

CS234 Reinforcement Learning

Yaro Kazakov

September 16, 2024

Abstract

These are study notes to prepare myself for PhD applications.

1 RL - Defining the Framework

Generally includes:


1. Optimization
 - (a) Goal is to find an optimal way to make decision
 - (b) Explicit notion of decision utility
 - (c) Example: finding minimum distance route between two cities given network of roads
2. Delayed Consequences
 - (a) Decisions now can impact things much later (Saving for retirement)
 - (b) When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer term ramifications. When learning: temporal credit assignment is hard (what caused later high or low rewards?)
3. Exploration
 - (a) Learning about the world by making decisions (Agent as scientist)
 - (b) Decisions impact what we learn about
 - i. Only get a reward for decision made
 - ii. Don't know what would have happened for other decision
 - iii. If we choose to go to Stanford instead of MIT, we will have different later experiences...
4. Generalization
 - (a) Policy is mapping from past experience to action

1.1 RL vs Other AI and ML

RL has access to labels.

UL does not.

IL reduces RL to SL. **IL is very popular in 2024**



	AI Planning	SL	UL	RL	IL
Optimization	X			X	X
Learns from experience		X	X	X	X
Generalization	X	X	X	X	X
Delayed Consequences	X			X	X
Exploration				X	

- SL = Supervised learning; UL = Unsupervised learning; RL = Reinforcement Learning; IL = Imitation Learning
- Imitation learning typically assumes input demonstrations of good policies
- IL reduces RL to SL. **For many good reasons, IL is very popular.**

Figure 1: RL compared to other Learning

1.2 Where is RL popular?

Enormous search or optimization problem with delayed outcomes.

No examples of desired behavior: e.g. because the goal is to go beyond human performance or there is no existing data for a task

1.3 History and Markov

History = $(o_0, a_0, r_1, o_1, \dots)$

Function of history is s_t . State is information assumed to determine what happens next.

Why is Markov assumption SO POPULAR?

1. Simple and often can be satisfied if include some history as part of the state
2. In practice often assume most recent observation is sufficient statistic of history
3. State representation has big implications for:
 - (a) Computational Complexity
 - (b) Data Required
 - (c) Resulting Performance

1.4 MDP model

Agent's representation of how world changes given agent's action

Transition/dynamics model predicts next agent state

$$p(S_{t+1} = s' | s_t = s, a_t = a) = p(s' | s, a)$$

Reward model predicts immediate reward

$$r(s, a) = E[r_t | s_t = s, a_t = a]$$

Policy determines how the agent chooses actions

Deterministic policy: $\pi(s) = a$, **Stochastic:** $\pi(a|s) = Pr(A_t = a | S_t = s)$

1.5 Evaluation and Control

Evaluation - Estimate/predict the expected rewards from following a given policy

Control - Optimization: find the best policy

Build Up in Complexity

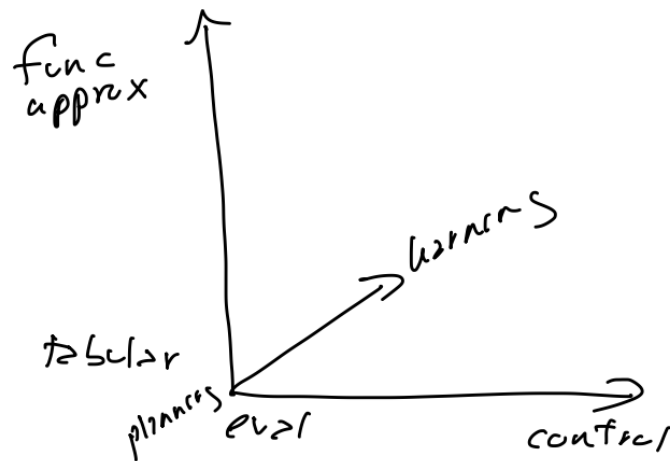


Figure 2: 3 dimensions of RL in terms of Complexity

1.5.1 Making Sequences of Good Decisions Given a Model of the World

1. Markov Processes (MPs)
2. Markov Reward Processes (MRPs)
3. Markov Decision Processes (MDPs)
4. Evaluation and Control in MDPs

Definition of **Markov Process**:

1. S is a (finite) set of states ($s \in S$)
2. P is dynamics/transition model that specifies $p(s_{t+1} = s' | s_t = s)$
3. Note: **NO rewards, NO actions**
4. This is very similar to MPs in Finance!!

Definition of **Markov Reward Processes (MRPs)**:

1. S is a (finite) set of states ($s \in S$)
2. P is dynamics/transition model that specifies $p(s_{t+1} = s' | s_t = s)$
3. R is a reward function $R(s_{t+1} = s') = E[r_t | s_t = s]$
4. Discount Factor $\gamma \in [0, 1]$
5. Note: **NO actions**

Definition of Horizon (H)/Terminal State(T) in Sutton and Barto:

1. Number of time steps in each episode
2. Can be infinite
3. Otherwise called finite Markov reward process

Discount Factor:

1. Mathematically convenient (avoid infinite returns and values)
2. Humans often act as if there's a discount factor < 1
3. If episode lengths are always finite, can use $\gamma = 1$
4. $\gamma = 0$: Only care about immediate reward
5. $\gamma = 1$: Future reward is as beneficial as immediate reward

1.6 Markov Reward Processes Formulation

$V(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s)V(s')$, i.e. immediate reward plus discounted sum of future rewards

In a matrix form can be written as:

$$V = R + \gamma * P * V$$

Analytic Solution:

$$V = (I - \gamma P)^{-1} R$$

Solving directly requires taking a matrix inverse $O(N^3)$ complexity. A computationally cheaper way is solving iteratively (Dynamic Programming).

1.7 Iterative Algorithm for Computing MRPs

1. Initialize $V_0(s) = 0$ for all s
2. For $k = 1$ until convergence
 - (a) For all s in S:
 - i. $V_k(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s)V_{k-1}(s')$

Computational Complexity is $O(|S|^2)$ for each iteration ($|S| = N$)

1.8 Summary

Reinforcement learning involves **learning, optimization, delayed consequences, generalization and exploration**