



Markov Decision Processes

Optimal Policies and Optimal Value Functions

Sarwan Ali

Department of Computer Science
Georgia State University

 Understanding Optimal Decision Making 

Today's Learning Journey

- 1 Introduction to Optimality in MDPs
- 2 Value Functions
- 3 Optimal Value Functions
- 4 Optimal Policies
- 5 Properties and Theorems
- 6 Examples and Applications
- 7 Summary and Next Steps

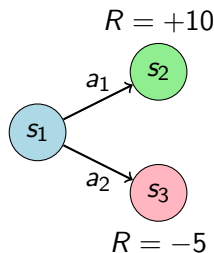
What Makes a Policy Optimal?

Key Question

How do we determine the **best** way to act in an MDP?

Intuitive Understanding:

- Maximize long-term rewards
- Balance immediate vs. future gains
- Handle uncertainty optimally



Challenge

Need mathematical framework to define and find optimal policies!

State Value Function $V^\pi(s)$

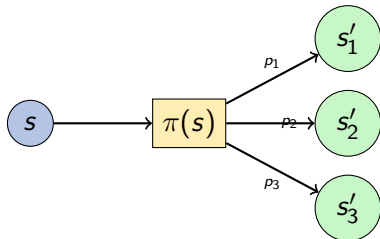
Definition

The **state value function** $V^\pi(s)$ is the expected return when starting from state s and following policy π :

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s \right]$$

Key Components:

- \mathbb{E}_π : Expectation under policy π
- γ^t : Discount factor for time t
- R_{t+1} : Reward at time $t + 1$
- $S_0 = s$: Starting state condition



Action Value Function $Q^\pi(s, a)$

Definition

The **action value function** $Q^\pi(s, a)$ is the expected return when starting from state s , taking action a , and then following policy π : $Q^\pi(s, a) = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t R_{t+1} \mid S_0 = s, A_0 = a]$

Relationship between V^π and Q^π

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

$$Q^\pi(s, a) = \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^\pi(s')]$$

Intuition

$Q^\pi(s, a)$ tells us "how good" it is to take action a in state s under policy π .

Bellman Equation for V^π

Theorem (Bellman Equation for State Value Function)

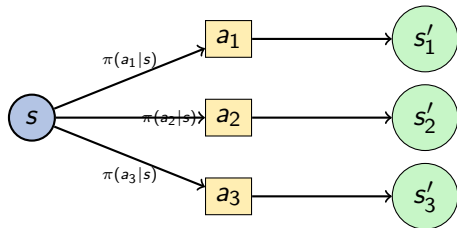
For any policy π and state s : $V^\pi(s) = \sum_a \pi(a|s) \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^\pi(s')]$

Breakdown:

$$V^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma V^\pi(S_{t+1}) | S_t = s] \quad (1)$$

$$= \sum_a \pi(a|s) \mathbb{E}[R_{t+1} + \gamma V^\pi(S_{t+1}) | S_t = s, A_t = a] \quad (2)$$

$$= \sum_a \pi(a|s) Q^\pi(s, a) \quad (3)$$



Key Insight

The value of a state equals the expected immediate reward plus the discounted value of the next state.

Optimal State Value Function $V^*(s)$

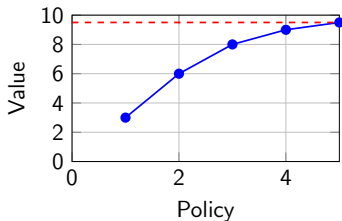
Definition

The **optimal state value function** $V^*(s)$ is:

$$V^*(s) = \max_{\pi} V^{\pi}(s) \quad \text{for all } s \in \mathcal{S}$$

Properties:

- Unique for each MDP
- Independent of initial policy
- Represents best possible performance
- Satisfies Bellman optimality equation



Interpretation

$V^*(s)$ is the maximum expected return achievable from state s .

Optimal Action Value Function $Q^*(s, a)$

Definition

The **optimal action value function** $Q^*(s, a)$ is:

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \quad \text{for all } s \in \mathcal{S}, a \in \mathcal{A}$$

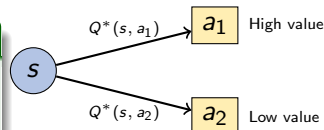
Relationship with V^*

$$V^*(s) = \max_a Q^*(s, a)$$

$$Q^*(s, a) = \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^*(s')]$$

Key Insight

$Q^*(s, a)$ gives us the expected return of taking action a in state s and then acting optimally thereafter.



Bellman Optimality Equation

Theorem (Bellman Optimality Equation for V^*)

$$V^*(s) = \max_a \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^*(s')]$$

Theorem (Bellman Optimality Equation for Q^*)

$$Q^*(s, a) = \sum_{s'} p(s'|s, a) \left[r(s, a, s') + \gamma \max_{a'} Q^*(s', a') \right]$$

System of Equations

For an MDP with n states, we have n nonlinear equations with n unknowns.

- If we can solve this system, we find V^*
- From V^* , we can derive the optimal policy
- Solution exists and is unique

Challenge

These are nonlinear equations due to the max operator!

Definition of Optimal Policy

Definition

A policy π^* is **optimal** if:

$$V^{\pi^*}(s) = V^*(s) \quad \text{for all } s \in \mathcal{S}$$

Theorem (Existence of Optimal Policy)

- *For any finite MDP, there exists at least one optimal policy*
- *All optimal policies share the same optimal value function V^**
- *All optimal policies share the same optimal action value function Q^**

Deterministic Optimal Policy

There always exists an optimal policy that is deterministic and stationary:

$$\pi^*(s) = \arg \max_a Q^*(s, a) = \arg \max_a \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^*(s')]$$

Policy Ordering

Definition (Policy Ordering)

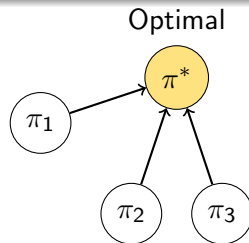
For two policies π and π' , we say $\pi \geq \pi'$ if: $V^\pi(s) \geq V^{\pi'}(s)$ for all $s \in \mathcal{S}$

Theorem (Optimal Policy Theorem)

There exists an optimal policy π^ such that $\pi^* \geq \pi$ for all policies π .*

Implications:

- Policies can be ranked
- Optimal policy dominates all others
- May be multiple optimal policies
- All optimal policies are equivalent



Extracting Optimal Policy from V^*

Method 1: One-step Lookahead

Given V^* , the optimal policy is: $\pi^*(s) = \arg \max_a \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^*(s')]$

Method 2: Using Q^*

Given Q^* , the optimal policy is: $\pi^*(s) = \arg \max_a Q^*(s, a)$

Example: Grid World

$V^* = 8.8$ ↓	$V^* = 9.2$ ↓	$V^* \odot 10$ ↓
$V^* = 8.4$	$V^* = 8.8$	$V^* = 9.2$
$V^* = 8$	$V^* = 8.4$	$V^* = 8.8$

Key Properties of Optimal Functions

Uniqueness

- V^* is unique for each MDP
- Q^* is unique for each MDP
- Multiple optimal policies may exist, but they all achieve the same value

Consistency

If π^* is optimal, then:

$$V^{\pi^*}(s) = V^*(s) \quad (4)$$

$$Q^{\pi^*}(s, a) = Q^*(s, a) \quad (5)$$

Principle of Optimality

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Computational Complexity

Challenges

Finding optimal policies is computationally intensive:

- Bellman optimality equations are nonlinear
- Direct solution requires solving system of equations
- Curse of dimensionality: state space grows exponentially

Solution Methods

Exact Methods:

- Value Iteration
- Policy Iteration
- Linear Programming

Approximate Methods:

- Function Approximation
- Monte Carlo Methods
- Temporal Difference Learning

Time Complexity

For finite MDPs: $O(|\mathcal{S}|^2 \cdot |\mathcal{A}|)$ per iteration for most algorithms.

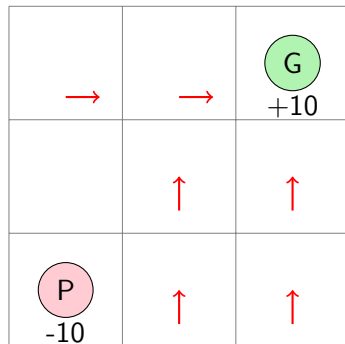
Example: Simple Grid World

Setup:

- 3×3 grid
- Goal at top-right (+10 reward)
- Pit at bottom-left (-10 reward)
- Step cost: -1
- $\gamma = 0.9$

Actions: Up, Down, Left, Right

Transition: 80% intended direction, 10% each perpendicular direction



Optimal Policy

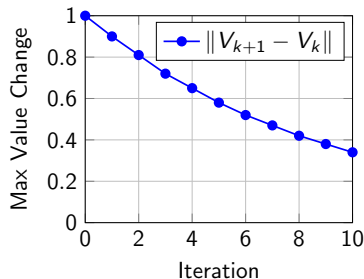
Value Iteration Convergence

Value Iteration Algorithm:

- 1 Initialize $V_0(s) = 0$ for all s
- 2 For $k = 0, 1, 2, \dots$:

$$V_{k+1}(s) = \max_a \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V_k(s')]$$

- 3 Stop when $\|V_{k+1} - V_k\| < \epsilon$



Theorem (Convergence of Value Iteration)

Value iteration converges to V^ for finite MDPs with $\gamma < 1$.*

$$V_k \rightarrow V^* \text{ as } k \rightarrow \infty$$

Real-World Applications

Robotics

- Path planning: $V^*(s)$ represents expected cost-to-go from current position
- Robot navigation in uncertain environments, Manipulation tasks with stochastic outcomes

Finance

- Portfolio optimization: $V^*(s)$ = maximum expected return from wealth state s
- Algorithmic trading: optimal buy/sell decisions, Risk management and hedging strategies

Game AI

- Chess, Go: $V^*(s)$ = value of board position s
- Video games: NPC behavior optimization, Multi-agent competitive environments

Healthcare

- Treatment planning: optimal therapy sequences
- Drug dosage optimization, Resource allocation in hospitals

Key Takeaways

Optimal Value Functions

- $V^*(s)$: Maximum expected return from state s
- $Q^*(s, a)$: Maximum expected return from taking action a in state s
- Both satisfy Bellman optimality equations. Unique solutions exist for finite MDPs

Optimal Policies

- Always exist for finite MDPs. Can be deterministic and stationary
- Extracted via: $\pi^*(s) = \arg \max_a Q^*(s, a)$. May be multiple optimal policies with same value

Bellman Optimality Equations

$$V^*(s) = \max_a \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma V^*(s')] \quad (6)$$

$$Q^*(s, a) = \sum_{s'} p(s'|s, a) [r(s, a, s') + \gamma \max_{a'} Q^*(s', a')] \quad (7)$$

Computational Challenges & Solutions

The Challenge

- Bellman optimality equations are **nonlinear**
- Curse of dimensionality: $|\mathcal{S}|$ and $|\mathcal{A}|$ can be huge
- Direct analytical solution often impossible

Solution Approaches

Dynamic Programming:

- Value Iteration
- Policy Iteration
- Guaranteed convergence
- Exact for finite MDPs

Approximate Methods:

- Reinforcement Learning
- Function Approximation
- Monte Carlo sampling
- Neural networks (Deep RL)

Next Topic Preview

We'll explore **Dynamic Programming** algorithms that solve these equations iteratively!

Practice Problems

Problem 1: Simple MDP

Consider a 2-state MDP with states $\{s_1, s_2\}$ and actions $\{a_1, a_2\}$:

- From s_1 : a_1 goes to s_2 (reward +1), a_2 stays in s_1 (reward 0)
- From s_2 : both actions return to s_1 (reward +2)
- $\gamma = 0.8$

Find: $V^*(s_1)$, $V^*(s_2)$, and π^*

Problem 2: Policy Comparison

For the grid world example, compare these policies:

- π_1 : Always go right
- π_2 : Always go towards goal (shortest path)
- π^* : Optimal policy

Calculate: $V^{\pi_1}(s)$, $V^{\pi_2}(s)$, $V^*(s)$ for center state

Questions?

Thank You!

✉ sali85@student.gsu.edu