

Q-Learning - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs	1
1.1 Introduction	1
1.2 Data and Notation	1
1.3 Model Formulation: Bellman Equations	2
1.4 Bellman Operators and Contraction	2
1.5 From Dynamic Programming to Stochastic Approximation	2
1.6 Convergence (Sketch)	3
1.7 Control and Exploration	3
1.8 Algorithm (Tabular Q-Learning)	4
1.9 Variants and Remarks	4
1.10 Summary of Variables and Their Dimensions	4
1.11 Summary	5

1 Mathematical Derivations & Proofs

1.1 Introduction

Q-Learning is an *off-policy*, model-free reinforcement learning algorithm that computes the unique fixed point of the Bellman *optimality* operator for the action-value function. It does so by stochastic approximation: at each time step it updates a single (\mathbf{s}, \mathbf{a}) entry of a table \mathbf{Q} toward a sample of the Bellman optimality target. Under suitable conditions (finite Markov Decision Process (MDP), sufficient exploration, diminishing step sizes), Q-Learning converges almost surely to the optimal action-value function q^* , and the greedy policy w.r.t. \mathbf{Q} is optimal.

1.2 Data and Notation

We consider a finite Markov Decision Process (MDP)

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{P}, \mathbf{r}, \gamma),$$

where: \mathcal{S} is a finite state set, $|\mathcal{S}| = S$. Each state $\mathbf{s} \in \mathbb{R}^{n_s}$ is a column vector of dimension $n_s \times 1$. \mathcal{A} is a finite action set, $|\mathcal{A}| = A$. Each action $\mathbf{a} \in \mathbb{R}^{n_a}$ is a column vector of dimension $n_a \times 1$. (For discrete actions, \mathcal{A} is a finite set.) $\mathbf{P}(\mathbf{s}' | \mathbf{s}, \mathbf{a})$ is the transition kernel (row-stochastic), i.e. $\sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) = 1$ for each (\mathbf{s}, \mathbf{a}) , $\mathbf{r}(\mathbf{s}, \mathbf{a}) \triangleq \mathbb{E}[R_{t+1} | S_t = \mathbf{s}, A_t = \mathbf{a}]$ is the expected immediate reward, and $\gamma \in [0, 1]$ is the discount factor.

A (stationary) policy π maps states to action distributions: $\pi(\mathbf{a} | \mathbf{s}) \in [0, 1]$, $\sum_a \pi(\mathbf{a} | \mathbf{s}) = 1$. The (random) return from time t is

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

The state-value and action-value functions under π are

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

Collect the action-values into a table $\mathbf{Q} \in \mathbb{R}^{S \times A}$ with entries $\mathbf{Q}[\mathbf{s}, \mathbf{a}] \approx q^\pi(\mathbf{s}, \mathbf{a})$ (or q^* , depending on context).

1.3 Model Formulation: Bellman Equations

For any policy π , q^π satisfies the *Bellman expectation equation*

$$q^\pi(\mathbf{s}, \mathbf{a}) = \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \sum_{\mathbf{a}'} \pi(\mathbf{a}' \mid \mathbf{s}') q^\pi(\mathbf{s}', \mathbf{a}'). \quad (1)$$

The *optimal* action-value function

$$q^*(\mathbf{s}, \mathbf{a}) \triangleq \max_\pi q^\pi(\mathbf{s}, \mathbf{a})$$

solves the *Bellman optimality equation*

$$q^*(\mathbf{s}, \mathbf{a}) = \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \max_{\mathbf{a}'} q^*(\mathbf{s}', \mathbf{a}'). \quad (2)$$

An optimal deterministic policy is obtained by greedy selection $\pi^*(\mathbf{s}) \in \arg \max_{\mathbf{a}} q^*(\mathbf{s}, \mathbf{a})$.

1.4 Bellman Operators and Contraction

Define the Bellman operators $(T^\pi, T^*) : \mathbb{R}^{S \times A} \rightarrow \mathbb{R}^{S \times A}$ by

$$(T^\pi Q)(\mathbf{s}, \mathbf{a}) \triangleq \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \sum_{\mathbf{a}'} \pi(\mathbf{a}' \mid \mathbf{s}') Q(\mathbf{s}', \mathbf{a}'), \quad (3)$$

$$(T^* Q)(\mathbf{s}, \mathbf{a}) \triangleq \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}'). \quad (4)$$

Equip $\mathbb{R}^{S \times A}$ with the sup norm $\|Q\|_\infty = \max_{\mathbf{s}, \mathbf{a}} |Q(\mathbf{s}, \mathbf{a})|$.

Contraction property. For any Q_1, Q_2 ,

$$\begin{aligned} |(T^* Q_1)(\mathbf{s}, \mathbf{a}) - (T^* Q_2)(\mathbf{s}, \mathbf{a})| &= \gamma \left| \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \left(\max_{\mathbf{a}'} Q_1(\mathbf{s}', \mathbf{a}') - \max_{\mathbf{a}'} Q_2(\mathbf{s}', \mathbf{a}') \right) \right| \\ &\leq \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) \max_{\mathbf{a}'} |Q_1(\mathbf{s}', \mathbf{a}') - Q_2(\mathbf{s}', \mathbf{a}')| \\ &\leq \gamma \|Q_1 - Q_2\|_\infty. \end{aligned}$$

Taking the maximum over (\mathbf{s}, \mathbf{a}) yields $\|T^* Q_1 - T^* Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$. Thus T^* is a γ -contraction.

Existence/uniqueness of the fixed point. By the Banach fixed-point theorem, T^* admits a unique fixed point Q^* and, for any Q , the synchronous iteration $Q_{k+1} = T^* Q_k$ converges to Q^* at rate $O(\gamma^k)$.

1.5 From Dynamic Programming to Stochastic Approximation

The DP update $Q \leftarrow T^* Q$ is not directly implementable without \mathbf{P}, \mathbf{r} . However, a *sample backup* at time t with observed tuple $(S_t, A_t, R_{t+1}, S_{t+1})$ provides the random target

$$Y_t \triangleq R_{t+1} + \gamma \max_{\mathbf{a}'} Q(S_{t+1}, \mathbf{a}').$$

Conditional on $(S_t = \mathbf{s}, A_t = \mathbf{a})$ and current Q , its expectation equals the Bellman optimality update:

$$\mathbb{E}[Y_t | S_t = \mathbf{s}, A_t = \mathbf{a}] = \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \sum_{\mathbf{s}'} \mathbf{P}(\mathbf{s}' | \mathbf{s}, \mathbf{a}) \max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}') = (T^*Q)(\mathbf{s}, \mathbf{a}). \quad (5)$$

Therefore a Robbins–Monro-type stochastic approximation to the T^* fixed point is:

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha_t(S_t, A_t) \left[R_{t+1} + \gamma \max_{\mathbf{a}'} Q_t(S_{t+1}, \mathbf{a}') - Q_t(S_t, A_t) \right], \quad (6)$$

leaving all other entries unchanged. The bracketed term is the *temporal-difference (TD) error*

$$\delta_t = R_{t+1} + \gamma \max_{\mathbf{a}'} Q_t(S_{t+1}, \mathbf{a}') - Q_t(S_t, A_t).$$

Equation (6) is precisely the tabular Q-Learning update.

1.6 Convergence (Sketch)

Assume:

1. Finite MDP; bounded rewards $|R_{t+1}| \leq R_{\max} < \infty$.
2. Every state-action pair is visited infinitely often (e.g., by an ε -greedy exploration with $\varepsilon_t > 0$ and $\sum_t \varepsilon_t = \infty$).
3. Step sizes satisfy Robbins–Monro conditions: $\alpha_t(\mathbf{s}, \mathbf{a}) \in (0, 1]$, $\sum_t \alpha_t(\mathbf{s}, \mathbf{a}) = \infty$ and $\sum_t \alpha_t(\mathbf{s}, \mathbf{a})^2 < \infty$ for all (\mathbf{s}, \mathbf{a}) .

Define the asynchronous operator $H_t(Q)(\mathbf{s}, \mathbf{a}) = \begin{cases} (T^*Q)(\mathbf{s}, \mathbf{a}), & (\mathbf{s}, \mathbf{a}) = (S_t, A_t), \\ Q(\mathbf{s}, \mathbf{a}), & \text{else.} \end{cases}$ Then Eqn. (6) can be

written as

$$Q_{t+1} = Q_t + \alpha_t(H_t(Q_t) - Q_t + M_{t+1}),$$

where (M_{t+1}) is a martingale-difference noise sequence induced by sampling and α_t is a diagonal matrix inserting $\alpha_t(S_t, A_t)$ in the visited component. Because T^* is a contraction, the associated ODE $\dot{Q} = T^*Q - Q$ is globally asymptotically stable with unique equilibrium Q^* . Standard stochastic approximation theory (e.g., Robbins–Monro/Borkar–Meyn) then implies $Q_t \rightarrow Q^*$ almost surely.

Key ingredients. (i) *Contraction*: T^* is a γ -contraction (sup norm). (ii) *Unbiasedness*: Eqn. (5) ensures the noise is a martingale difference with bounded variance. (iii) *Sufficient excitation*: infinite visits guarantee each component is updated infinitely often. Combining (i)–(iii) yields a.s. convergence.

1.7 Control and Exploration

Q-Learning is *off-policy*: the target uses $\max_{\mathbf{a}'} Q(\cdot, \mathbf{a}')$ irrespective of the behavior policy. In practice, one uses an ε -greedy behavior policy

$$\pi_t(\mathbf{a} | \mathbf{s}) = \begin{cases} 1 - \varepsilon_t + \varepsilon_t/A, & \mathbf{a} \in \arg \max_{\mathbf{a}'} Q_t(\mathbf{s}, \mathbf{a}'), \\ \varepsilon_t/A, & \text{otherwise,} \end{cases}$$

with $\varepsilon_t \downarrow 0$ slowly to ensure persistent exploration early and greedy exploitation asymptotically.

1.8 Algorithm (Tabular Q-Learning)

1. **Input:** discount $\gamma \in [0, 1]$; step-size schedule $\alpha_t(\mathbf{s}, \mathbf{a})$; exploration schedule ε_t .
2. **Initialize:** $Q_0(\mathbf{s}, \mathbf{a})$ arbitrarily (e.g., zeros) for all (\mathbf{s}, \mathbf{a}) .
3. **For episodes** $e = 1, 2, \dots$:
 - (a) Initialize S_0 .
 - (b) For $t = 0, 1, 2, \dots$ until termination:
 - i. Select A_t by ε_t -greedy w.r.t. $Q_t(S_t, \cdot)$.
 - ii. Observe R_{t+1} and S_{t+1} .
 - iii. Update
$$Q_{t+1}(S_t, A_t) \leftarrow Q_t(S_t, A_t) + \alpha_t(S_t, A_t) \left(R_{t+1} + \gamma \max_{\mathbf{a}'} Q_t(S_{t+1}, \mathbf{a}') - Q_t(S_t, A_t) \right).$$
 - iv. $S_t \leftarrow S_{t+1}$.
4. **Output:** greedy policy $\hat{\pi}(\mathbf{s}) \in \arg \max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})$.

1.9 Variants and Remarks

- **Double Q-Learning.** To mitigate positive bias from max, maintain two tables $Q^{(1)}, Q^{(2)}$ and alternate updates using one to select and the other to evaluate:

$$Q^{(1)} \leftarrow Q^{(1)} + \alpha \left(r + \gamma Q^{(2)}(\mathbf{s}', \arg \max_{\mathbf{a}} Q^{(1)}(\mathbf{s}', \mathbf{a})) - Q^{(1)}(\mathbf{s}, \mathbf{a}) \right),$$

and symmetrically for $Q^{(2)}$.

- **On-policy SARSA.** Replace the target $\max_{\mathbf{a}'} Q(\mathbf{s}', \mathbf{a}')$ with $Q(\mathbf{s}', A')$ where A' is the next action taken under the behavior policy (on-policy TD control).
- **Function approximation.** With $Q_{\theta}(\mathbf{s}, \mathbf{a})$ (e.g., neural networks), replace the tabular update by gradient descent on the TD loss $\frac{1}{2} (r + \gamma \max_{\mathbf{a}'} Q_{\theta}(\mathbf{s}', \mathbf{a}') - Q_{\theta}(\mathbf{s}, \mathbf{a}))^2$; stability typically requires target networks and replay (DQN).

1.10 Summary of Variables and Their Dimensions

- \mathcal{S} : finite state set, $|\mathcal{S}| = S$.
- \mathcal{A} : finite action set, $|\mathcal{A}| = A$.
- $\mathbf{P}(s' | \mathbf{s}, \mathbf{a})$: transition probabilities; for each (\mathbf{s}, \mathbf{a}) , a probability vector in \mathbb{R}^S .
- $\mathbf{r}(\mathbf{s}, \mathbf{a}) \in \mathbb{R}$: expected immediate reward (scalar).
- $\gamma \in [0, 1]$: discount factor (scalar).
- $\mathbf{Q} \in \mathbb{R}^{S \times A}$: Q-table, with entries $\mathbf{Q}[\mathbf{s}, \mathbf{a}]$ approximating $q^*(\mathbf{s}, \mathbf{a})$.
- $\alpha_t(\mathbf{s}, \mathbf{a}) \in (0, 1]$: step size at time t for component (\mathbf{s}, \mathbf{a}) (scalar).
- $\varepsilon_t \in [0, 1]$: exploration parameter at time t (scalar).
- $R_{t+1} \in \mathbb{R}$, $S_t \in \mathcal{S}$, $A_t \in \mathcal{A}$: sampled reward, state, and action.

1.11 Summary

Starting from the Bellman optimality equation Eqn. (2), we introduced the optimality operator T^* , proved it is a γ -contraction (ensuring a unique fixed point q^*), and showed that the Q-Learning update Eqn. (6) is a Robbins–Monro stochastic approximation to this fixed point, using unbiased single-sample targets Eqn. (5). Under standard conditions (finite MDP, sufficient exploration, diminishing step sizes), the tabular algorithm converges almost surely to q^* ; the greedy policy w.r.t. the learned \mathbf{Q} is then optimal.