Robust Reinforcement Learning: Decision Making with Model Uncertainty

Shaocong Ma

February 7, 2026

University of Maryland
scma0908@umd.edu

Reinforcement Learning

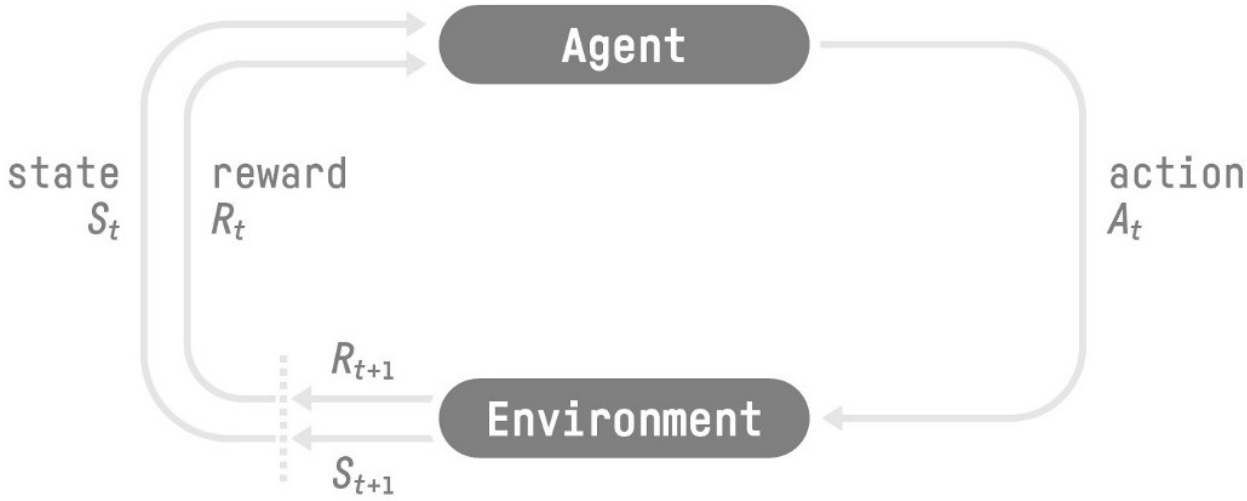

Figure 1: Agent-Environment Interaction[1]

The data in RL (s_t, a_t, r_t, s_{t+1}) generates the *Markov decision process*:

$$s_1, a_1, r_1, s_2, a_2, r_2, \ldots$$

[1] Source: *Reinforcement Learning: An Introduction, Richard Sutton and Andrew G. Barto.*

Reinforcement Learning

Goal: Maximize the expected cumulative reward.

For an agent, the objective is to learn a policy π that maximizes the expected sum of future rewards, typically defined as:

$$V^{\pi}(s) := \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t \middle| s\right]$$

Where:

- r_t is the reward at time step t
- $\gamma \in [0, 1]$ is the discount factor
- T is the time horizon (can be infinite for continuing tasks)

Reinforcement Learning

Greedy Algorithm
Maximize Instant Reward
$\max_a R(s, a)$

RL Algorithm
Maximize Long-Term Reward
$\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$

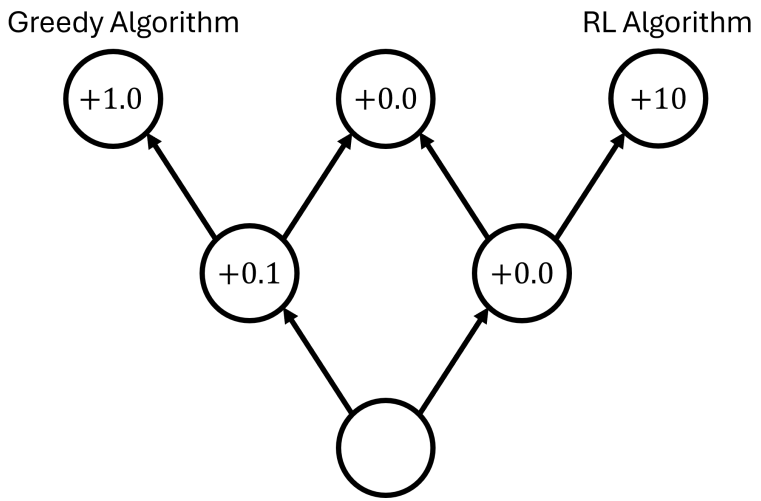

Figure 2: Comparison between a Greedy Algorithm and a RL Algorithm.

Uncertainty in Real World

Real-World Environments Are Inherently Uncertain.

Example 1:

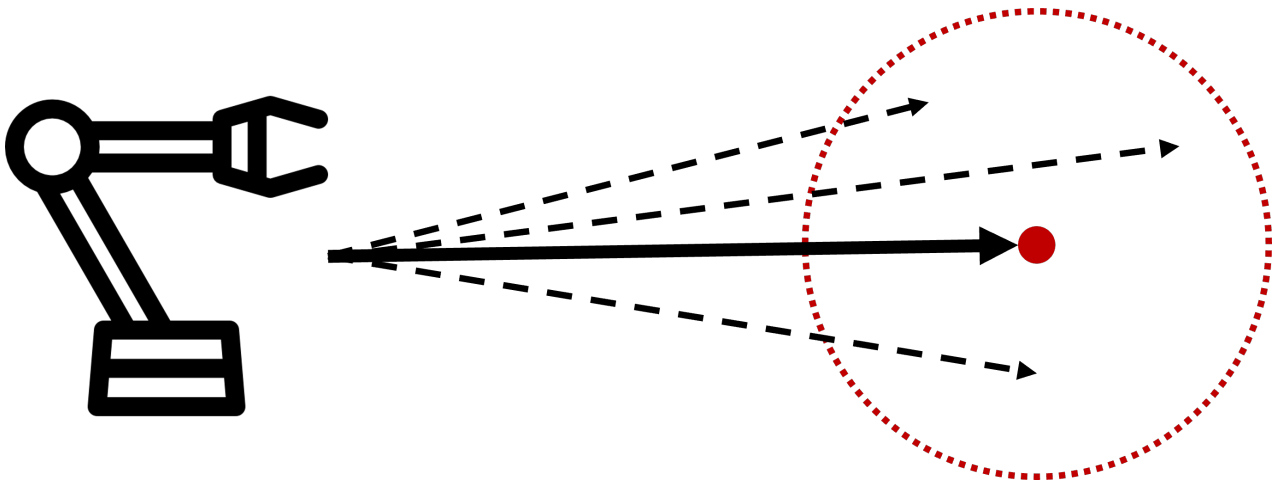

Figure 3: The robot aims to reach the red point. However, due to the inherent uncertainty in the environment, different robots may arrive at any point within the surrounding neighborhood with different probabilities.

Uncertainty in Real World

Example 2:

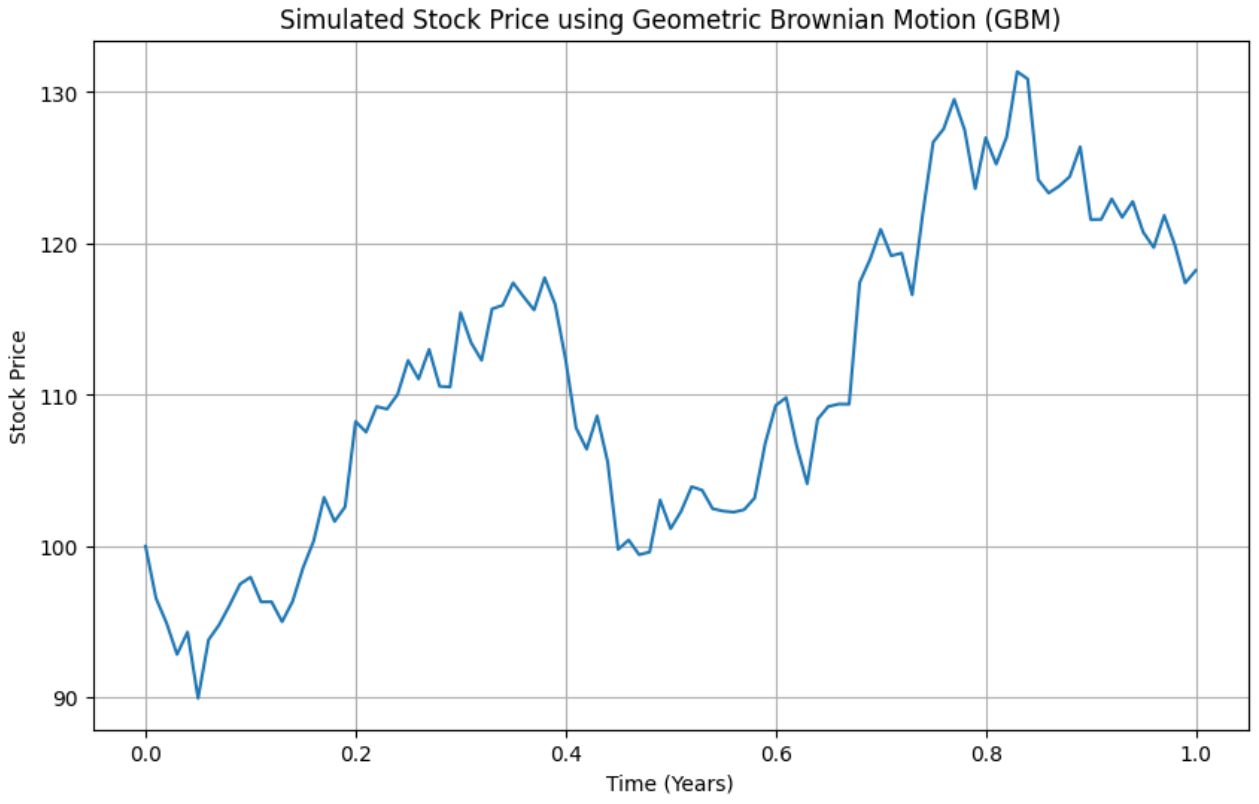

Figure 4: 10x leverage maximizes the expected future return. However, the capital can potentially be wiped out.

Robust Reinforcement Learning

RL Algorithm
Maximize Long-Term Reward

$\max_\pi \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$

Robust RL Algorithm
Maximize Long-Term Reward under the Worst Case

$\max_\pi \min_{\mathbb{P}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$

$\mathbb{P}$ belongs to the uncertainty set, e.g. $\{\mathbb{P} : d(\mathbb{P}, \mathbb{P}_0) \leq \delta\}$.

Example 3:

	Buy Insurance	Not Buy Insurance
Event Occurs (Worst Case)	\$500 (premium) + \$1000 (deductible)	\$10,000 (full loss)
No Event Occurs	\$500 (premium)	\$0 (no loss)

Table 1: Cost comparison between buying and not buying insurance.

Challenges in Robust Reinforcement Learning

RL Algorithm	Robust RL Algorithm
Bellman Operator	Robust Bellman Operator
$\mathcal{T}Q(s,a) =$	$\tilde{\mathcal{T}}Q(s,a) = r(s,a) +$
$r(s,a) + \gamma \max_a E_{s' \sim \mathbb{P}(\cdot \mid s,a)} Q(s',a)$	$\gamma \max_a \min_{\mathbb{P}} E_{s' \sim \mathbb{P}(\cdot \mid s,a)} Q(s',a)$

- Policy evaluation:
  - Common approach: Estimate when $\min_{\mathbb{P}}$ is achieved[2].
  - **Challenge**: Find a Model-Free Approach.
- Extend to Multiple Agents:
  - Common approach: Treat the environment as an implicit player. Solve NE of a general sum game[3].
  - **Challenge**: PPAD complete.

[2] e.g. *Robust Conservative Policy Iteration (Draft).*

[3] *Robust Multi-Agent Reinforcement Learning with Model Uncertainty.*

Learn the robust value function **without** estimating the worst-case transition probability[4]:

- Solve the explicit expression of the worst-case value function.
  - e.g. p-norm: $\{\mathbb{P}_0 + \rho : \|\rho\|_p \leq \beta, \sum \rho_i = 0\}$.

$$V(s) \leftarrow V(s) + \alpha \left(\underbrace{r(s,a) + \gamma V(s') - V(s)}_{\text{stand. TD err. under } P_0} - \gamma \langle \mathrm{O}_{\beta,p}(V^\pi), V \rangle \right), \quad (1)$$

Let q satisfy $\frac{1}{q} + \frac{1}{p} = 1$.

$$\mathrm{O}_{\beta,p}(V)(s') := \beta \frac{\mathrm{sign}(V(s') - \omega_q(V)) |V(s') - \omega_q(V)|^{q-1}}{\kappa_q(V)^{q-1}},$$

where $\omega_q(V) := \mathbf{arg\,min}_\omega \|V - \omega 1_{|S|}\|_q$ and
$\kappa_q(V) := \mathbf{min}_\omega \|V - \omega 1_{|S|}\|_q$.

(1) Cannot be extended to the continuous state space. (2) p-norm can be replaced with other norm but maybe hard to solve.

[4] *Policy Gradient for Rectangular Robust Markov Decision Processes*

Nash Equilibrium

$$V_{\pi}^{(j)} \geq V_{\pi^{(j)} \times \pi^{(\setminus j)}}^{(j)} \text{ for all } j$$

Correlated Equilibrium

$$V_{\pi}^{(j)} \geq V_{\phi_j \circ \pi}^{(j)} \text{ for all } j$$

For Robust NE and Robust CE, replace the value function V with the robust value function.

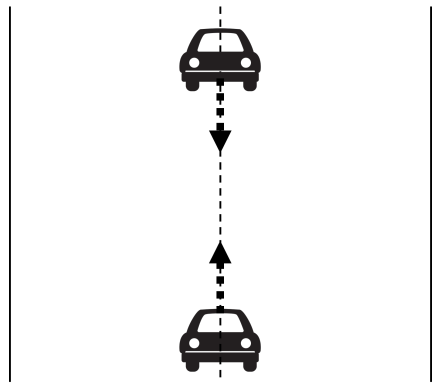

Figure 5: NE: (L, R) and (R, L). CE: $(A, \neg A)$ for all $A \in \{R, L\}$

Solve NE of a general sum game: PPAD complete. The set of CE is larger than NE; therefore, it is easier to solve.[5]

A tractable notion for robust Markov games:

- Robust correlated equilibrium.

[5] Decentralized Robust V-learning for Solving Markov Games with Model Uncertainty