Q-Learning: A Historical Synthesis

Robbins-Monro + Bellman + Epsilon-Greedy

A Journey Through Reinforcement Learning History

Outline

The Problem of Root Finding Under Noise

Classical Problem: Find θ^* such that $M(\theta^*) = 0$

Challenge: We cannot observe $M(\theta)$ directly—only noisy samples

$$Y = M(\theta) + \varepsilon$$

where ε is random noise with $\mathbb{E}[\varepsilon] = 0$.

Context (1951):

- Herbert Robbins & Sutton Monro at UNC Chapel Hill
- Motivated by bioassay problems (finding drug dosage thresholds)
- Published: "A Stochastic Approximation Method" (Annals of Math. Stat.)

The Robbins-Monro Algorithm

The Update Rule:

$$\theta_{n+1} = \theta_n - a_n M(\theta_n, X_n)$$

where $\mathbb{E}[M(\theta, X)] = 0$ has unique solution θ^*.

Key Conditions on Step Sizes:

$$\sum_{n=1}^{\infty} a_n = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} a_n^2 < \infty$$

Example: $a_n = 1/n$ satisfies both conditions.

Theorem (Robbins-Monro 1951): Under mild regularity conditions, $\theta_n \xrightarrow{\text{a.s.}} \theta^*$.

Example: Online Mean Estimation

Problem: Estimate $\theta^* = \mathbb{E}[X]$ from samples $X_1, X_2, \ldots$

Let $M(\theta, X) := \theta - X$. Then $\mathbb{E}[M(\theta, X)] = \theta - \theta^* = 0$ iff $\theta = \theta^*$.

RM update:

$$\theta_{n+1} = \theta_n - a_n(\theta_n - X_n)$$

Rewrite as weighted average:

$$\theta_{n+1} = (1 - a_n)\theta_n + a_n X_n$$

Equivalently: SGD on loss $L(\theta) = \frac{1}{2}\|X - \theta\|^2$ since $\nabla_\theta L = \theta - X$.

Key point: RM does not require L to exist—it only needs $\mathbb{E}[M(\theta, X)] = 0$ at the solution.

(Example adapted from Wikipedia)

Why Robbins-Monro Matters for RL

The Template for Learning from Noisy Feedback:

$$\text{New Estimate} = \text{Old Estimate} + \text{Step Size} \times \text{Error Signal}$$

Intuition:

- Large steps early (explore the space)
- Small steps later (converge precisely)
- Noise averages out over time

This pattern appears everywhere in RL:

- TD-learning
- Q-learning
- Policy gradient methods
- Actor-critic algorithms

Richard Bellman and the Curse of Dimensionality

Richard Bellman (1920–1984):

- RAND Corporation, 1950s
- Coined "dynamic programming" (1953)
- Published *Dynamic Programming* (1957)

The Core Insight:
Optimal decisions have a recursive structure—the *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."*

Deriving Bellman from Random Horizons

Setup: Let $T \sim \text{Geometric}(1-\gamma)$, so $P(T=k) = (1-\gamma)\gamma^k$.

Define the **normalized** value (a true expectation):

$$\tilde{Q}(s,a) = \mathbb{E}\left[\sum_{k=0}^{T} r_k \;\middle|\; s_0 = s, a_0 = a\right] = (1-\gamma)\sum_{k=0}^{\infty} \gamma^k \, \mathbb{E}[r_k]$$

The **standard** Q-function drops the normalizing constant $(1-\gamma)$:

$$Q(s,a) = \sum_{k=0}^{\infty} \gamma^k \, \mathbb{E}[r_k] = \frac{1}{1-\gamma}\tilde{Q}(s,a)$$

Why drop it? The arg max is unchanged, and Q has cleaner recursion.

Unrolling One Step

For $\tilde{Q}$: Using $P(T \geq 1) = \gamma$ and memorylessness:

$$\tilde{Q}(s, a) = (1 - \gamma)R(s, a) + \gamma \cdot \mathbb{E}_{s'} \left[\max_{a'} \tilde{Q}(s', a') \right]$$

For $Q = \tilde{Q}/(1 - \gamma)$: Divide through:

$$Q(s, a) = R(s, a) + \gamma \cdot \mathbb{E}_{s'} \left[\max_{a'} Q(s', a') \right]$$

The $(1 - \gamma)$ cancels, giving the standard Bellman equation.

Key insight: The memoryless property of the geometric distribution yields the recursive structure.

The Bellman Equation

Value Function: $V^*(s)$ = optimal expected cumulative reward from state s

Bellman Optimality Equation:

$$V^*(s) = \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^*(s') \right]$$

Action-Value Form (crucial for Q-learning):

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a')$$

The Q-function: Expected return starting from state s, taking action a, then acting optimally.

The Contraction Property

Bellman Operator:

$$(TQ)(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a')$$

Key Property: T is a contraction in the max-norm:

$$\|TQ_1 - TQ_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

Consequence (Banach Fixed Point Theorem):

- Q^* exists and is unique
- Repeated application $Q \leftarrow TQ$ converges to Q^*
- This is **value iteration**

Problem: Requires knowing $P(s'|s,a)$ and $R(s,a)$—the model!

The Exploration-Exploitation Dilemma

The Fundamental Tradeoff:

- **Exploitation:** Use current knowledge to maximize reward
- **Exploration:** Gather new information for future benefit

Historical Roots:

- Sequential experimental design (1940s–1950s)
- Multi-armed bandit problem (Robbins 1952)
- Thompson Sampling (Thompson 1933, rediscovered later)

Why It Matters:
A purely greedy learner may get stuck in local optima, never discovering better actions.

Epsilon-Greedy: Simple and Effective

The Algorithm:

$$a = \begin{cases} \arg\max_a Q(s,a) & \text{with probability } 1-\varepsilon \\ \text{random action} & \text{with probability } \varepsilon \end{cases}$$

Properties:

- Ensures every action tried infinitely often (given infinite time)
- Simple to implement and analyze
- ε can decay over time: $\varepsilon_t \to 0$

Historical Note:

- No single "inventor"—emerged from operations research
- Closely related to randomized strategies in game theory
- Became standard in RL through Sutton & Barto's work (1980s)

Why Exploration is Necessary for Convergence

For Q-learning to converge, we need:

Every state-action pair (s, a) must be visited infinitely often.

Epsilon-greedy guarantees this:

- Probability of any action in any state $\geq \varepsilon/|A|$
- Under mild reachability assumptions, all (s, a) pairs visited

Alternatives:

- Boltzmann/softmax exploration
- Upper Confidence Bound (UCB)
- Optimism in the face of uncertainty
- Intrinsic motivation / curiosity

Christopher Watkins and Q-Learning

Chris Watkins (1989):

- PhD thesis: "Learning from Delayed Rewards" (Cambridge)
- Published with Dayan (1992): convergence proof

The Breakthrough:
Combine Bellman's equation with Robbins-Monro updates to learn Q^* **without knowing the model**.

The Q-Learning Update:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$

This is Robbins-Monro applied to the Bellman equation!

Dissecting the Q-Learning Update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \underbrace{\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right]}_{\text{Temporal Difference (TD) Error}}$$

Component	**Origin**
α_t (step size)	Robbins-Monro (1951)
$r_t + \gamma \max_{a'} Q(s_{t+1}, a')$	Bellman target (1957)
TD error structure	Stochastic approximation
Exploration policy	Epsilon-greedy / bandits

Key insight: The observed (r_t, s_{t+1}) is a *sample* from the unknown MDP dynamics.

The Convergence Theorem

Theorem (Watkins & Dayan, 1992):
Q-learning converges to Q^* with probability 1, provided:

1. All state-action pairs visited infinitely often
2. Step sizes satisfy: $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$
3. Rewards are bounded

Proof Sketch:

- View Q-learning as stochastic approximation (Robbins-Monro)
- The Bellman operator is a contraction
- Apply general SA convergence theory (e.g., Jaakkola et al. 1994)

Epsilon-greedy ensures condition 1!

The Complete Q-Learning Algorithm

1. Initialize $Q(s, a)$ arbitrarily for all (s, a)
2. For each episode:
   1. Initialize state s
   2. For each step:
      - Choose a from s using ε-greedy w.r.t. Q
      - Take action a, observe r, s'
      - Update:

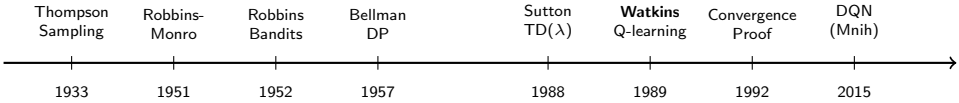
$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

      - $s \leftarrow s'$
   3. Until s is terminal

Three pillars working together:

- Bellman equation provides the learning target
- Robbins-Monro provides the update mechanism
- Epsilon-greedy provides the exploration

Timeline: The Road to Q-Learning



Key Contributors:

- Robbins & Monro: Stochastic approximation framework
- Bellman: Optimality principle and recursive structure
- Sutton: Temporal difference learning bridge
- Watkins: The synthesis into Q-learning

The Legacy: From Q-Learning to Deep RL

The Original Formula Lives On:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

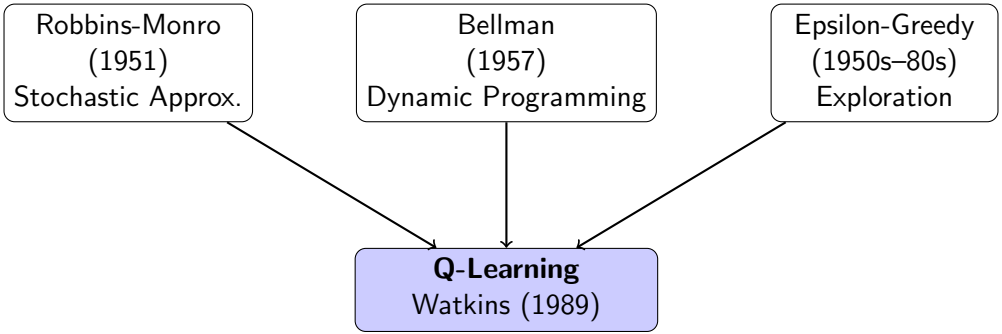
Modern Extensions:

- **DQN (2015):** Q represented by deep neural network
- **Double DQN:** Reduces overestimation bias
- **Dueling DQN:** Separates value and advantage
- **Rainbow:** Combines multiple improvements
- **Distributional RL:** Learn full return distribution

The three pillars remain:

Bellman structure + stochastic approximation + exploration

Summary: Q-Learning as a Synthesis



Q-Learning = Robbins-Monro + Bellman + Epsilon-Greedy

A beautiful synthesis of ideas spanning 40 years of mathematics, operations research, and computer science.

References

- Robbins, H. & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*.
- Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.
- Watkins, C. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge.
- Watkins, C. & Dayan, P. (1992). Q-learning. *Machine Learning*.
- Sutton, R. & Barto, A. (2018). *Reinforcement Learning: An Introduction*. 2nd ed.
- Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*.