Lecture 3

Value Functions, Bellman Equations, and Optimality

Goals of this lecture

1. Introduce state-value and action-value functions for evaluating policies.
2. Derive and interpret the Bellman expectation equations for value functions.
3. Formulate policy evaluation as a linear system in finite MDPs.
4. Explain greedy policies and the policy improvement theorem.
5. Define optimal value functions and derive the Bellman optimality equations.
6. Illustrate these concepts using a 2×2 Gridworld example.

3.1 State and Action Value Functions

We have established that, under the Markov assumption, it is sufficient to restrict attention to Markov stationary policies. We will therefore restrict our attention from now on to $\pi \in \Pi_M$. This enables us to reason about policies in terms of their behavior from each state individually, without reference to full histories. To evaluate and compare such policies, we define the value functions v^π and q^π, which quantify the expected return under a given policy π. These functions play a central role in both theoretical analysis and algorithm design.

State-value function. The *state-value function* $v^\pi(s)$ gives the expected return when the agent starts in state s and thereafter follows policy π:

$$v^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s], \quad \text{where} \quad G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k}.$$

The function $v^\pi(s)$ quantifies how desirable it is to be in state s when following policy π, and provides a foundation for comparing different policies based on their long-term behavior. Crucially, under the Markov and stationarity assumptions, this expectation is independent of the specific time step t; the value depends only on the state s, not on when the agent is in that state.

Action-value function. The *action-value function* $q^\pi(s, a)$ gives the expected return when the agent starts in state s, takes action a, and thereafter follows policy π:

$$q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a].$$

As with v^π, this quantity is independent of the specific time step t, depending only on the state-action pair (s, a) under stationarity. While $v^\pi(s)$ reflects the overall quality of a state under policy π, $q^\pi(s, a)$ captures the expected outcome of committing to action a at state s. This finer granularity makes q^π particularly valuable for policy improvement, as it allows us to ask: "How much better (or worse) is this action compared to what the policy would normally do?"

Relation between v^π and q^π. The state-value and action-value functions are closely related. In fact, $v^\pi(s)$ can be recovered from $q^\pi(s, a)$ by averaging over the policy's action distribution at state s:

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s)\, q^\pi(s, a).$$

This expresses the expected return at state s as a weighted combination of the expected returns for each possible action, weighted by the probability that policy π chooses each action. It highlights how q^π captures the fine-grained action-level structure that underlies the coarser state-level valuation encoded by v^π.

Rewriting the RL objective. The state-value and action-value functions provide a compact way to express the reinforcement learning objective. Given an initial state distribution ρ, the expected return under policy π is:

$$\mathbb{E}_\pi[G_0 \mid S_0 \sim \rho] = \mathbb{E}_{S_0 \sim \rho}\left[v^\pi(S_0)\right] = \sum_{s \in \mathcal{S}} \rho(s)\, v^\pi(s).$$

Thus, the RL objective reduces to finding a policy that maximizes this expected value:

$$\max_\pi \; \mathbb{E}_{S_0 \sim \rho}\left[v^\pi(S_0)\right].$$

Moreover, since $v^\pi(s) = \mathbb{E}_{A_0 \sim \pi(\cdot|s)}[q^\pi(s, A_0)] = \sum_{a \in \mathcal{A}} \pi(a|s) q^\pi(s, a)$, we can also rewrite the objective as:

$$\mathbb{E}_\pi[G_0 \mid S_0 \sim \rho] = \mathbb{E}_{S_0 \sim \rho,\, A_0 \sim \pi(\cdot|S_0)}\left[q^\pi(S_0, A_0)\right] = \sum_{s \in \mathcal{S}} \rho(s) \sum_{a \in \mathcal{A}} \pi(a \mid s) q^\pi(s, a),$$

which naturally leads to or, in summation form:

$$\max_\pi \mathbb{E}_{S_0 \sim \rho,\, A_0 \sim \pi(\cdot|S_0)}\left[q^\pi(S_0, A_0)\right].$$

3.2 Bellman Expectation Equations

The value functions v^π and q^π introduced earlier satisfy recursive identities known as the *Bellman expectation equations*. These identities express the value of a state (or state-action pair) in terms of immediate rewards and the value of subsequent states, under the assumption that the agent continues to follow policy π.

State-value function. The state-value function satisfies the following recursive equation:

$$v^\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma v^\pi(S_{t+1}) \mid S_t = s \right].$$

This equation states that the value of a state s under policy π is equal to the expected immediate reward plus the discounted expected value of the next state.

In environments with finite state, action, and reward spaces, this can be written in summation form as:

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma v^\pi(s') \right].$$

The following theorem rigorously show the derivation of the Bellman Expectation Equation for the value function v^π.

Theorem 3.1 (Bellman Expectation Equation for v^π)**.** *Let π be any Markov policy and $v^\pi(s)$ its corresponding state-value function under the discounted return:*

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k}.$$

Then for all $s \in \mathcal{S}$, the value function satisfies the recursive identity:

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma v^\pi(s') \right].$$

Proof. Recall the definition of the value function:

$$v^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s], \quad \text{where} \quad G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k}.$$

By unfolding the return:

$$v^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] = \mathbb{E}_\pi[R_{t+1} \mid S_t = s] + \gamma \, \mathbb{E}_\pi[G_{t+1} \mid S_t = s].$$

(i) Immediate reward term. Apply the law of total expectation over actions:

$$\begin{aligned} \mathbb{E}_\pi[R_{t+1} \mid S_t = s] &= \sum_{a \in \mathcal{A}} \pi(a \mid s) \, \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] \\ &= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \, r. \end{aligned}$$

(ii) Expected future value term. Again applying the law of total expectation:

$$\mathbb{E}_\pi[G_{t+1} \mid S_t = s] = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} p(s' \mid s, a) \, v^\pi(s').$$

(iii) Combine both terms. We now substitute both components back into the original expression:

$$v^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma v^\pi(s') \right],$$

which completes the proof. □

Action-value function. The action-value function satisfies a similar recursive identity:

$$q^\pi(s,a) = \mathbb{E}_\pi \left[R_{t+1} + \gamma q^\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a \right].$$

Here, the next action A_{t+1} is drawn from the policy $\pi(\cdot \mid S_{t+1})$, reflecting continued use of π.

In the finite case, this expands to:

$$q^\pi(s,a) = \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma \sum_{a' \in \mathcal{A}} \pi(a' \mid s')\, q^\pi(s', a') \right].$$

Theorem 3.2 (Bellman Expectation Equation for q^π)**.** *Let π be any Markov policy and $q^\pi(s,a)$ its corresponding action-value function under the discounted return:*

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k}.$$

Then for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, the action-value function satisfies the recursive identity:

$$q^\pi(s,a) = \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma \sum_{a' \in \mathcal{A}} \pi(a' \mid s')\, q^\pi(s', a') \right].$$

Proof. The proof is analogous to the value function case and is omitted here. □

Relationship Between v^π and q^π. The state-value and action-value functions are deeply interconnected and can be computed from one another.

- **From q^π to v^π:**

$$v^\pi(s) = \mathbb{E}_\pi \left[q^\pi(S_t, A_t) | S_t = s \right] = \sum_{a \in \mathcal{A}} \pi(a \mid s)\, q^\pi(s,a).$$

  This represents the expected value of the action selected by π at state s.

- **From v^π to q^π:**

$$q^\pi(s,a) = \mathbb{E} \left[R_{t+1} + \gamma v^\pi(S_{t+1}) | S_t = s, A_t = a \right] = \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \, \left[r + \gamma v^\pi(s') \right].$$

  This gives the expected return of taking action a in state s, then following policy π.

These expressions are fundamental for policy evaluation and are heavily used in both value-based and policy-based reinforcement learning algorithms.

While these Bellman expectation equations define the value functions precisely, they are *implicit*: the unknown function appears on both sides of the equation. At first glance, it is not obvious how to compute v^π or q^π directly. The next section shows how, in the finite state and action case, these equations can be reformulated as a linear system and solved using matrix inversion.

3.3 Exact Policy Evaluation in Finite MDPs

We have seen that the value function v^π satisfies a recursive identity—known as the Bellman expectation equation—that expresses each state's value in terms of the expected reward and the value of successor states. While this provides an elegant conceptual definition, it also reveals an important practical difficulty: the value function is defined implicitly in terms of itself.

To make this precise, recall that for every state $s \in \mathcal{S}$, the value under policy π satisfies:

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s) \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s', r \mid s, a) \left[r + \gamma v^\pi(s')\right].$$

This defines a system of $|\mathcal{S}|$ equations in $|\mathcal{S}|$ unknowns—one for each state—but the equations are coupled: to compute $v^\pi(s)$, we must already know the values $v^\pi(s')$ for all possible next states s'.

Thus, computing v^π requires solving a set of interdependent equations. In general, this cannot be done by direct substitution or incremental computation alone. However, in the case of finite state and action spaces, we can express the full system in matrix-vector form, enabling the use of efficient numerical linear algebra techniques for exact solution.

Matrix notation. Let us fix a canonical ordering of the states $\mathcal{S} = \{s_1, \ldots, s_n\}$, and define:

- $\mathbf{v}_\pi \in \mathbb{R}^n$: the column vector of state values, with entries $v^\pi(s_i)$, i.e. $[\mathbf{v}^\pi]_i = v^\pi(s_i)$,
- $\mathbf{r}_\pi \in \mathbb{R}^n$: the expected immediate reward vector under policy π, with components

$$[\mathbf{r}_\pi]_i = \sum_a \pi(a \mid s_i) \sum_{s',r} p(s', r \mid s_i, a)\, r,$$

- $\mathbf{P}_\pi \in \mathbb{R}^{n\times n}$: the state-transition matrix induced by π, where

$$[\mathbf{P}_\pi]_{ij} = \sum_a \pi(a \mid s_i)\, p(s_j \mid s_i, a), \quad \text{and } p(s_j \mid s_i, a) = \sum_r p(s_j, r \mid s_i, a).$$

Bellman equation in matrix form. We begin with the Bellman expectation equation for the value of state s_i:

$$v^\pi(s_i) = \sum_{a\in\mathcal{A}} \pi(a \mid s_i) \sum_{s_j\in\mathcal{S}} \sum_{r\in\mathcal{R}} p(s_j, r \mid s_i, a)\ [r + \gamma v^\pi(s_j)].$$

We now distribute the sum:

$$v^\pi(s_i) = \sum_a \pi(a \mid s_i) \sum_{s_j,r} p(s_j, r \mid s_i, a)\, r + \gamma \sum_a \pi(a \mid s_i) \sum_{s_j,r} p(s_j, r \mid s_i, a)\, v^\pi(s_j).$$

In the second term, the value $v^\pi(s_j)$ is independent of r, so we can factor it out:

$$v^\pi(s_i) = \sum_a \pi(a \mid s_i) \sum_{s_j,r} p(s_j, r \mid s_i, a)\, r + \gamma \sum_a \pi(a \mid s_i) \sum_{s_j} \left(\sum_r p(s_j, r \mid s_i, a) \right) v^\pi(s_j).$$

Now define:

$$p(s_j \mid s_i, a) := \sum_r p(s_j, r \mid s_i, a),$$

and plug this in:

$$v^\pi(s_i) = \sum_a \pi(a \mid s_i) \sum_{s_j, r} p(s_j, r \mid s_i, a)\, r + \gamma \sum_a \pi(a \mid s_i) \sum_{s_j} p(s_j \mid s_i, a)\, v^\pi(s_j).$$

We now group the first term and define:

$$[\mathbf{r}_\pi]_i := \sum_a \pi(a \mid s_i) \sum_{s_j, r} p(s_j, r \mid s_i, a)\, r,$$

and define the entries of the transition matrix under policy π:

$$[\mathbf{P}_\pi]_{ij} := \sum_a \pi(a \mid s_i)\, p(s_j \mid s_i, a).$$

Thus, the second term becomes:

$$\sum_{s_j} [\mathbf{P}_\pi]_{ij} \cdot v^\pi(s_j) = [\mathbf{P}_\pi \mathbf{v}_\pi]_i.$$

Putting it all together, we arrive at:

$$v^\pi(s_i) = [\mathbf{r}_\pi]_i + \gamma [\mathbf{P}_\pi \mathbf{v}_\pi]_i,$$

which holds for all states s_i, so we recover the matrix form:

$$\mathbf{v}_\pi = \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{v}_\pi.$$

Rearranging gives a linear system:

$$(\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{v}_\pi = \mathbf{r}_\pi,$$

where $\mathbf{I}$ is the identity matrix.

Existence and uniqueness of the solution. For $0 < \gamma < 1$, the matrix $(\mathbf{I} - \gamma \mathbf{P}_\pi)$ is invertible because $\gamma \mathbf{P}_\pi$ is a contraction (its spectral radius is strictly less than 1). Therefore, the system has a unique solution:

$$\mathbf{v}_\pi = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \mathbf{r}_\pi.$$

Interpretation. This formulation shows that evaluating a fixed policy π in a finite MDP reduces to solving a system of linear equations. Although the Bellman equations are defined recursively, the matrix form allows us to solve for all state values simultaneously using standard techniques. This insight forms the foundation for exact dynamic programming methods such as policy iteration and provides a baseline for evaluating the performance of approximate methods in large or continuous environments.

In the next section, we build on this foundation to explore how value estimates can be used to improve policies—and ultimately compute optimal behavior.

3.4 Greedy Policies and Policy Improvement

Motivation for Greedy Improvement. Once we have evaluated a policy π by computing its action-value function $q^\pi(s,a)$, it becomes natural to ask: can we do better? Recall that the state-value function is given by

$$v^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s)\, q^\pi(s,a),$$

which represents the expected return in state s under policy π.

If there exists an action a such that $q^\pi(s,a) > v^\pi(s)$, then that action outperforms the policy's current behavior at s. This observation suggests a simple strategy for improvement: define a new policy that, at each state, chooses the action with the highest estimated return.

Greedy Policy. We say that a policy π' is *greedy* with respect to q^π if, for each state $s \in \mathcal{S}$, it satisfies:

$$\pi'(s) \in \arg\max_{a\in\mathcal{A}} q^\pi(s,a).$$

This construction selects actions that appear best under the value estimates computed from π. Using the Bellman expectation identity for $q^\pi(s,a)$, we can equivalently write:

$$\pi'(s) \in \arg\max_{a\in\mathcal{A}} \sum_{s',r} p(s',r \mid s,a)\, \left[r + \gamma v^\pi(s')\right].$$

Such greedy policies form the basis of policy improvement algorithms. We now formalize why they lead to improved or equally good performance.

Theorem 3.3 (Policy Improvement)**.** *Let π be a Markov policy, and let π' be a policy greedy with respect to q^π, i.e.,*

$$\pi'(s) \in \arg\max_{a\in\mathcal{A}} q^\pi(s,a) \quad \textit{for all } s \in \mathcal{S}.$$

Then the new policy π' is at least as good as π, formally:

$$v^{\pi'}(s) \geq v^\pi(s), \quad \forall s \in \mathcal{S}.$$

Furthermore, if for some state $s \in \mathcal{S}$, the chosen greedy action satisfies:

$$q^\pi(s,\pi'(s)) > v^\pi(s),$$

then $v^{\pi'}(s) > v^\pi(s)$, making π' strictly better than π.

Proof. The simplest proof of this result follows from Monotone operator theory. We relegate it until, such theory si properly developed next class. □

3.5 Optimality and Bellman Optimality Equations

Optimal Value Functions. The *optimal state-value function* v^* specifies the best achievable return from each state:

$$v^*(s) := \max_{\pi\in\Pi_M} v^\pi(s),$$

and similarly, the *optimal action-value function* is defined as:

$$q^*(s,a) := \max_{\pi\in\Pi_M} q^\pi(s,a).$$

These functions quantify the long-term performance of the best possible behavior an agent can exhibit, assuming it starts in a given state or state-action pair.

Bellman Optimality Equations. The optimal value functions satisfy the following recursive identities, known as the *Bellman optimality equations*:

$$v^*(s) = \max_{a \in \mathcal{A}} \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v^*(s') \right],$$

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

These equations characterize the value of being in a state (or taking an action) assuming optimal behavior thereafter.

Theorem 3.4 (Bellman Optimality Principle)**.** *Let v^* and q^* be the optimal state-value and action-value functions for a finite MDP with discount factor $0 < \gamma < 1$. Then:*

1. *The functions v^* and q^* satisfy the Bellman optimality equations:*

$$v^*(s) = \max_{a \in \mathcal{A}} \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v^*(s') \right],$$

$$q^*(s,a) = \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \max_{a' \in \mathcal{A}} q^*(s', a') \right].$$

2. *Any policy π^* that is greedy with respect to v^* or q^* is optimal; i.e.,*

$$\pi^*(s) \in \arg\max_{a \in \mathcal{A}} q^*(s,a) \quad \Rightarrow \quad v^{\pi^*}(s) = v^*(s) \quad \forall s \in \mathcal{S}.$$

Corollary 3.1 (Existence of Deterministic Optimal Policy)**.** *For any finite MDP with discount factor $0 < \gamma < 1$, there exists a deterministic Markov policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that is optimal. That is, there exists a policy such that:*

$$\pi^*(s) \in \arg\max_{a \in \mathcal{A}} q^*(s,a) \quad \textit{for all } s \in \mathcal{S},$$

and this policy achieves the optimal value function:

$$v^{\pi^*}(s) = v^*(s) \quad \textit{for all } s \in \mathcal{S}.$$

Implications. This result has several important consequences:

- To find an optimal policy, it suffices to find a value function v (or q) that satisfies the Bellman optimality equation.
- Any greedy policy with respect to such a function is guaranteed to be optimal.
- There is no loss of generality in focusing on deterministic, Markov stationary policies when searching for optimal solutions.

This theoretical foundation underlies many practical algorithms, including policy iteration, value iteration, and Q-learning, which we will study in future lectures.

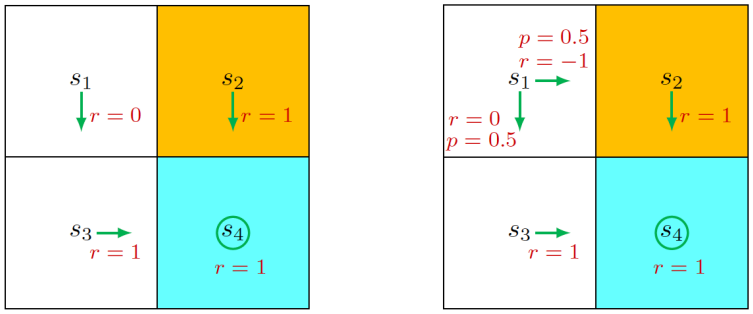
3.6 Running Example: 2×2 Gridworld

We revisit a simplified 2×2 Gridworld to illustrate value functions concretely. The agent starts in one of four states, with transitions governed by a fixed policy. The goal is to reach the terminal state s_4, which yields a reward of $+1$. Other rewards vary depending on the transition.

State-Value Function under Different Policies. We consider two fixed policies:

- **Policy A (Deterministic):** Always move down if possible, otherwise go right if possible.
- **Policy B (Stochastic):** From state s_1, move down or right with equal probability, otherwise follow policy A.

Figure 3.1 illustrates the transition dynamics under each policy.



(a) Deterministic policy (b) Stochastic policy

Figure 3.1: Transition and reward dynamics under two different policies.

Computed State Values under π_{det}. Computing the value function v^{π}_{det} is simple in this case; we assume $\gamma = 0.9$. Starting from s_4:

$$v^{\pi}(s_4) = \mathbb{E}_{\pi}[G_t \mid S_t = s_4] = \sum_{k=0}^{\infty} \gamma^k \cdot 1 = \frac{1}{1 - 0.9} = 10.$$

Continuing with s_3, note that the policy π_{det} deterministically moves right to s_4, which yields a reward of 1 and transitions to a state with value 10:

$$v^{\pi_{\text{det}}}(s_3) = \mathbb{E}_{\pi_{\text{det}}}[R_{t+1} + \gamma v^{\pi_{\text{det}}}(S_{t+1}) \mid S_t = s_3] = 1 + \gamma v^{\pi_{\text{det}}}(s_4) = 1 + 0.9 \cdot 10 = 10.$$

From s_2, the policy π_{det} moves down to s_4 and receives a reward of 1:

$$v^{\pi_{\text{det}}}(s_2) = \mathbb{E}_{\pi_{\text{det}}}[R_{t+1} + \gamma v^{\pi_{\text{det}}}(S_{t+1}) \mid S_t = s_2] = 1 + \gamma v^{\pi_{\text{det}}}(s_4) = 1 + 0.9 \cdot 10 = 10.$$

From s_1, the deterministic policy π_{det} moves down to s_2, where we already know the value is 10. The immediate reward is 0:

$$v^{\pi_{\text{det}}}(s_1) = \mathbb{E}_{\pi_{\text{det}}}[R_{t+1} + \gamma v^{\pi_{\text{det}}}(S_{t+1}) \mid S_t = s_1] = 0 + \gamma v^{\pi_{\text{det}}}(s_2) = 0.9 \cdot 10 = 9.$$

Computed State Values under π_{stoch}. For the stochastic policy π_{stoch}, the transitions from s_2, s_3, and s_4 are the same as in the deterministic case:

$$v^{\pi_{\text{stoch}}}(s_4) = 10, \quad v^{\pi_{\text{stoch}}}(s_3) = 10, \quad v^{\pi_{\text{stoch}}}(s_2) = 10.$$

At s_1, the policy chooses between: - going right to s_3 with reward -1, and - going down to s_2 with reward 0, each with probability 0.5. Thus:

$$\begin{aligned} v^{\pi_{\text{stoch}}}(s_1) &= 0.5 \cdot (-1 + \gamma \cdot v(s_3)) + 0.5 \cdot (0 + \gamma \cdot v(s_2)) \\ &= 0.5 \cdot (-1 + 9) + 0.5 \cdot (9) = 4 + 4.5 = 8.5. \end{aligned}$$

These values are consistent with Table 3.1.

The table below shows the computed state-value function $v^{\pi}(s)$ for each policy, assuming $\gamma = 0.9$.

State	s_1	s_2	s_3	s_4
$v^{\pi_{\text{det}}}(s)$	9	10	10	10
$v^{\pi_{\text{stoch}}}(s)$	8.5	10	10	10

Table 3.1: State-value function $v^\pi(s)$ under two policies.

Action Values and Policy Improvement. To illustrate the usefulness of action-values $q^\pi(s, a)$, consider the stochastic policy π_{stoch} at state s_1. The two available actions are:

- **down** $\to s_2$ with reward 0,
- **right** $\to s_3$ with reward -1.

We compute:

$$q^{\pi_{\text{stoch}}}(s_1, \text{down}) = 0 + \gamma \cdot v^{\pi_{\text{stoch}}}(s_3) = 0.9 \cdot 10 = 9.0,$$

$$q^{\pi_{\text{stoch}}}(s_1, \text{right}) = -1 + \gamma \cdot v^{\pi_{\text{stoch}}}(s_2) = -1 + 0.9 \cdot 10 = 8.0.$$

The action "down" yields a higher expected return. This suggests that a *greedy policy improvement* step would prefer "down" deterministically—matching the behavior of the deterministic policy. This illustrates how q^π helps identify better actions even when starting from a suboptimal policy. This simple gridworld demonstrates how value functions propagate rewards, and how action-values reveal suboptimalities in policies—central concepts in reinforcement learning.