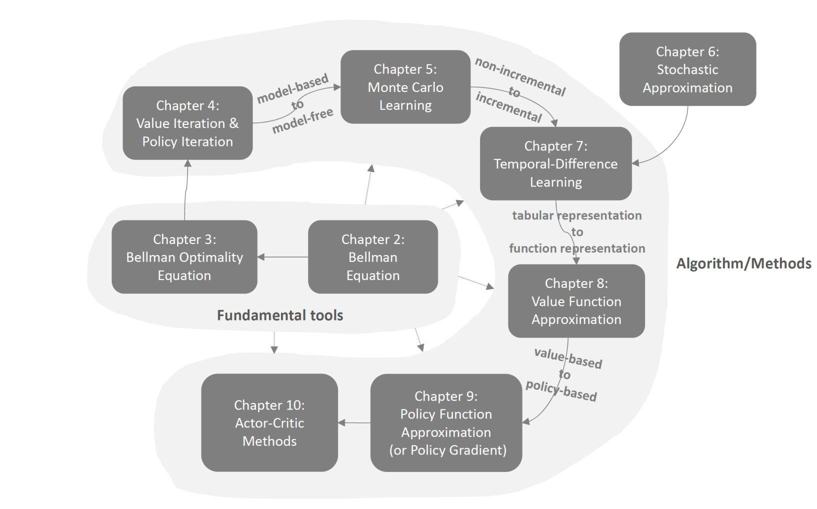
Optimal Policy

and

Bellman Optimality Equation

Shiyu Zhao

Outline



Chapter 4:
Value Iteration &
Policy Iteration
model-based
to
model-free
Chapter 5:
Monte Carlo
Learning
non-incremental
to
incremental
Chapter 6:
Stochastic
Approximation
Chapter 7:
Temporal-Difference
Learning
tabular representation
to
function representation
Chapter 3:
Bellman Optimality
Equation
Chapter 2:
Bellman
Equation
Algorithm/Methods
Chapter 8:
Value Function
Approximation
Fundamental tools
value-based
to
policy-based
Chapter 10:
Actor-Critic
Methods
Chapter 9:
Policy Function
Approximation
(or Policy Gradient)

Outline

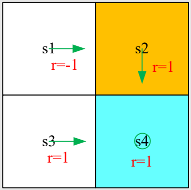
In this lecture:

- Core concepts: optimal state value and optimal policy
- A fundamental tool: the Bellman optimality equation (BOE)

Outline

Outline

Motivating examples



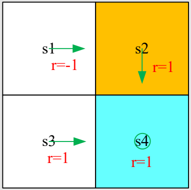
Bellman equation:

$$
\begin{aligned}
v_\pi(s_1) &= -1 + \gamma v_\pi(s_2), \\
v_\pi(s_2) &= +1 + \gamma v_\pi(s_4), \\
v_\pi(s_3) &= +1 + \gamma v_\pi(s_4), \\
v_\pi(s_4) &= +1 + \gamma v_\pi(s_4).
\end{aligned}
$$

State value: Let $\gamma = 0.9$. Then, it can be calculated that

$$v_\pi(s_4) = v_\pi(s_3) = v_\pi(s_2) = 10, \quad v_\pi(s_1) = 8.$$

Motivating examples



Action value: consider s_1

$$q_\pi(s_1, a_1) = -1 + \gamma v_\pi(s_1) = 6.2,$$

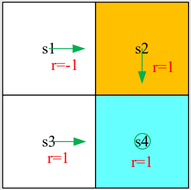
$$q_\pi(s_1, a_2) = -1 + \gamma v_\pi(s_2) = 8,$$

$$q_\pi(s_1, a_3) = 0 + \gamma v_\pi(s_3) = 9,$$

$$q_\pi(s_1, a_4) = -1 + \gamma v_\pi(s_1) = 6.2,$$

$$q_\pi(s_1, a_5) = 0 + \gamma v_\pi(s_1) = 7.2.$$

Motivating examples



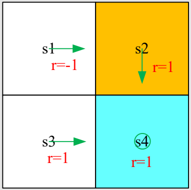
Question: While the policy is not good, how can we improve it?

Answer: by using action values.

The current policy $\pi(a|s_1)$ is

$$\pi(a|s_1) = \begin{cases} 1 & a = a_2 \\ 0 & a \neq a_2 \end{cases}$$

Motivating examples



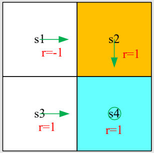
Question: While the policy is not good, how can we improve it?

Answer: by using action values.

The current policy $\pi(a|s_1)$ is

$$\pi(a|s_1) = \begin{cases} 1 & a = a_2 \\ 0 & a \neq a_2 \end{cases}$$

Motivating examples



Observe the action values that we obtained just now:

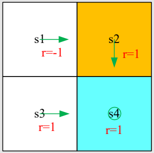
$$q_\pi(s_1, a_1) = 6.2, q_\pi(s_1, a_2) = 8, q_\pi(s_1, a_3) = 9,$$
$$q_\pi(s_1, a_4) = 6.2, q_\pi(s_1, a_5) = 7.2.$$

What if we select the greatest action value? Then, a new policy is obtained:

$$\pi_{\text{new}}(a|s_1) = \begin{cases} 1 & a = a^* \\ 0 & a \neq a^* \end{cases}$$

where $a^* = \arg\max_a q_\pi(s_1, a) = a_3$.

Motivating examples



Observe the action values that we obtained just now:

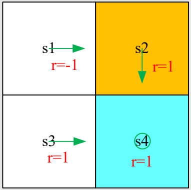
$$q_\pi(s_1, a_1) = 6.2, q_\pi(s_1, a_2) = 8, q_\pi(s_1, a_3) = 9,$$
$$q_\pi(s_1, a_4) = 6.2, q_\pi(s_1, a_5) = 7.2.$$

What if we select the greatest action value? Then, a new policy is obtained:

$$\pi_{\text{new}}(a|s_1) = \begin{cases} 1 & a = a^* \\ 0 & a \neq a^* \end{cases}$$

where $a^* = \arg\max_a q_\pi(s_1, a) = a_3$.

Motivating examples



Question: why doing this can improve the policy?

- Intuition: action values can be used to evaluate actions.
- Math: nontrivial and will be introduced in this lecture.

Outline

Optimal policy

The state value could be used to evaluate if a policy is good or not: if

$$v_{\pi_1}(s) \geq v_{\pi_2}(s) \quad \text{for all } s \in \mathcal{S}$$

then π_1 is "better" than π_2.

The definition leads to many questions:

- Does the optimal policy exist?
- Is the optimal policy unique?
- Is the optimal policy stochastic or deterministic?
- How to obtain the optimal policy?

To answer these questions, we study the *Bellman optimality equation*.

Optimal policy

The state value could be used to evaluate if a policy is good or not: if

$$v_{\pi_1}(s) \geq v_{\pi_2}(s) \quad \text{for all } s \in \mathcal{S}$$

then π_1 is "better" than π_2.

Definition

A policy π^* is optimal if $v_{\pi^*}(s) \geq v_\pi(s)$ for all s and for any other policy π.

The definition leads to many questions:

- Does the optimal policy exist?
- Is the optimal policy unique?
- Is the optimal policy stochastic or deterministic?
- How to obtain the optimal policy?

To answer these questions, we study the *Bellman optimality equation*.

Optimal policy

The state value could be used to evaluate if a policy is good or not: if

$$v_{\pi_1}(s) \geq v_{\pi_2}(s) \quad \text{for all } s \in \mathcal{S}$$

then π_1 is "better" than π_2.

Definition

A policy π^* is optimal if $v_{\pi^*}(s) \geq v_\pi(s)$ for all s and for any other policy π.

The definition leads to many questions:

- Does the optimal policy exist?
- Is the optimal policy unique?
- Is the optimal policy stochastic or deterministic?
- How to obtain the optimal policy?

To answer these questions, we study the *Bellman optimality equation*.

Optimal policy

The state value could be used to evaluate if a policy is good or not: if

$$v_{\pi_1}(s) \geq v_{\pi_2}(s) \quad \text{for all } s \in \mathcal{S}$$

then π_1 is "better" than π_2.

Definition

A policy π^* is optimal if $v_{\pi^*}(s) \geq v_\pi(s)$ for all s and for any other policy π.

The definition leads to many questions:

- Does the optimal policy exist?
- Is the optimal policy unique?
- Is the optimal policy stochastic or deterministic?
- How to obtain the optimal policy?

To answer these questions, we study the *Bellman optimality equation*.

Outline

Bellman optimality equation (elementwise form):

$$v(s) = \qquad \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S}$$

Bellman optimality equation (BOE)

Bellman optimality equation (elementwise form):

$$v(s) = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S}$$

Bellman optimality equation (BOE)

Bellman optimality equation (elementwise form):

$$
\begin{aligned}
v(s) &= \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S} \\
&= \max_{\pi} \sum_a \pi(a|s) q(s,a) \quad s \in \mathcal{S}
\end{aligned}
$$

Bellman optimality equation (BOE)

Bellman optimality equation (elementwise form):

$$\begin{aligned} v(s) &= \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S} \\ &= \max_{\pi} \sum_a \pi(a|s)q(s,a) \quad s \in \mathcal{S} \end{aligned}$$

Remarks:

- $p(r|s,a), p(s'|s,a)$ are known.
- $v(s), v(s')$ are unknown and to be calculated.
- Is $\pi(s)$ known or unknown?

Bellman optimality equation (BOE)

Bellman optimality equation (matrix-vector form):

$$v = \max_{\pi}(r_\pi + \gamma P_\pi v)$$

where the elements corresponding to s or s' are

$$[r_\pi]_s \triangleq \sum_a \pi(a|s) \sum_r p(r|s,a)r,$$

$$[P_\pi]_{s,s'} = p(s'|s) \triangleq \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)$$

Here $\max_\pi$ is performed elementwise.

Bellman optimality equation (BOE)

Bellman optimality equation (matrix-vector form):

$$v = \max_{\pi}(r_\pi + \gamma P_\pi v)$$

BOE is tricky yet elegant!

- Why elegant? It describes the optimal policy and optimal state value in an elegant way.
- Why tricky? There is a maximization on the right-hand side, which may not be straightforward to see how to compute.
- Many questions to answer:
  - Algorithm: how to solve this equation?
  - Existence: does this equation have solutions?
  - Uniqueness: is the solution to this equation unique?
  - Optimality: how is it related to optimal policy?

Bellman optimality equation (BOE)

Bellman optimality equation (matrix-vector form):

$$v = \max_{\pi}(r_\pi + \gamma P_\pi v)$$

BOE is tricky yet elegant!

- Why elegant? It describes the optimal policy and optimal state value in an elegant way.
- Why tricky? There is a maximization on the right-hand side, which may not be straightforward to see how to compute.
- Many questions to answer:
  - Algorithm: how to solve this equation?
  - Existence: does this equation have solutions?
  - Uniqueness: is the solution to this equation unique?
  - Optimality: how is it related to optimal policy?

Outline

Maximization on the right-hand side of BOE

BOE: elementwise form

$$v(s) = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S}$$

BOE: matrix-vector form $v = \max_{\pi}(r_{\pi} + \gamma P_{\pi} v)$

Maximization on the right-hand side of BOE

BOE: elementwise form

$$v(s) = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S}$$

BOE: matrix-vector form $v = \max_{\pi}(r_{\pi} + \gamma P_{\pi} v)$

Example (How to solve two unknowns from one equation)

Consider two variables $x, a \in \mathbb{R}$. Suppose they satisfy

$$x = \max_a (2x - 1 - a^2).$$

This equation has two unknowns. To solve them, first consider the right hand side. Regardless the value of x, $\max_a(2x - 1 - a^2) = 2x - 1$ where the maximization is achieved when $a = 0$. Second, when $a = 0$, the equation becomes $x = 2x - 1$, which leads to $x = 1$. Therefore, $a = 0$ and $x = 1$ are the solution of the equation.

Maximization on the right-hand side of BOE

Fix $v'(s)$ first and solve π:

$$\begin{aligned}v(s) &= \max_\pi \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right), \quad \forall s \in \mathcal{S} \\ &= \max_\pi \sum_a \pi(a|s) q(s,a)\end{aligned}$$

Maximization on the right-hand side of BOE

Fix $v'(s)$ first and solve π:

$$\begin{aligned} v(s) &= \max_\pi \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right), \quad \forall s \in \mathcal{S} \\ &= \max_\pi \sum_a \pi(a|s) q(s,a) \end{aligned}$$

Example (How to solve $\max_\pi \sum_a \pi(a|s)q(s,a)$)

Suppose $q_1, q_2, q_3 \in \mathbb{R}$ are given. Find c_1^*, c_2^*, c_3^* solving

$$\max_{c_1,c_2,c_3} c_1 q_1 + c_2 q_2 + c_3 q_3.$$

where $c_1 + c_2 + c_3 = 1$ and $c_1, c_2, c_3 \geq 0$.

Without loss of generality, suppose $q_3 \geq q_1, q_2$. Then, the optimal solution is $c_3^* = 1$ and $c_1^* = c_2^* = 0$. That is because for any c_1, c_2, c_3

$$q_3 = (c_1 + c_2 + c_3) q_3 = c_1 q_3 + c_2 q_3 + c_3 q_3 \geq c_1 q_1 + c_2 q_2 + c_3 q_3.$$

Maximization on the right-hand side of BOE

Fix $v'(s)$ first and solve π:

$$v(s) = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right), \quad \forall s \in \mathcal{S}$$
$$= \max_{\pi} \sum_a \pi(a|s) q(s,a)$$

Inspired by the above example, considering that $\sum_a \pi(a|s) = 1$, we have

$$\max_{\pi} \sum_a \pi(a|s) q(s,a) = \max_{a \in \mathcal{A}(s)} q(s,a),$$

where the optimality is achieved when

$$\pi(a|s) = \begin{cases} 1 & a = a^* \\ 0 & a \neq a^* \end{cases}$$

where $a^* = \arg\max_a q(s,a)$.

Outline

Solve the Bellman optimality equation

The BOE is $v = \max_\pi(r_\pi + \gamma P_\pi v)$. Let

$$f(v) := \max_\pi(r_\pi + \gamma P_\pi v)$$

Then, the Bellman optimality equation becomes

$$v = f(v)$$

where

$$[f(v)]_s = \max_\pi \sum_a \pi(a|s)q(s,a), \quad s \in \mathcal{S}$$

Next, how to solve the equation?

Solve the Bellman optimality equation

The BOE is $v = \max_\pi (r_\pi + \gamma P_\pi v)$. Let

$$f(v) := \max_\pi (r_\pi + \gamma P_\pi v)$$

Then, the Bellman optimality equation becomes

$$v = f(v)$$

where

$$[f(v)]_s = \max_\pi \sum_a \pi(a|s) q(s,a), \quad s \in \mathcal{S}$$

Next, how to solve the equation?

Outline

Preliminaries: Contraction mapping theorem

Some concepts:

- Fixed point: $x \in X$ is a fixed point of $f : X \to X$ if

$$f(x) = x$$

- Contraction mapping (or contractive function): f is a contraction mapping if

$$\|f(x_1) - f(x_2)\| \leq \gamma \|x_1 - x_2\|$$

where $\gamma \in (0, 1)$.

  - γ must be strictly less than 1 so that many limits such as $\gamma^k \to 0$ as $k \to 0$ hold.
  - Here $\|\cdot\|$ can be any vector norm.

Preliminaries: Contraction mapping theorem

Some concepts:

- Fixed point: $x \in X$ is a fixed point of $f : X \to X$ if

$$f(x) = x$$

- Contraction mapping (or contractive function): f is a contraction mapping if

$$\|f(x_1) - f(x_2)\| \leq \gamma \|x_1 - x_2\|$$

  where $\gamma \in (0, 1)$.
  - γ must be strictly less than 1 so that many limits such as $\gamma^k \to 0$ as $k \to 0$ hold.
  - Here $\| \cdot \|$ can be any vector norm.

Preliminaries: Contraction mapping theorem

Examples to demonstrate the concepts.

Example

- $x = f(x) = 0.5x$, $x \in \mathbb{R}$.

  It is easy to verify that $x = 0$ is a fixed point since $0 = 0.5 \times 0$.

  Moreover, $f(x) = 0.5x$ is a contraction mapping because $\|0.5x_1 - 0.5x_2\| = 0.5\|x_1 - x_2\| \leq \gamma\|x_1 - x_2\|$ for any $\gamma \in [0.5, 1)$.

Preliminaries: Contraction mapping theorem

Examples to demonstrate the concepts.

Example

- $x = f(x) = 0.5x$, $x \in \mathbb{R}$.

  It is easy to verify that $x = 0$ is a fixed point since $0 = 0.5 \times 0$.

  Moreover, $f(x) = 0.5x$ is a contraction mapping because

  $\|0.5x_1 - 0.5x_2\| = 0.5\|x_1 - x_2\| \le \gamma\|x_1 - x_2\|$ for any $\gamma \in [0.5, 1)$.

- $x = f(x) = Ax$, where $x \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}$ and $\|A\| \le \gamma < 1$.

  It is easy to verify that $x = 0$ is a fixed point since $0 = A0$. To see the contraction property,

  $\|Ax_1 - Ax_2\| = \|A(x_1 - x_2)\| \le \|A\|\|x_1 - x_2\| \le \gamma\|x_1 - x_2\|$.

  Therefore, $f(x) = Ax$ is a contraction mapping.

Preliminaries: Contraction mapping theorem

Theorem (Contraction Mapping Theorem)

For any equation that has the form of $x = f(x)$, if f is a contraction mapping, then

- *Existence: there exists a fixed point x^* satisfying $f(x^*) = x^*$.*
- *Uniqueness: The fixed point x^* is unique.*
- *Algorithm: Consider a sequence $\{x_k\}$ where $x_{k+1} = f(x_k)$, then $x_k \to x^*$ as $k \to \infty$. Moreover, the convergence rate is exponentially fast.*

For the proof of this theorem, see the book.

Preliminaries: Contraction mapping theorem

Examples:

- $x = 0.5x$, where $f(x) = 0.5x$ and $x \in \mathbb{R}$

  $x^* = 0$ is the unique fixed point. It can be solved iteratively by

$$x_{k+1} = 0.5x_k$$

- $x = Ax$, where $f(x) = Ax$ and $x \in \mathbb{R}^n, A \in \mathbb{R}^{n\times n}$ and $\|A\| < 1$

  $x^* = 0$ is the unique fixed point. It can be solved iteratively by

$$x_{k+1} = Ax_k$$

Preliminaries: Contraction mapping theorem

Examples:

- $x = 0.5x$, where $f(x) = 0.5x$ and $x \in \mathbb{R}$
  $x^* = 0$ is the unique fixed point. It can be solved iteratively by
  $$x_{k+1} = 0.5x_k$$
- $x = Ax$, where $f(x) = Ax$ and $x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}$ and $\|A\| < 1$
  $x^* = 0$ is the unique fixed point. It can be solved iteratively by
  $$x_{k+1} = Ax_k$$

Outline

Contraction property of BOE

Let's come back to the Bellman optimality equation:

$$v = f(v) = \max_{\pi}(r_\pi + \gamma P_\pi v)$$

For the proof of this lemma, see our book.

Contraction property of BOE

Let's come back to the Bellman optimality equation:

$$v = f(v) = \max_{\pi}(r_{\pi} + \gamma P_{\pi} v)$$

Theorem (Contraction Property)

$f(v)$ is a contraction mapping satisfying

$$\|f(v_1) - f(v_2)\| \leq \gamma \|v_1 - v_2\|$$

where γ is the discount rate!

For the proof of this lemma, see our book.

Solve the Bellman optimality equation

Applying the contraction mapping theorem gives the following results.

Theorem (Existence, Uniqueness, and Algorithm)

For the BOE $v = f(v) = \max_\pi(r_\pi + \gamma P_\pi v)$*, there always* **exists** *a solution* v^* *and the solution is* **unique**. *The solution could be solved iteratively by*

$$v_{k+1} = f(v_k) = \max_\pi(r_\pi + \gamma P_\pi v_k)$$

This sequence $\{v_k\}$ *converges to* v^* **exponentially fast** *given any initial guess* v_0. *The convergence rate is determined by* γ.

Solve the Bellman optimality equation

The iterative algorithm:

Matrix-vector form:

$$v_{k+1} = f(v_k) = \max_\pi (r_\pi + \gamma P_\pi v_k)$$

Elementwise form:

$$\begin{aligned} v_{k+1}(s) &= \max_\pi \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v_k(s') \right) \\ &= \max_\pi \sum_a \pi(a|s) q_k(s,a) \\ &= \max_a q_k(s,a) \end{aligned}$$

Solve the Bellman optimality equation

The iterative algorithm:

Matrix-vector form:

$$v_{k+1} = f(v_k) = \max_\pi (r_\pi + \gamma P_\pi v_k)$$

Elementwise form:

$$\begin{aligned}
v_{k+1}(s) &= \max_\pi \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v_k(s') \right) \\
&= \max_\pi \sum_a \pi(a|s) q_k(s,a) \\
&= \max_a q_k(s,a)
\end{aligned}$$

Solve the Bellman optimality equation

Procedure summary:

- For any s, current estimated value $v_k(s)$
- For any $a \in \mathcal{A}(s)$, calculate $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_k(s')$
- Calculate the greedy policy π_{k+1} for s as

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = a_k^*(s) \\ 0 & a \neq a_k^*(s) \end{cases}$$

  where $a_k^*(s) = \arg\max_a q_k(s,a)$.
- Calculate $v_{k+1}(s) = \max_a q_k(s,a)$

The above algorithm is actually the value iteration algorithm as discussed in the next lecture.

Solve the Bellman optimality equation

Procedure summary:

- For any s, current estimated value $v_k(s)$
- For any $a \in \mathcal{A}(s)$, calculate
  $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_k(s')$
- Calculate the greedy policy π_{k+1} for s as

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = a_k^*(s) \\ 0 & a \neq a_k^*(s) \end{cases}$$

  where $a_k^*(s) = \arg\max_a q_k(s,a)$.
- Calculate $v_{k+1}(s) = \max_a q_k(s,a)$

The above algorithm is actually the value iteration algorithm as discussed in the next lecture.

Solve the Bellman optimality equation

Procedure summary:

- For any s, current estimated value $v_k(s)$
- For any $a \in \mathcal{A}(s)$, calculate
  $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_k(s')$
- Calculate the greedy policy π_{k+1} for s as

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = a_k^*(s) \\ 0 & a \neq a_k^*(s) \end{cases}$$

  where $a_k^*(s) = \arg\max_a q_k(s,a)$.

- Calculate $v_{k+1}(s) = \max_a q_k(s,a)$

The above algorithm is actually the value iteration algorithm as discussed in the next lecture.

Solve the Bellman optimality equation

Procedure summary:

- For any s, current estimated value $v_k(s)$
- For any $a \in \mathcal{A}(s)$, calculate
  $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a) v_k(s')$
- Calculate the greedy policy π_{k+1} for s as

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = a_k^*(s) \\ 0 & a \neq a_k^*(s) \end{cases}$$

  where $a_k^*(s) = \arg\max_a q_k(s,a)$.
- Calculate $v_{k+1}(s) = \max_a q_k(s,a)$

The above algorithm is actually the value iteration algorithm as discussed in the next lecture.

Solve the Bellman optimality equation

Procedure summary:

- For any s, current estimated value $v_k(s)$
- For any $a \in \mathcal{A}(s)$, calculate
  $q_k(s,a) = \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_k(s')$
- Calculate the greedy policy π_{k+1} for s as

$$\pi_{k+1}(a|s) = \begin{cases} 1 & a = a_k^*(s) \\ 0 & a \neq a_k^*(s) \end{cases}$$

  where $a_k^*(s) = \arg\max_a q_k(s,a)$.
- Calculate $v_{k+1}(s) = \max_a q_k(s,a)$

The above algorithm is actually the value iteration algorithm as discussed in the next lecture.

Example

Example: Manually solve the BOE.

- Why manually? Can understand better.
- Why so simple example? Can be calculated manually.

Actions: a_ℓ, a_0, a_r represent go left, stay unchanged, and go right.

Reward: entering the target area: $+1$; try to go out of boundary -1.

Example

The values of $q(s, a)$

q-value table	a_ℓ	a_0	a_r
s_1	$-1 + \gamma v(s_1)$	$0 + \gamma v(s_1)$	$1 + \gamma v(s_2)$
s_2	$0 + \gamma v(s_1)$	$1 + \gamma v(s_2)$	$0 + \gamma v(s_3)$
s_3	$1 + \gamma v(s_2)$	$0 + \gamma v(s_3)$	$-1 + \gamma v(s_3)$

Consider $\gamma = 0.9$

Example

The values of $q(s,a)$

q-value table	a_ℓ	a_0	a_r
s_1	$-1+\gamma v(s_1)$	$0+\gamma v(s_1)$	$1+\gamma v(s_2)$
s_2	$0+\gamma v(s_1)$	$1+\gamma v(s_2)$	$0+\gamma v(s_3)$
s_3	$1+\gamma v(s_2)$	$0+\gamma v(s_3)$	$-1+\gamma v(s_3)$

Consider $\gamma = 0.9$

Example

Our objective is to find $v^*(s_i)$ and π^*

$k = 0$:

v-value: select $v_0(s_1) = v_0(s_2) = v_0(s_3) = 0$

q-value (using the previous table):

	a_ℓ	a_0	a_r
s_1	-1	0	1
s_2	0	1	0
s_3	1	0	-1

Greedy policy (select the greatest q-value)

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

v-value: $v_1(s) = \max_a q_0(s, a)$

$$v_1(s_1) = v_1(s_2) = v_1(s_3) = 1$$

This this policy good? Yes!

Example

Our objective is to find $v^*(s_i)$ and π^*

$k = 0$:

v-value: select $v_0(s_1) = v_0(s_2) = v_0(s_3) = 0$

q-value (using the previous table):

	a_ℓ	a_0	a_r
s_1	-1	0	1
s_2	0	1	0
s_3	1	0	-1

Greedy policy (select the greatest q-value)

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

v-value: $v_1(s) = \max_a q_0(s, a)$

$$v_1(s_1) = v_1(s_2) = v_1(s_3) = 1$$

This this policy good? Yes!

Example

Our objective is to find $v^*(s_i)$ and π^*

$k = 0$:

v-value: select $v_0(s_1) = v_0(s_2) = v_0(s_3) = 0$

q-value (using the previous table):

	a_ℓ	a_0	a_r
s_1	-1	0	1
s_2	0	1	0
s_3	1	0	-1

Greedy policy (select the greatest q-value)

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

v-value: $v_1(s) = \max_a q_0(s,a)$

$$v_1(s_1) = v_1(s_2) = v_1(s_3) = 1$$

This this policy good? Yes!

Example

- $k = 1$:

  Excise: With $v_1(s)$ calculated in the last step, calculate by yourself.

  q-value:

	a_ℓ	a_0	a_r
s_1	-0.1	0.9	1.9
s_2	0.9	1.9	0.9
s_3	1.9	0.9	-0.1

  Greedy policy (select the greatest q-value):

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

  The policy is the same as the previous one, which is already optimal.

  v-value: $v_2(s) = ...$

- $k = 2, 3, \ldots$

Example

- $k = 1$:

  Excise: With $v_1(s)$ calculated in the last step, calculate by yourself.

  q-value:

	a_ℓ	a_0	a_r
s_1	-0.1	0.9	1.9
s_2	0.9	1.9	0.9
s_3	1.9	0.9	-0.1

  Greedy policy (select the greatest q-value):

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

  The policy is the same as the previous one, which is already optimal.

  v-value: $v_2(s) = ...$

- $k = 2, 3, \ldots$

Example

- $k = 1$:

  Excise: With $v_1(s)$ calculated in the last step, calculate by yourself.

  q-value:

	a_ℓ	a_0	a_r
s_1	-0.1	0.9	1.9
s_2	0.9	1.9	0.9
s_3	1.9	0.9	-0.1

  Greedy policy (select the greatest q-value):

$$\pi(a_r|s_1) = 1, \quad \pi(a_0|s_2) = 1, \quad \pi(a_\ell|s_3) = 1$$

  The policy is the same as the previous one, which is already optimal.

  v-value: $v_2(s) = ...$

- $k = 2, 3, \ldots$

Outline

Policy optimality

Suppose v^* is the solution to the Bellman optimality equation. It satisfies

$$v^* = \max_\pi (r_\pi + \gamma P_\pi v^*)$$

Suppose

$$\pi^* = \arg\max_\pi (r_\pi + \gamma P_\pi v^*)$$

Then

$$v^* = r_{\pi^*} + \gamma P_{\pi^*} v^*$$

Therefore, π^* is a policy and $v^* = v_{\pi^*}$ is the corresponding state value.

Is π^* the optimal policy? Is v^* the greatest state value can be achieved?

Policy optimality

Theorem (Policy Optimality)

Suppose that v^* is the unique solution to $v = \max_\pi(r_\pi + \gamma P_\pi v)$, and v_π is the state value function satisfying $v_\pi = r_\pi + \gamma P_\pi v_\pi$ for any given policy π, then

$$v^* \geq v_\pi, \quad \forall \pi$$

For the proof, please see our book.

Now we understand why we study the BOE. That is because it describes the optimal state value and optimal policy.

Optimal policy

What does an optimal policy π^* look like?

Theorem (Greedy Optimal Policy)

For any $s \in \mathcal{S}$, the deterministic greedy policy

$$\pi^*(a|s) = \begin{cases} 1 & a = a^*(s) \\ 0 & a \neq a^*(s) \end{cases} \tag{1}$$

is an optimal policy solving the BOE. Here,

$$a^*(s) = \arg\max_a q^*(a, s),$$

where $q^*(s,a) := \sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v^*(s')$.

Proof: simple. $\pi^*(s) = \arg\max_\pi \sum_a \pi(a|s) \underbrace{\left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v^*(s') \right)}_{q^*(s,a)}$

Outline

Analyzing optimal policies

What factors determine the optimal policy?

It can be clearly seen from the BOE

$$v(s) = \max_{\pi} \sum_{a} \pi(a|s) \left(\sum_{r} p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right)$$

that there are three factors:

- Reward design: r
- System model: $p(s'|s,a)$, $p(r|s,a)$
- Discount rate: γ
- $v(s), v(s'), \pi(a|s)$ are unknowns to be calculated

Next, we use examples to show how changing r and γ can change the optimal policy.

Analyzing optimal policies

What factors determine the optimal policy?

It can be clearly seen from the BOE

$$v(s) = \max_{\pi} \sum_{a} \pi(a|s) \left(\sum_{r} p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right)$$

that there are three factors:

- Reward design: r
- System model: $p(s'|s,a)$, $p(r|s,a)$
- Discount rate: γ
- $v(s), v(s'), \pi(a|s)$ are unknowns to be calculated

Next, we use examples to show how changing r and γ can change the optimal policy.

Analyzing optimal policies

What factors determine the optimal policy?

It can be clearly seen from the BOE

$$v(s) = \max_{\pi} \sum_a \pi(a|s) \left(\sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v(s') \right)$$

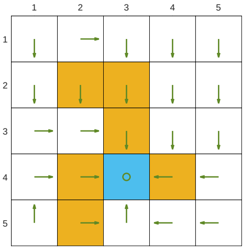
that there are three factors:

- Reward design: r
- System model: $p(s'|s,a)$, $p(r|s,a)$
- Discount rate: γ
- $v(s), v(s'), \pi(a|s)$ are unknowns to be calculated

Next, we use examples to show how changing r and γ can change the optimal policy.

Analyzing optimal policies

The optimal policy and the corresponding optimal state value are obtained by solving the BOE.

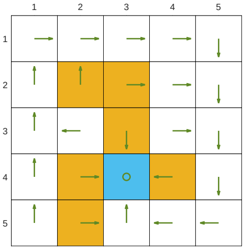


	1	2	3	4	5
1	5.8	5.6	6.2	6.5	5.8
2	6.5	7.2	8.0	7.2	6.5
3	7.2	8.0	10.0	8.0	7.2
4	8.0	10.0	10.0	10.0	8.0
5	7.2	9.0	10.0	9.0	8.1

(a) $r_{\text{boundary}} = r_{\text{forbidden}} = -1,\ r_{\text{target}} = 1,\ \gamma = 0.9$

The optimal policy dares to take risks: entering forbidden areas!!

If we change $\gamma = 0.9$ to $\gamma = 0.5$



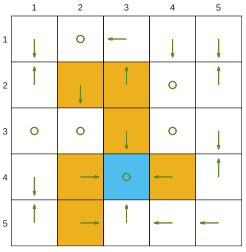
	1	2	3	4	5
1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.1
3	0.0	0.0	2.0	0.1	0.1
4	0.0	2.0	2.0	2.0	0.2
5	0.0	1.0	2.0	1.0	0.5

(b) The discount rate is $\gamma = 0.5$. Others are the same as (a).

The optimal policy becomes shorted-sighted! Avoid all the forbidden areas!

Analyzing optimal policies

If we change γ to 0



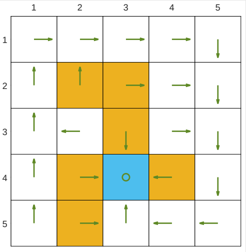
	1	2	3	4	5
1	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	0.0	0.0
4	0.0	1.0	1.0	1.0	0.0
5	0.0	0.0	1.0	0.0	0.0

(c) The discount rate is $\gamma = 0$. Others are the same as (a).

The optimal policy becomes extremely short-sighted! Also, choose the action that has the greatest *immediate reward*! Cannot reach the target!

Analyzing optimal policies

If we increase the punishment when entering forbidden areas ($r_{\text{forbidden}} = -1$ to $r_{\text{forbidden}} = -10$)



	1	2	3	4	5
1	3.5	3.9	4.3	4.8	5.3
2	3.1	3.5	4.8	5.3	5.9
3	2.8	2.5	10.0	5.9	6.6
4	2.5	10.0	10.0	10.0	7.3
5	2.3	9.0	10.0	9.0	8.1

(d) $r_{\text{forbidden}} = -10$. Others are the same as (a).

The optimal policy would also avoid the forbidden areas.

Analyzing optimal policies

What if we change $r \rightarrow ar + b$?

For example,

$$r_{\text{boundary}} = r_{\text{forbidden}} = -1, \quad r_{\text{target}} = 1$$

becomes

$$r_{\text{boundary}} = r_{\text{forbidden}} = 0, \quad r_{\text{target}} = 2, \quad r_{\text{otherstep}} = 1$$

The optimal policy remains the same!

What matters is not the absolute reward values! It is their relative values!

Analyzing optimal policies

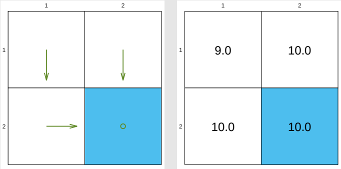
Theorem (Optimal Policy Invariance)

Consider a Markov decision process with $v^* \in \mathbb{R}^{|\mathcal{S}|}$ as the optimal state value satisfying $v^* = \max_\pi(r_\pi + \gamma P_\pi v^*)$. If every reward r is changed by an affine transformation to $ar + b$, where $a, b \in \mathbb{R}$ and $a \neq 0$, then the corresponding optimal state value v' is also an affine transformation of v^*:

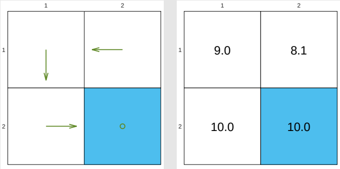
$$v' = av^* + \frac{b}{1-\gamma}\mathbf{1},$$

where $\gamma \in (0,1)$ is the discount rate and $\mathbf{1} = [1, \ldots, 1]^T$. Consequently, the optimal policies are invariant to the affine transformation of the reward signals.

Meaningless detour?



(a) Optimal policy



(b) Not optimal

The policy in (a) is optimal, the policy in (b) is not.

Question: Why the optimal policy is not (b)? Why does the optimal policy not take meaningless detours? There is no punishment for taking detours!!

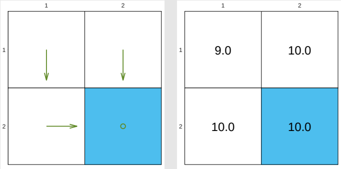
Due to the discount rate!

Policy (a): $\text{return} = 1 + \gamma 1 + \gamma^2 1 + \cdots = 1/(1-\gamma) = 10.$

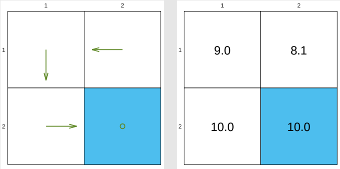
Policy (b): $\text{return} = 0 + \gamma 0 + \gamma^2 1 + \gamma^3 1 + \cdots = \gamma^2/(1-\gamma) = 8.1$

Analyzing optimal policies

Meaningless detour?



(a) Optimal policy



(b) Not optimal

The policy in (a) is optimal, the policy in (b) is not.

Question: Why the optimal policy is not (b)? Why does the optimal policy not take meaningless detours? There is no punishment for taking detours!!

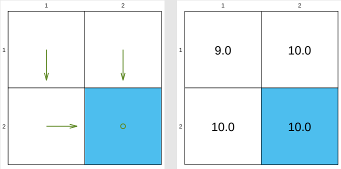
Due to the discount rate!

Policy (a): $\text{return} = 1 + \gamma 1 + \gamma^2 1 + \cdots = 1/(1-\gamma) = 10.$

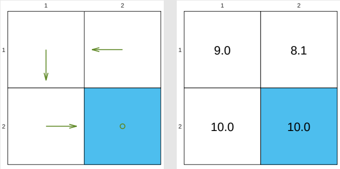
Policy (b): $\text{return} = 0 + \gamma 0 + \gamma^2 1 + \gamma^3 1 + \cdots = \gamma^2/(1-\gamma) = 8.1$

Analyzing optimal policies

Meaningless detour?



(a) Optimal policy



(b) Not optimal

The policy in (a) is optimal, the policy in (b) is not.

Question: Why the optimal policy is not (b)? Why does the optimal policy not take meaningless detours? There is no punishment for taking detours!!

Due to the discount rate!

Policy (a): $\text{return} = 1 + \gamma 1 + \gamma^2 1 + \cdots = 1/(1-\gamma) = 10.$

Policy (b): $\text{return} = 0 + \gamma 0 + \gamma^2 1 + \gamma^3 1 + \cdots = \gamma^2/(1-\gamma) = 8.1$

Summary

Bellman optimality equation:

- Elementwise form:

$$v(s) = \max_{\pi} \sum_{a} \pi(a|s) \underbrace{\left(\sum_{r} p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v(s') \right)}_{q(s,a)}, \quad \forall s \in \mathcal{S}$$

- Matrix-vector form:

$$v = \max_{\pi}(r_{\pi} + \gamma P_{\pi} v)$$

Summary

Questions about the Bellman optimality equation:

- Existence: does this equation have solutions?
  - Yes, by the contraction mapping Theorem
- Uniqueness: is the solution to this equation unique?
  - Yes, by the contraction mapping Theorem
- Algorithm: how to solve this equation?
  - Iterative algorithm suggested by the contraction mapping Theorem
- Optimality: why we study this equation
  - Because its solution corresponds to the optimal state value and optimal policy.

Finally, we understand why it is important to study the BOE!