MDP Problems

# Infinite-Horizon Discounted MDP

**Definition**

A discounted MDP is a tuple $\langle S, A, p, r, \gamma \rangle$, consisting of

- $S$: set of states (state space)
- $A$: set of actions (action space)
- $p$: state transition probability
  $p(s'|s,a) := \mathrm{Prob}(s_{t+1} = s'|s_t = s, a_t = a)$
- $r$: reward function
  $r(s_t, a_t) = r_t$
- $\gamma \in (0,1)$: discount factor

# The MDP Problem

To find an **optimal policy** that *maximizes the expected cumulative reward*:

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

- Difficult to solve *Q) Why?*
- Solution we'll study: Dynamic Programming (DP)

# Assumptions

- Stationary rewards and transition probabilities: $r(s, a)$ and $p(s'|s, a)$ do not vary over time.

# Assumptions

- Stationary rewards and transition probabilities:
  $r(s, a)$ and $p(s'|s, a)$ do not vary over time.

- Finite state and action sets:
  Thus, the reward $r(s, a)$ is bounded, i.e., there exists a constant $M$ such that
  $$|r(s, a)| < M \quad \forall s \in S, \, \forall a \in A.$$

# Assumptions

- Stationary rewards and transition probabilities:
  $r(s,a)$ and $p(s'|s,a)$ do not vary over time.

- Finite state and action sets:
  Thus, the reward $r(s,a)$ is bounded, i.e., there exists a constant $M$ such that

$$|r(s,a)| < M \quad \forall s \in S,\ \forall a \in A.$$

- Discounting: $0 \leq \gamma < 1$

# Policy Evaluation

- Let's first consider a simple problem of evaluating

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

given a policy $\pi$ (Policy evaluation).

# Policy Evaluation

- Let's first consider a simple problem of evaluating

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

given a policy $\pi$ (Policy evaluation).

- A key concept for the policy evaluation is the *value function*.

**Definition (Value function)**

The value function $v^{\pi}(s)$ of a policy $\pi$ is the *expected return starting from state* $s$ under executing $\pi$:

$\boxed{=\text{cumulative reward}}$

# Policy Evaluation

- Let's first consider a simple problem of evaluating

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

given a policy $\pi$ (Policy evaluation).

- A key concept for the policy evaluation is the *value function*.

**Definition (Value function)**

The value function $v^{\pi}(s)$ of a policy $\pi$ is the *expected return starting from state* $s$ under executing $\pi$:

=cumulative reward

$$v^{\pi}(s) := \mathbb{E}^{\pi}\left[\sum_{\tau=t}^{\infty}\gamma^{\tau-t}r(s_\tau,a_\tau)\mid s_t=s\right]$$

## Stationary Policies

- In principle, a policy is given by

$$\pi := (\pi_0, \pi_1, \pi_2, \ldots)$$

## Stationary Policies

- In principle, a policy is given by

$$\pi := (\pi_0, \pi_1, \pi_2, \ldots)$$

- The policy is said to be *stationary* if

$$\pi_i = \pi_j \quad \forall i, j.$$

## Stationary Policies

- In principle, a policy is given by

$$\pi := (\pi_0, \pi_1, \pi_2, \ldots)$$

- The policy is said to be *stationary* if

$$\pi_i = \pi_j \quad \forall i, j.$$

- We just consider a stationary policy as a decision rule, i.e.,

$$\pi(s_t) = a_t \quad \text{(deterministic stationary)}$$

or

$$\pi(a_t|s_t) \quad \text{(stochastic stationary)}$$

# Policy Evaluation

Idea:

- Decompose the value function into

# Policy Evaluation

Idea:

- Decompose the value function into
  $(i)$ immediate reward; plus

# Policy Evaluation

Idea:

- Decompose the value function into
  $(i)$ immediate reward; plus
  $(ii)$ discounted value of next state:

# Policy Evaluation

Idea:

- Decompose the value function into
  $(i)$ immediate reward; plus
  $(ii)$ discounted value of next state:

$$
\begin{aligned}
v^{\pi}(s) &= \mathbb{E}^{\pi}[\underbrace{r(s_t, a_t)}_{\text{immediate reward}} + \gamma \underbrace{v^{\pi}(s_{t+1})}_{\text{value of next state}} \mid s_t = s] \\
&= \sum_{a \in A} \pi(a|s) \left( r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a) v^{\pi}(s') \right)
\end{aligned}
$$

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

- $v^\pi := (v^\pi(1), \ldots, v^\pi(n)) \in \mathbb{R}^n$

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

- $v^\pi := (v^\pi(1), \ldots, v^\pi(n)) \in \mathbb{R}^n$
- $R^\pi := (\sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) r(1, \boldsymbol{a}), \ldots, \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) r(n, \boldsymbol{a})) \in \mathbb{R}^n$

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

- $v^\pi := (v^\pi(1), \ldots, v^\pi(n)) \in \mathbb{R}^n$
- $R^\pi := (\sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) r(1, \boldsymbol{a}), \ldots, \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) r(n, \boldsymbol{a})) \in \mathbb{R}^n$
  When $\pi$ is deterministic, it's simplified to
  $R^\pi := (r(1, \pi(1)), \ldots, r(n, \pi(n))) \in \mathbb{R}^n$

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

- $v^\pi := (v^\pi(1), \ldots, v^\pi(n)) \in \mathbb{R}^n$
- $R^\pi := (\sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) r(1, \boldsymbol{a}), \ldots, \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) r(n, \boldsymbol{a})) \in \mathbb{R}^n$
  When $\pi$ is deterministic, it's simplified to
  $R^\pi := (r(1, \pi(1)), \ldots, r(n, \pi(n))) \in \mathbb{R}^n$
- Transition probability matrix:

$$P^\pi := \begin{bmatrix} \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) p(1|1, \boldsymbol{a}) & \cdots & \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) p(n|1, \boldsymbol{a}) \\ \vdots & \ddots & \vdots \\ \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) p(1|n, \boldsymbol{a}) & \cdots & \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) p(n|n, \boldsymbol{a}) \end{bmatrix} \in \mathbb{R}^{n \times n}$$

# Vector Form

Let $S := \{1, \ldots, n\}$ and $A := \{1, \ldots, m\}$.
We often use the following notation:

- $v^\pi := (v^\pi(1), \ldots, v^\pi(n)) \in \mathbb{R}^n$
- $R^\pi := (\sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) r(1, \boldsymbol{a}), \ldots, \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) r(n, \boldsymbol{a})) \in \mathbb{R}^n$
  When $\pi$ is deterministic, it's simplified to
  $R^\pi := (r(1, \pi(1)), \ldots, r(n, \pi(n))) \in \mathbb{R}^n$

  in deterministic =>
  P =[ [Sig_(a){p(1|1,a)} ... Sig_(a){p(n|1,a)}],
  ...
  [Sig_(a){p(1|n,a)} ... Sig_(a){p(n|n,a)}]]
- Transition probability matrix:

$$P^\pi := \begin{bmatrix} \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) p(1|1, \boldsymbol{a}) & \cdots & \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|1) p(n|1, \boldsymbol{a}) \\ \vdots & \ddots & \vdots \\ \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) p(1|n, \boldsymbol{a}) & \cdots & \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|n) p(n|n, \boldsymbol{a}) \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Then, the policy evaluation equation can be expressed as

$$\underbrace{v^\pi}_{\boxed{\text{nx1}}} = \underbrace{R^\pi}_{\boxed{\text{nx1}}} + \gamma \underbrace{P^\pi}_{\boxed{\text{nxn}}} \underbrace{v^\pi}_{\boxed{\text{nx1}}}.$$

## Properties

- The eigenvalues of $P^\pi$ are less than or equal to $1$.

## Properties

- The eigenvalues of $P^\pi$ are less than or equal to $1$.
- The linear equation $v = R^\pi + \gamma P^\pi v$ has a unique solution. Q) Why?

# Properties

- The eigenvalues of $P^\pi$ are less than or equal to $1$.
- The linear equation $v = R^\pi + \gamma P^\pi v$ has a unique solution. Q) Why?
- The unique solution is given by

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

# Properties

- The eigenvalues of $P^\pi$ are less than or equal to $1$.
- The linear equation $v = R^\pi + \gamma P^\pi v$ has a unique solution. Q) Why?
- The unique solution is given by

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

- This method is inefficient for large-scale problems.

# Value Iteration

## Operator Form

- Let $\mathcal{T}^\pi : \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$\mathcal{T}^\pi v := R^\pi + \gamma P^\pi v.$$

## Operator Form

- Let $\mathcal{T}^\pi : \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$\mathcal{T}^\pi v := R^\pi + \gamma P^\pi v.$$

- Thus,

$$(\mathcal{T}^\pi v)(\boldsymbol{s}) = \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|\boldsymbol{s}) \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

# Operator Form

- Let $\mathcal{T}^\pi : \mathbb{R}^n \to \mathbb{R}^n$ be defined by

$$\mathcal{T}^\pi v := R^\pi + \gamma P^\pi v.$$

- Thus,

$$(\mathcal{T}^\pi v)(\boldsymbol{s}) = \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|\boldsymbol{s}) \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

- The policy evaluation equation can be expressed as

$$v^\pi = \mathcal{T}^\pi v^\pi,$$

or

$$v^\pi(s) = (\mathcal{T}^\pi v^\pi)(s),$$

which is a fixed point problem.

# Contraction Property

**Definition**

An operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ is said to be a $\gamma$-*contraction* with respect to $\|\cdot\|$ if

$$\|\mathcal{T}v - \mathcal{T}v'\| \leq \gamma\|v - v'\| \quad \forall v, v' \in \mathbb{R}^n.$$

# Contraction Property

**Definition**

An operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ is said to be a $\gamma$-*contraction* with respect to $\|\cdot\|$ if

$$\|\mathcal{T}v - \mathcal{T}v'\| \leq \gamma\|v - v'\| \quad \forall v, v' \in \mathbb{R}^n.$$

**Proposition**

*The operator $\mathcal{T}^\pi$ is a $\gamma$-contraction with respect to $\|\cdot\|_\infty$ for any stationary policy $\pi$.*

## Banach Fixed Point Theorem

**Theorem**

*Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ be a contraction mapping. Then,*

# Banach Fixed Point Theorem

**Theorem**

*Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ be a contraction mapping. Then,*

- *$\mathcal{T}$ admits a unique fixed point $v^* \in \mathbb{R}^n$, i.e., $v^* = \mathcal{T} v^*$.*

# Banach Fixed Point Theorem

**Theorem**

*Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ be a contraction mapping. Then,*

- *$\mathcal{T}$ admits a unique fixed point $v^* \in \mathbb{R}^n$, i.e., $v^* = \mathcal{T} v^*$.*
- *$v^*$ can be found by value iteration:*
  *start with an arbitrary $v_0 \in \mathbb{R}^n$ and define a sequence $\{v_k\}$ by*

  $$v_{k+1} := \mathcal{T} v_k.$$

  *Then, $v_k \to v^*$.*

# Banach Fixed Point Theorem

**Theorem**

*Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ be a contraction mapping. Then,*

- *$\mathcal{T}$ admits a unique fixed point $v^* \in \mathbb{R}^n$, i.e., $v^* = \mathcal{T}v^*$.*
- *$v^*$ can be found by value iteration:*
  *start with an arbitrary $v_0 \in \mathbb{R}^n$ and define a sequence $\{v_k\}$ by*

  $$v_{k+1} := \mathcal{T}v_k.$$

  *Then, $v_k \to v^*$.*

Remark:

- Our policy evaluation equation has a unique solution.
- $v^\pi$ can be obtained by value iteration.

# Value Iteration Algorithm for Policy Evaluation

Input: stationary policy $\pi$

# Value Iteration Algorithm for Policy Evaluation

Input: stationary policy $\pi$

- Initialize $v_0$ as an arbitrary vector in $\mathbb{R}^n$;

# Value Iteration Algorithm for Policy Evaluation

Input: stationary policy $\pi$

- Initialize $v_0$ as an arbitrary vector in $\mathbb{R}^n$;
- Repeat until convergence

$$v_{k+1} := \mathcal{T}^{\pi} v_k;$$

# From policy evaluation to optimal policy

To find an **optimal policy** that *maximizes the expected cumulative reward*:

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

## From policy evaluation to optimal policy

To find an **optimal policy** that *maximizes the expected cumulative reward*:

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

**Definition (Optimal value function)**

The optimal value function $v^*(s)$ is the maximum value function over all policies:

# From policy evaluation to optimal policy

To find an **optimal policy** that *maximizes the expected cumulative reward*:

$$\max_{\pi \in \Pi} \ \mathbb{E}^{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

**Definition (Optimal value function)**

The optimal value function $v^*(s)$ is the maximum value function over all policies:

$$v^*(s) := \max_{\pi \in \Pi} \ v^{\pi}(s) = \max_{\pi \in \Pi} \ \mathbb{E}^{\pi}\left[\sum_{\tau=t}^{\infty} \gamma^{\tau-t} r(s_\tau, a_\tau) \mid s_t = s\right]$$

## Bellman Operator

**Definition**

We define the *Bellman operator* $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\mathcal{T}v := \sup_{\pi}\{R^\pi + \gamma P^\pi v\},$$

# Bellman Operator

**Definition**

We define the *Bellman operator* $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\mathcal{T}v := \sup_{\pi}\{R^{\pi} + \gamma P^{\pi} v\},$$

i.e.,

$$(\mathcal{T}v)(\boldsymbol{s}) := \sup_{\pi} \sum_{\boldsymbol{a} \in A} \pi(\boldsymbol{a}|\boldsymbol{s}) \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

## Bellman Operator

**Definition**

We define the *Bellman operator* $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\mathcal{T}v := \sup_{\pi}\{R^\pi + \gamma P^\pi v\},$$

i.e.,

$$(\mathcal{T}v)(\boldsymbol{s}) := \sup_{\pi} \sum_{\boldsymbol{a}\in A} \pi(\boldsymbol{a}|\boldsymbol{s})\left[r(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})v(\boldsymbol{s}')\right].$$

- We can rewrite the Bellman operator as

$$(\mathcal{T}v)(\boldsymbol{s}) = \max_{\boldsymbol{a}\in A}\left[r(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})v(\boldsymbol{s}')\right].$$

Q) Why?

# Contraction Property and Monotonicity

**Proposition**

*The Bellman operator $\mathcal{T}$ is a $\gamma$-contraction mapping with respect to $\|\cdot\|_\infty$, i.e.,*

$$\|\mathcal{T}v - \mathcal{T}v'\|_\infty \leq \gamma \|v - v'\|_\infty \quad \forall v, v' \in \mathbb{R}^n.$$

*Furthermore, it is monotone, i.e.,*

$$\mathcal{T}v \leq \mathcal{T}v' \quad \forall v, v' \in \mathbb{R}^n \text{ s.t. } v \leq v'.$$

# Bellman Equation

Operator form:

$$v = \mathcal{T} v.$$

Equation form:

$$v(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A} \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

# Bellman Equation

Operator form:

$$v = \mathcal{T} v.$$

Equation form:

$$v(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A} \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

- This equation has a unique solution. Q) Why?

# Bellman Equation

Operator form:

$$v = \mathcal{T} v.$$

Equation form:

$$v(\boldsymbol{s}) = \max_{\boldsymbol{a} \in A} \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \right].$$

- This equation has a unique solution. Q) Why?
- The unique solution is the optimal value function $v^*$. Q) Why?

# Optimal Value Function and Bellman Equation

**Theorem**

*The Bellman equation has a unique solution, which coincides with the optimal value function, i.e.,*

$$v^* = \mathcal{T} v^*.$$

# Optimal Value Function and Bellman Equation

**Theorem**

*The Bellman equation has a unique solution, which coincides with the optimal value function, i.e.,*

$$v^* = \mathcal{T} v^*.$$

- How to solve this equation?

## Value Iteration

- Initialize $v_0$ as an arbitrary vector in $\mathbb{R}^n$;
- Repeat until convergence

$$v_{k+1} := \mathcal{T} v_k;$$

## Value Iteration

- Initialize $v_0$ as an arbitrary vector in $\mathbb{R}^n$;
- Repeat until convergence

$$v_{k+1} := \mathcal{T} v_k;$$

Then,

$$v_k \to v^*.$$

# Existence of Optimal Policies

- Does an optimal policy exist?
- How to construct an optimal policy?

# Existence of Optimal Policies

- Does an optimal policy exist?
- How to construct an optimal policy?

**Theorem**

*Suppose that $S$ and $A$ are finite sets. Then, there exists an optimal policy, which is deterministic and stationary.*

## Existence of Optimal Policies

- Does an optimal policy exist?
- How to construct an optimal policy?

**Theorem**

*Suppose that $S$ and $A$ are finite sets. Then, there exists an optimal policy, which is deterministic and stationary.*

- Such an optimal policy can be obtained as

# Existence of Optimal Policies

- Does an optimal policy exist?
- How to construct an optimal policy?

**Theorem**

*Suppose that $S$ and $A$ are finite sets. Then, there exists an optimal policy, which is deterministic and stationary.*

- Such an optimal policy can be obtained as

$$A \to \left\{ \pi^*(\boldsymbol{s}) \right\} \in \overset{\boxed{\text{argmax\_a}}}{\cancel{\max_{\boldsymbol{a}\in A}}} \left[ r(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'\in S} p(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) v^*(\boldsymbol{s}') \right] \quad \forall \boldsymbol{s} \in S.$$

# Policy Iteration

# Idea of Policy Iteration

- In value iteration, we only update information about the value function.
- Can we also use information about policies?
- Yes, we can iteratively update both value functions and policies.

# Policy Iteration

- Initialize $\pi_0$ as an arbitrary deterministic stationary policy;
- Repeat until convergence
  - (policy evaluation) Compute the value function of $\pi_k$ by solving

$$v^{\pi_k} = \mathcal{T}^{\pi_k} v^{\pi_k};$$

  - (policy improvement) Update the policy as

$$\pi_{k+1}(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a} \in A} \left[ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}' \in S} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v^{\pi_k}(\boldsymbol{s}') \right];$$

Properties:

- The sequence of policies converges to an optimal policy.
  Q) Why?

# Monotonic Improvement

**Lemma**

*Let $\{\pi_k\}$ be the sequence of policies obtained by policy iteration. Then, we have*

$$v^{\pi_k} \le v^{\pi_{k+1}} \quad \forall k.$$

**Theorem (Monotone Convergence Theorem)**

*Let $\{x_k\}$ be a monotonically non-decreasing, bounded sequence of real vectors. Then, the sequence has a finite limit.*

# Convergence of Policy Iteration

**Theorem**

*Let $\{\pi_k\}$ be the sequence of policies obtained by policy iteration. Then, it converges to an optimal policy, i.e.,*

$$\pi_k \to \pi^* \quad \textit{as } k \to \infty.$$

# Value Iteration (VI) vs Policy Iteration (PI)

- VI: $v_{k+1} \leftarrow \mathcal{T} v_k$

# Value Iteration (VI) vs Policy Iteration (PI)

- VI: $v_{k+1} \leftarrow \mathcal{T} v_k$
- PI: (policy evaluation)
  $v^{\pi_k} = \mathcal{T}^{\pi_k} v^{\pi_k}$ and
  (policy improvement)
  $\pi_{k+1}(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a}} \{ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \}$

# Value Iteration (VI) vs Policy Iteration (PI)

- VI: $v_{k+1} \leftarrow \mathcal{T} v_k$
- PI: (policy evaluation)
  $v^{\pi_k} = \mathcal{T}^{\pi_k} v^{\pi_k}$ and
  (policy improvement)
  $\pi_{k+1}(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a}} \{ r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'} p(\boldsymbol{s}' | \boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}') \}$
- VI is simpler; but PI is often faster (but not always)

# Value Iteration (VI) vs Policy Iteration (PI)

- VI: $v_{k+1} \leftarrow \mathcal{T} v_k$
- PI: (policy evaluation)
  $v^{\pi_k} = \mathcal{T}^{\pi_k} v^{\pi_k}$ and
  (policy improvement)
  $\pi_{k+1}(\boldsymbol{s}) \in \arg\max_{\boldsymbol{a}} \{r(\boldsymbol{s}, \boldsymbol{a}) + \gamma \sum_{\boldsymbol{s}'} p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) v(\boldsymbol{s}')\}$
- VI is simpler; but PI is often faster (but not always)
- Both are very important algorithms which are the basis of various RL methods.