An Introduction to Reinforcement Learning

March 11, 2024

This Course

- What is Reinforcement Learning (RL)
  - Examples and Mathematical Definition
  - Supervised/Unsupervised Learning and RL
  - Dynamic Programming and RL
- RL in the Economics Literature
  - Single-Agent RL
  - Multi-Agent RL

What is RL

- Reinforcement Learning is about an Agent learns via interacting with an Environment
- Literal Decomposition:
  - Reinforcement: Reward-Driven
  - Learning: Optimal Policy
- Components:
  - State of the Environment
  - Action taken by the Agent
  - Reward as a sequence of the State and the Action

What is Reinforcement Learning?

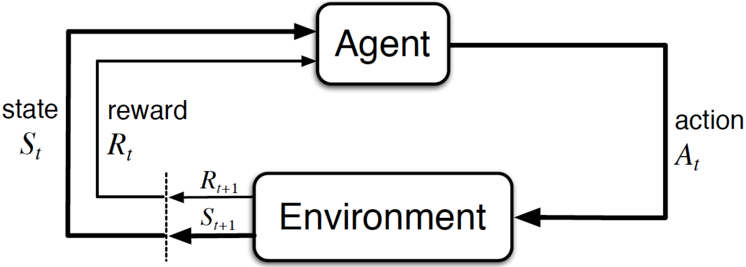


Figure: Agent-Envrionement Interaction by Sutton and Barto (2018)

What is RL: Example I

- State: current position
- Action: Up, Low, Left, Right
- Reward: ?

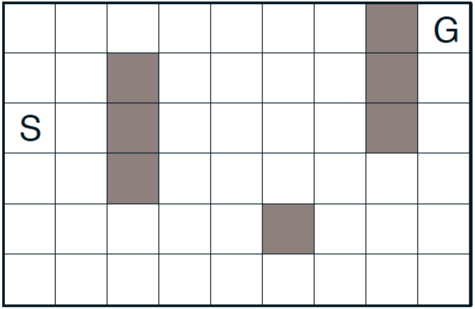


Figure: An Maze Problem

What is RL: Example II

- The *Frozen-Lake* Environment:
  "The ice is slippery, so you won't always move in the direction you intend."

```
SFFF       (S: starting point, safe)
FHFH       (F: frozen surface, safe)
FFFH       (H: hole, fall to your doom)
HFFG       (G: goal, where the frisbee is located)
```

Figure: Frozen-Lake

What is RL: Example III

- The *Cart-Pole* Environment:
  GIF
- State:
  - Cart Position: [-4.8, 4.8]
  - Cart Velocity: [-Inf, Inf]
  - Pole Angle: [-24°, 24°]
  - Pole Angular Velocity: [-Inf, Inf]
- Action: 0 (Left) or 1 (Right)
- Reward: $+1$ for every step

What is RL: Example IV

- A consumption-saving model (finite or infinite horizon) in macroeconomics
- State: (k_t, ϵ_t), where $k_t \in [k_{\min}, k_{\max}]$ is the capital holding, $\epsilon_t \in \{0, 1\}$ is the employment status
- Action: c_t, the consumption
- Reward: $u(c_t)$, the utility

What is RL: Mathematical Definition

- Definition: A Markov decision process (MDP) is a 4-tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where:
  - $\mathcal{S}$ is a set of states called the state space
  - $\mathcal{A}$ is a set of actions called the action space
  - $P(s, a, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the prob. that action a in state s at time t will lead to state s' at time $t + 1$
  - $R(s, a, s')$ is the immediate reward received after transitioning from state s to state s', due to action a

What is RL: Mathematical Definition

- Definition: A Markov decision process (MDP) is a 4-tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where:
  - $\mathcal{S}$ is a set of states called the state space
  - $\mathcal{A}$ is a set of actions called the action space
  - $P(s, a, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the prob. that action a in state s at time t will lead to state s' at time $t+1$
  - $R(s, a, s')$ is the immediate reward received after transitioning from state s to state s', due to action a
- RL solves an MDP problem:
  - An Agent observes state $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$ based on a policy $g \in \mathcal{S} \to \mathcal{A}$, the environment produces a reward r_t and moves to s_{t+1}
  - The goal is to find an optimal policy that obtaining accumulative rewards $\sum_{i=1}^{n} \gamma^t R_t$ using a Training Algorithm

Introduction: Agent

- The decision-making policy:
  - Indirect: value function approach: $V(s)$ or $Q(s, a)$
  - Direct: policy function approach: $a = g(s)$
  - How to parameterize the value/policy function?
- The behavioral policy:
  - E.g., the ϵ-greedy policy:

$$\pi(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a = \text{argmax}_{a'} Q(s, a') \\ \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{otherwise} \end{cases}$$

  - The *exploration-exploitation trade-off*
- Other structures facilitate the solution: e.g. the "memory for experiences"

Introduction: Training Algorithm

- Define the accumulative reward $G_t = \sum_{t=1}^{n} \gamma^t R_t$
- The celebrated Bellman Equation:

$$\begin{aligned} V_*(s) &= \max_a \mathbb{E}\left[R_t + \gamma G_{t+1} \mid S_t = s, A_t = a\right] \\ &= \max_a \mathbb{E}\left[R_t + \gamma V_*\left(S_{t+1}\right) \mid S_t = s, A_t = a\right] \\ &= \max_a R_t + \gamma \sum_{s'} P(s'|s,a) V_*(s') \end{aligned}$$

- Version for State-Action Value Function (Q-Function):

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

- Another version of Bellman Equation for Policy Evaluation:

$$\begin{aligned} V_g(s) &= \mathbb{E}\left[R_t + \gamma G_{t+1} \mid S_t = s, A_t \sim g(s)\right] \\ &= \mathbb{E}\left[R_t + \gamma V_g\left(S_{t+1}\right) \mid S_t = s, A_t \sim g(s)\right] \end{aligned}$$

Machine Learning: SL, UL, RL

- Three broad categories: Supervised Learning (SL), Unsupervised Learning (UL) and Reinforcement Learning (RL)
- SL: "You know what is true"
  - Data: $\{x_i, y_i\}_{i=1...N}$
  - Task: find $f : \mathbb{X} \to \mathbb{Y}$ such that $f(x) \approx y$
- UL: "You DON'T know what is true"
  - Data: $\{x_i\}_{i=1...N}$
  - Task: find some sort of underlying structure, correctly label/group the data based on x_i
- RL: "You know what SHALL be true"
  - Data: $\{x_t\}_{t=1...T}$ is our generated state, $\{r_t\}_{i=1...T}$ "signals of correctness"
  - Task: find $f : \mathbb{X} \to \mathbb{Y}$ an optimal policy function

Supervised Learning: An illustration

The "Hello World" problem in supervised learning

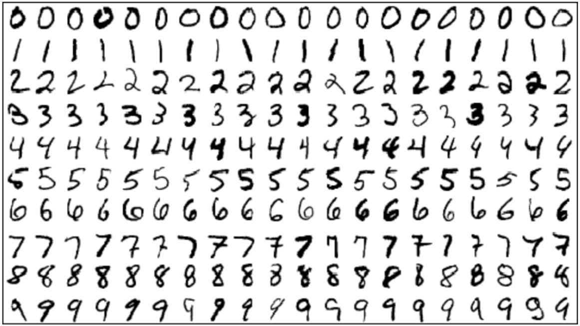
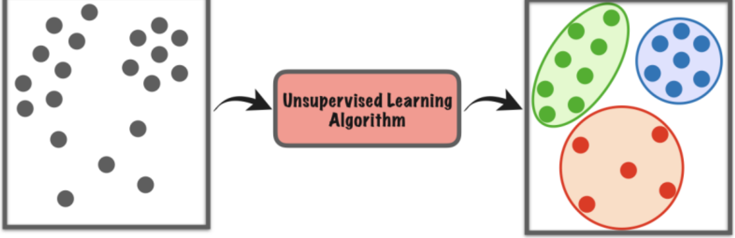


Figure: MNIST data

Unsupervised Learning: An illustration



Optimal Control: DP and RL

- Recall the Bellman Equation in terms of Q-Function:
  $Q_*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q_*(s',a')$
- Dynamic Programming (DP): P is known and closed-form
- In practice:
  - P is not known or hard to express in closed-form
  - $\mathcal{S}, \mathcal{A}$ is continuous/high-dimensional
  - the $\max$ operator is computationally expensive
- Problem 1: Simulation. The celebrated Q-learning algorithm:
  $Q^{i+1}(s,a) = (1-\alpha) Q^i(s,a) + \alpha(r + \gamma \max_{a'} Q^i(s',a'))$
- Problem 2 & 3: we use Neural Network (Deep RL)
  - Critic: A Value Network $Q_\theta(s,a)$
  - Actor: A Policy Network $g_\phi(s)$

RL in Economics: Literature

- DRL in a Monetary Model (Chen, Joseph, Kumhof, Pan and Zhou, 2021)
- AI, algorithmic pricing and collusion (Calvano, Calzolari, Denicolo and Pastorello, 2020)
- AI as structural estimation: Deep Blue, Bonanza, and AlphaGo (Igami, 2020)
- RL for Optimization of COVID-19 Mitigation policies (Kompella, Capobianco, Jong, Browne, Fox, Meyers, Wurman and Stone, 2020)
- ...

Multi-Agent Learning and Game Theory

- Link: Multi-Agent Hide and Seek
- The learning of other agents would make the Environment non-stationary
- Many game-theory settings have been studied previously for Multi-Agent learning, "Evolutionary Game Theory"
- It is non-trivial to build up learning algorithms even for those simple games

Multi-Agent Learning and Game Theory

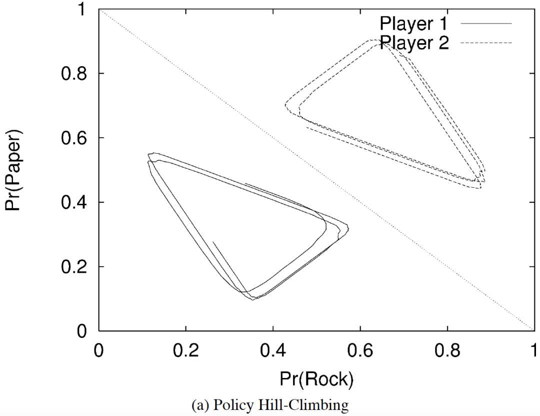


Figure: Non-Convergence in Rock-Paper-Scissor

Multi-Agent Learning and Game Theory

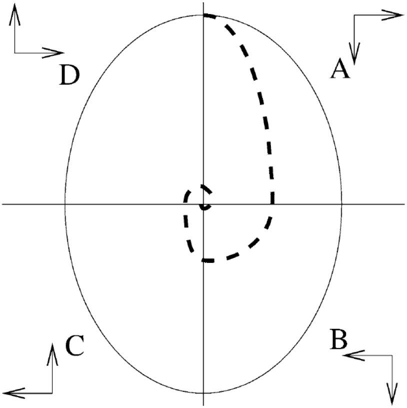
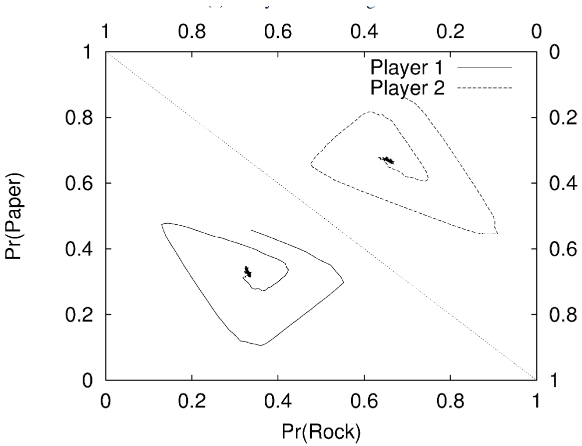


Figure: The "Win-or-Learn-Fast" Algorithm

Multi-Agent Learning and Game Theory



(b) WoLF Policy Hill-Climbing

Figure: Convergence in Rock-Paper-Scissor with WoLF

Multi-Agent Reinforcement Learning

- Link: AI-Economist with tax policies (Zheng, Trott, Srinivasa, Naik, Gruesbeck, Parkes and Socher, 2020)
- MARL in Cheap Talk (Condorelli and Furlan, 2023)
- MARL in Stackelberg Game (my working paper)

Conclusion

- RL is nothing far away from economists
- RL could potentially help us to solve some complex settings where we should rely on simulations to solve agents' decision-makings
- MARL could even go further to study more interactive settings
  - policy-makers' problem in macro
  - strategic plays in game theory
  - firms' interaction in IO
  - ...

Calvano, Emilio, Giacomo Calzolari, Vincenzo Denicolo, and Sergio Pastorello, "Artificial intelligence, algorithmic pricing, and collusion," *American Economic Review*, 2020, *110* (10), 3267–3297.

Chen, Mingli, Andreas Joseph, Michael Kumhof, Xinlei Pan, and Xuan Zhou, "Deep reinforcement learning in a monetary model," *arXiv preprint arXiv:2104.09368*, 2021.

Condorelli, Daniele and Massimiliano Furlan, "Cheap Talking Algorithms," *arXiv preprint arXiv:2310.07867*, 2023.

Igami, Mitsuru, "Artificial intelligence as structural estimation: Deep Blue, Bonanza, and AlphaGo," *The Econometrics Journal*, 2020, *23* (3), S1–S24.

Kompella, Varun, Roberto Capobianco, Stacy Jong, Jonathan Browne, Spencer Fox, Lauren Meyers, Peter Wurman, and Peter Stone, "Reinforcement learning for optimization of COVID-19 mitigation policies," *arXiv preprint arXiv:2010.10560*, 2020.

Sutton, Richard S and Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 2018.

Zheng, Stephan, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David C Parkes, and Richard Socher, "The ai economist: Improving equality and productivity with ai-driven tax policies," *arXiv preprint arXiv:2004.13332*, 2020.