# Finite-Horizon Policy Evaluation

Reinforcement Learning

**Sai Sampath Kedari**
sampath@umich.edu

## Contents

**1 Setup** **2**
1.1 Finite-Horizon Model . . . 2
1.2 Histories . . . 2
1.3 Random Variables . . . 2
1.4 Policies (HR) . . . 2
1.5 Controlled Markov Property . . . 3

**2 The Value Function and Policy Evaluation** **3**
2.1 Value Function . . . 3
2.2 Policy Evaluation . . . 3

**3 Bellman Policy-Evaluation Recursion for HR Policies** **4**
3.1 Proof . . . 4
3.2 Policy Evaluation Algorithm (HR) . . . 6

**4 Specialization to History-Dependent Deterministic Policies (HR → HD)** **6**
4.1 Setting . . . 6
4.2 Derivation from the HR recursion . . . 6
4.3 Remark . . . 7

**5 Specialization to Markovian Policies** **7**
5.1 Randomized Markovian Policies (HR → MR) . . . 7
5.2 Deterministic Markovian Policies (MR → MD) . . . 7
5.3 Summary of Structural Reductions . . . 8

**6 Computational Cost of Policy Evaluation** **8**
6.1 History-Dependent Policies (HR / HD) . . . 8
6.2 Markovian Policies (MR / MD) . . . 9
6.3 Summary . . . 9

# 1 Setup

We consider a finite-horizon Markov decision process (MDP) as defined in Puterman (Ch. 2, Ch. 4). All basic probabilistic constructions and measurability issues are assumed established in Documents 1 and 2.

## 1.1 Finite-Horizon Model

The planning horizon is $N < \infty$. Decision epochs are $t = 1, 2, \ldots, N$, with actions chosen at epochs $t = 1, \ldots, N-1$. The state space $S$ is finite. For each $s \in S$, the set $A_s$ denotes the finite set of feasible actions in state $s$.

For each $t = 1, \ldots, N-1$, $s \in S$, and $a \in A_s$,

$$p_t(j \mid s, a), \qquad j \in S,$$

denotes the transition probability to the next state. Stage rewards are $r_t(s, a)$ for $t = 1, \ldots, N-1$, and the terminal reward is $r_N(s)$.

## 1.2 Histories

At decision epoch $t$, the history is

$$h_t = (s_1, a_1, s_2, \ldots, a_{t-1}, s_t) \in H_t := (S \times A)^{t-1} \times S.$$

We adopt the convention that when the history is $h_t$, the symbol $s_t$ denotes the current state embedded in $h_t$. In particular, conditioning on $H_t = h_t$ determines $X_t = s_t$.

## 1.3 Random Variables

Let

$$X_t = \text{state at epoch } t, \qquad Y_t = \text{action chosen at epoch } t,$$

and define the history process recursively by

$$H_1 = X_1, \qquad H_t = (H_{t-1}, Y_{t-1}, X_t), \quad t \geq 2.$$

## 1.4 Policies (HR)

A randomized history-dependent (HR) policy is a sequence

$$\pi = (d_1, \ldots, d_{N-1}),$$

where each decision rule maps histories to action distributions:

$$d_t : H_t \to \mathcal{P}(A_{s_t}).$$

We write

$$q_{d_t(h_t)}(a) := \mathbb{P}^\pi(Y_t = a \mid H_t = h_t), \qquad a \in A_{s_t}.$$

### 1.5 Controlled Markov Property

Under the MDP model,

$$\mathbb{P}^{\pi}(X_{t+1} = j \mid H_t = h_t,\, Y_t = a) = p_t(j \mid s_t, a),$$

for all histories $h_t$ with current state $s_t$, actions $a \in A_{s_t}$, and $j \in S$.

The immediate reward $r_t(s_t, a)$ is a deterministic function of the current state and action.

## 2 The Value Function and Policy Evaluation

Fix a randomized history-dependent policy

$$\pi = (d_1, d_2, \ldots, d_{N-1}) \in \Pi^{\mathrm{HR}}.$$

### 2.1 Value Function

Fix a policy $\pi \in \Pi^{\mathrm{HR}}$. For each decision epoch $t = 1, 2, \ldots, N$ and each history $h_t \in H_t$, the *value function* $u_t^{\pi}(h_t)$ represents the expected total reward that will be obtained *from decision epoch* $t$ *onward*, given that the system has reached history $h_t$ at epoch $t$ and that policy $\pi$ is followed thereafter.

Formally, for $t < N$,

$$u_t^{\pi}(h_t) := \mathbb{E}^{\pi}\left[\sum_{n=t}^{N-1} r_n(X_n, Y_n) + r_N(X_N) \,\middle|\, H_t = h_t\right],$$

and at the terminal epoch,

$$u_N^{\pi}(h_N) := r_N(s_N),$$

where $s_N$ is the terminal state embedded in $h_N$.

Thus, $u_t^{\pi}(h_t)$ is a *continuation value*: it accounts only for rewards that are yet to be realized from epoch $t$ onward and does not include any rewards accumulated prior to reaching $h_t$.

When $t = 1$ and $h_1 = s$, this coincides with Puterman's notation $v_N^{\pi}(s)$, where the subscript denotes the horizon length rather than the starting epoch.

### 2.2 Policy Evaluation

The *policy evaluation problem* consists of computing the value function $\{u_t^{\pi}\}_{t=1}^{N}$ for a fixed policy $\pi$.

In particular, the quantity of interest is

$$u_1^{\pi}(s),$$

the expected total reward obtained when the process starts in state $s$ at epoch 1 and policy $\pi$ is followed throughout the horizon.

A direct computation would require enumerating all possible future state–action trajectories and averaging the associated rewards, which is computationally infeasible. Dynamic programming overcomes this difficulty by exploiting the temporal structure of the problem and computing the value functions recursively, working backward from the terminal epoch.

The next theorem derives the backward recursion that enables policy evaluation.

## 3 Bellman Policy-Evaluation Recursion for HR Policies

**Theorem 3.1** (Bellman policy-evaluation recursion for HR policies)**.** *Let $\pi = (d_1, d_2, \ldots, d_{N-1}) \in \Pi^{\mathrm{HR}}$ be a randomized history-dependent policy. Then, for every $t = 1, 2, \ldots, N-1$ and every history $h_t \in H_t$ with current state $s_t$,*

$$u_t^\pi(h_t) = \sum_{a \in A_{s_t}} q_{d_t(h_t)}(a) \left( r_t(s_t, a) + \sum_{j \in S} p_t(j \mid s_t, a)\, u_{t+1}^\pi(h_t, a, j) \right),$$

*with terminal condition*

$$u_N^\pi(h_N) = r_N(s_N).$$

*Here, $(h_t, a, j) \in H_{t+1}$ denotes the history at epoch $t+1$ formed by appending the action $a$ and the next state $j$ to the history $h_t$.*

### 3.1 Proof

We repeatedly use the law of total expectation for discrete random variables, stated here for convenience:

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]] = \sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y)\, \mathbb{E}[X \mid Y = y].$$

The same identity holds under conditional expectation.

Fix a decision epoch $t \leq N-1$ and a history $h_t \in H_t$. Let $s_t$ denote the current state embedded in $h_t$.

By definition of the value function,

$$u_t^\pi(h_t) = \mathbb{E}^\pi \left[ \sum_{n=t}^{N-1} r_n(X_n, Y_n) + r_N(X_N) \,\middle|\, H_t = h_t \right].$$

**Step 1: Condition on the action $Y_t$.**

Given $H_t = h_t$, the action $Y_t$ takes values in $A_{s_t}$. Applying the law of total expectation,

$$u_t^\pi(h_t) = \sum_{a \in A_{s_t}} \mathbb{P}^\pi(Y_t = a \mid H_t = h_t)\, \mathbb{E}^\pi \left[ \sum_{n=t}^{N-1} r_n(X_n, Y_n) + r_N(X_N) \,\middle|\, H_t = h_t,\, Y_t = a \right].$$

Under policy $\pi$,

$$\mathbb{P}^\pi(Y_t = a \mid H_t = h_t) = q_{d_t(h_t)}(a).$$

**Step 2: Expand and regroup the return.**

Expand the return:

$$\sum_{n=t}^{N-1} r_n(X_n, Y_n) + r_N(X_N) = r_t(X_t, Y_t) + \sum_{n=t+1}^{N-1} r_n(X_n, Y_n) + r_N(X_N).$$

Define

$$G_{t+1} = \sum_{n=t+1}^{N-1} r_n(X_n, Y_n) + r_N(X_N).$$

Thus,

$$G_t = r_t(X_t, Y_t) + G_{t+1}.$$

**Step 3: Evaluate the immediate reward term.**

Once $H_t = h_t$ and $Y_t = a$ are fixed, the state is $X_t = s_t$. Therefore $r_t(X_t, Y_t) = r_t(s_t, a)$ is deterministic, and

$$\mathbb{E}^\pi[r_t(X_t, Y_t) \mid H_t = h_t,\, Y_t = a] = r_t(s_t, a).$$

**Step 4: Condition on the next state $X_{t+1}$.**

Conditioned on $H_t = h_t$ and $Y_t = a$, the next state $X_{t+1}$ takes values in $S$. Applying the law of total expectation,

$$\mathbb{E}^\pi[G_{t+1} \mid H_t = h_t,\, Y_t = a] = \sum_{j \in S} \mathbb{P}^\pi(X_{t+1} = j \mid H_t = h_t,\, Y_t = a)\, \mathbb{E}^\pi[G_{t+1} \mid H_t = h_t,\, Y_t = a,\, X_{t+1} = j].$$

By the controlled Markov property,

$$\mathbb{P}^\pi(X_{t+1} = j \mid H_t = h_t,\, Y_t = a) = p_t(j \mid s_t, a).$$

If $H_t = h_t$, $Y_t = a$, and $X_{t+1} = j$, then the next history is

$$h_{t+1} = (h_t, a, j).$$

By definition of the value function,

$$\mathbb{E}^\pi[G_{t+1} \mid H_t = h_t,\, Y_t = a,\, X_{t+1} = j] = u^\pi_{t+1}(h_t, a, j).$$

Hence,

$$\mathbb{E}^\pi[G_{t+1} \mid H_t = h_t,\, Y_t = a] = \sum_{j \in S} p_t(j \mid s_t, a)\, u^\pi_{t+1}(h_t, a, j).$$

**Step 5: Assemble the recursion.**

Substituting the above results,

$$u^\pi_t(h_t) = \sum_{a \in A_{s_t}} q_{d_t(h_t)}(a) \left[ r_t(s_t, a) + \sum_{j \in S} p_t(j \mid s_t, a)\, u^\pi_{t+1}(h_t, a, j) \right].$$

At the terminal epoch,

$$u^\pi_N(h_N) = r_N(s_N).$$

**Proof technique summary.** The proof relies on three ingredients: (i) additivity of the return and linearity of conditional expectation, (ii) the law of total expectation applied successively to $Y_t$ and $X_{t+1}$, and (iii) the controlled Markov property of the MDP. No additional assumptions or "Bellman magic" are required. □

### 3.2 Policy Evaluation Algorithm (HR)

Theorem 3.1 yields a backward-induction algorithm for computing $u_1^\pi(s)$ for all $s \in S$.

Finite-Horizon Policy Evaluation Algorithm (HR Policies)

1. Set $t = N$ and define $u_N^\pi(h_N) = r_N(s_N)$ for all $h_N \in H_N$.
2. If $t = 1$, stop. Otherwise, decrement $t$ by one.
3. For each $h_t \in H_t$, compute

$$u_t^\pi(h_t) = \sum_{a \in A_{s_t}} q_{d_t(h_t)}(a) \left( r_t(s_t, a) + \sum_{j \in S} p_t(j \mid s_t, a)\, u_{t+1}^\pi(h_t, a, j) \right).$$

4. Return to step 2.

Because the number of histories grows exponentially with the horizon, this algorithm is generally infeasible for large $N$ and serves primarily as a theoretical foundation for more structured policy classes considered in subsequent sections.

## 4 Specialization to History-Dependent Deterministic Policies (HR → HD)

A history-dependent deterministic (HD) policy is a special case of a randomized history-dependent (HR) policy in which each decision rule selects a single action with probability one.

### 4.1 Setting

Let $\pi = (d_1, d_2, \dots, d_{N-1}) \in \Pi^{\mathrm{HD}}$ be a deterministic history-dependent policy. Each decision rule

$$d_t : H_t \to A$$

assigns a unique feasible action at each history:

$$d_t(h_t) \in A_{s_t}, \qquad h_t \in H_t.$$

Equivalently, the induced action distribution at history $h_t$ is degenerate:

$$q_{d_t(h_t)}(a) = \mathbf{1}\{a = d_t(h_t)\}, \qquad a \in A_{s_t}.$$

### 4.2 Derivation from the HR recursion

Start from the Bellman policy-evaluation recursion for HR policies:

$$u_t^\pi(h_t) = \sum_{a \in A_{s_t}} q_{d_t(h_t)}(a) \left( r_t(s_t, a) + \sum_{j \in S} p_t(j \mid s_t, a)\, u_{t+1}^\pi(h_t, a, j) \right).$$

Under an HD policy, the action distribution assigns probability one to $a = d_t(h_t)$ and zero to all other actions. Therefore, the sum over actions collapses to the single term corresponding to $d_t(h_t)$:

$$u_t^\pi(h_t) = r_t\big(s_t, d_t(h_t)\big) + \sum_{j \in S} p_t\big(j \mid s_t, d_t(h_t)\big)\, u_{t+1}^\pi\big(h_t, d_t(h_t), j\big).$$

**Corollary 4.0.1** (Policy evaluation for HD policies)**.** *Let $\pi \in \Pi^{\mathrm{HD}}$. Then for all $t = 1, \ldots, N-1$ and all $h_t \in H_t$ with current state $s_t$,*

$$\boxed{u_t^\pi(h_t) = r_t\big(s_t, d_t(h_t)\big) + \sum_{j \in S} p_t\big(j \mid s_t, d_t(h_t)\big)\, u_{t+1}^\pi\big(h_t, d_t(h_t), j\big),}$$

*with terminal condition $u_N^\pi(h_N) = r_N(s_N)$.*

### 4.3 Remark

Relative to the HR recursion, the only change is that the expectation over actions is removed. The value function remains history-dependent, but the action taken at each history is fixed rather than randomized.

## 5 Specialization to Markovian Policies

### 5.1 Randomized Markovian Policies (HR → MR)

**Assumption (Markovian decision rule).** Let $\pi = (d_1, \ldots, d_{N-1}) \in \Pi^{\mathrm{MR}}$ be a randomized Markovian policy. Each decision rule depends only on the current state:

$$q_{d_t(h_t)}(a) = q_{d_t(s_t)}(a), \qquad a \in A_{s_t}.$$

**Consequence (state-based value function).** Under this assumption, the value function depends on the history $h_t$ only through the current state $s_t$. Hence there exists a function $v_t^\pi : S \to \mathbb{R}$ such that

$$u_t^\pi(h_t) = v_t^\pi(s_t), \qquad h_t \in H_t.$$

**Policy-evaluation recursion (MR).** Substituting this restriction into the HR recursion yields, for $t = 1, \ldots, N-1$,

$$\boxed{v_t^\pi(s) = \sum_{a \in A_s} q_{d_t(s)}(a) \left( r_t(s,a) + \sum_{j \in S} p_t(j \mid s, a)\, v_{t+1}^\pi(j) \right),}$$

with terminal condition $v_N^\pi(s) = r_N(s)$.

### 5.2 Deterministic Markovian Policies (MR → MD)

**Assumption (deterministic decision rule).** Let $\pi = (d_1, \ldots, d_{N-1}) \in \Pi^{\mathrm{MD}}$, where each

$$d_t : S \to A_s$$

selects a single action. The induced action distribution is degenerate:

$$q_{d_t(s)}(a) = \mathbf{1}\{a = d_t(s)\}.$$

**Policy-evaluation recursion (MD).** Substituting this into the MR recursion removes the expectation over actions:

$$\boxed{v_t^\pi(s) = r_t\big(s, d_t(s)\big) + \sum_{j\in S} p_t\big(j \mid s, d_t(s)\big)\, v_{t+1}^\pi(j),}$$

for $t = 1, \dots, N-1$, with $v_N^\pi(s) = r_N(s)$.

### 5.3 Summary of Structural Reductions

| Policy class | Policy dependence | Value function domain | Bellman recursion structure |
|---|---|---|---|
| HR | full history $h_t$ | $u_t : H_t \to \mathbb{R}$ | $\sum_a q(a \mid h_t)\left[r_t + \sum_j p(\cdot)\, u_{t+1}(h_t, a, j)\right]$ |
| HD | full history $h_t$ | $u_t : H_t \to \mathbb{R}$ | $r_t + \sum_j p(\cdot)\, u_{t+1}(h_t, d_t(h_t), j)$ |
| MR | current state $s_t$ | $v_t : S \to \mathbb{R}$ | $\sum_a q(a \mid s_t)\left[r_t + \sum_j p(\cdot)\, v_{t+1}(j)\right]$ |
| MD | current state $s_t$ | $v_t : S \to \mathbb{R}$ | $r_t + \sum_j p(\cdot)\, v_{t+1}(j)$ |

## 6 Computational Cost of Policy Evaluation

The specialization hierarchy developed above

$$\text{HR} \;\to\; \text{HD} \;\to\; \text{MR} \;\to\; \text{MD}$$

is motivated not only by structural simplicity but by a fundamental reduction in computational cost.

Let $K = |S|$ denote the number of states and

$$L := \max_{s\in S} |A_s|$$

the maximum number of feasible actions per state.

### 6.1 History-Dependent Policies (HR / HD)

At decision epoch $t$, the number of possible histories is

$$|H_t| = K^t L^{t-1}.$$

For each history $h_t$, evaluating the Bellman recursion requires a sum over $K$ possible next states. Therefore, policy evaluation at epoch $t$ requires

$$O\big(K^{t+1} L^{t-1}\big)$$

operations.

Summing over epochs $t = 1, \dots, N-1$, the total cost is

$$\sum_{t=1}^{N-1} O\big(K^{t+1}L^{t-1}\big) = O\left(K \sum_{t=1}^{N-1} (KL)^t\right),$$

which is exponential in the horizon length $N$.

## 6.2 Markovian Policies (MR / MD)

Under a Markovian policy, the value function depends only on the current state. At each epoch, only $K$ states must be evaluated.

**Randomized Markovian (MR).** For each state, the recursion involves a sum over $L$ actions and $K$ next states. Thus, the cost per epoch is

$$O(K^2L),$$

and the total cost over the horizon is

$$O((N-1)K^2L).$$

**Deterministic Markovian (MD).** When the policy is deterministic, the expectation over actions disappears. Each state requires only a sum over $K$ next states, yielding

$$O(K^2)$$

operations per epoch and

$$O((N-1)K^2)$$

operations in total.

## 6.3 Summary

| Policy class | Evaluations per epoch | Total cost |
|---|---|---|
| HR / HD | $\lvert H_t \rvert = K^tL^{t-1}$ histories | exponential in $N$ |
| MR | $K$ states × $L$ actions | $O((N-1)K^2L)$ |
| MD | $K$ states × 1 action | $O((N-1)K^2)$ |

The exponential-to-polynomial reduction in computational complexity is the primary practical justification for restricting attention to Markovian policies and underlies the importance of the state-based value function.