强化学习理论基础

Statistical RL Theory

Contents.

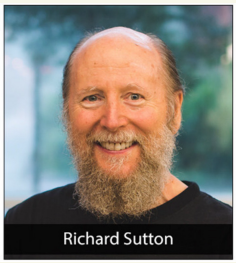
1. Introduction
2. Planning in MDP
3. Tabular Analysis
4. State Abstractions
5. Off-Policy Evaluation
6. Policy Gradient
7. PAC Exploration
8. Linear Bandits & MDP
9. Multi-Agent RL
10. Partial Observability

tabular case

→ offline RL.

function approximation case

→ online RL

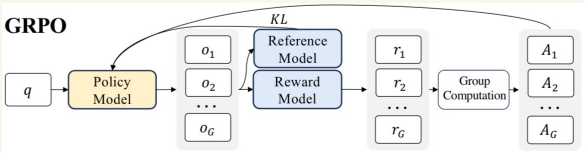


Reference.

- Reinforcement Learning: An Introduction, Rich Sutton & Andrew Barto.
- Reinforcement Learning: Theory and Algorithms, A. J. K. S
  (Alekh Agarwal. Nan Jiang. Sham Kakade. Wen Sun)

Introdution to RL

- Various RL applications:
  - Games Alpha GO
  - Control Optimal Control
  - Automatic Driving
  - Large Language Models



DeepSeekMath

- What is Reinforcement Learning (RL) ?

等待
连接
交互

The idea that we learn by interacting with our environment is probably the first to occur to us when we think about the nature of learning. When an infant plays, waves its arms, or looks about, it has no explicit teacher, but it does have a direct sensorimotor connection to its environment. Exercising this connection produces a wealth of information about cause and effect, about the consequences of actions, and about what to do in order to achieve goals. Throughout our lives, such interactions are undoubtedly a major source of knowledge about our environment and ourselves. Whether we are learning to drive a car or to hold a conversation, we are acutely aware of how our environment responds to what we do, and we seek to influence what happens through our behavior. Learning from interaction is a foundational idea underlying nearly all theories of learning and intelligence.

delayed reward

In this book we explore a *computational* approach to learning from interaction. Rather than directly theorizing about how people or animals learn, we primarily explore idealized learning situations and evaluate the effectiveness of various learning methods. That is, we adopt the perspective of an artificial intelligence researcher or engineer. We explore designs for machines that are effective in solving learning problems of scientific or economic interest, evaluating the designs through mathematical analysis or computational experiments. The approach we explore, called *reinforcement learning*, is much more focused on goal-directed learning from interaction than are other approaches to machine learning.

Statistically efficient.

computationally efficient

decision

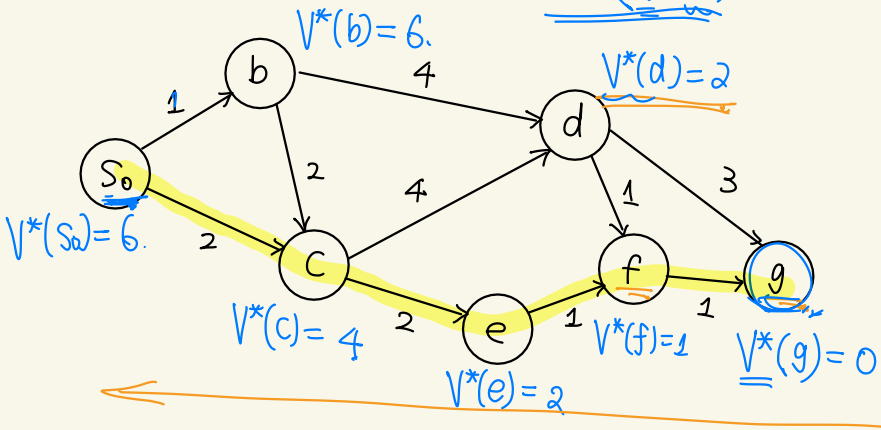
Model-based : planning

Shortest Path

policy 策略

$G=(V,E)$ → vertex ↔ state, edge ↔ action

Backup-Diagram



bottom-up ✓ < top-down

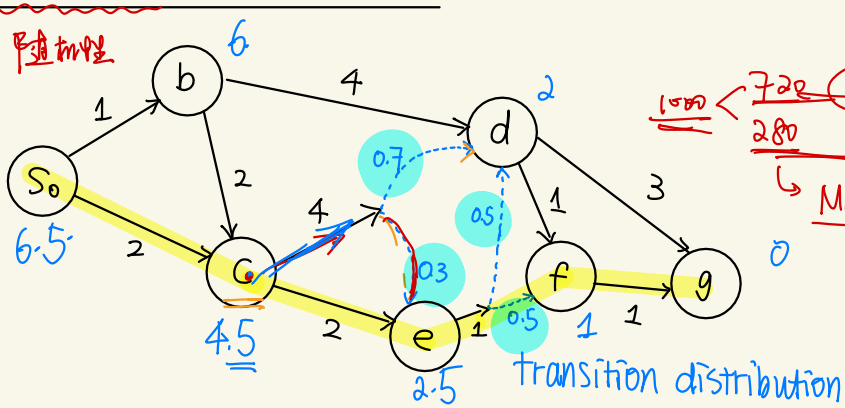
Bellman equation: $V^*(d) = \min\{V^*(f)+1,\ V^*(g)+3\}$

Greedy is suboptimal due to delayed reward

⇒ need long-time planning

Stochastic Shortest Path

随机的



transition distribution

1000 < 720 c→d 0.72

280 c→e 0.28

↳ Monte Carlo (Sampling)

MDP

Bellman equation: $V^*(c) = \min\{V^*(e)+2,\ 4+0.7\times V^*(d)+0.3\times V^*(e)\}$

Model-free → learning

trajectory 1: $s_0 \to c \to d \to \cdots$

trajectory 2: $s_0 \to c \to e \to \cdots$

trial-and-error

• How many trajectories do we need to compute a near-optimal policy ?

(data)

offline

online

Exploration 探索

Exploitation 利用

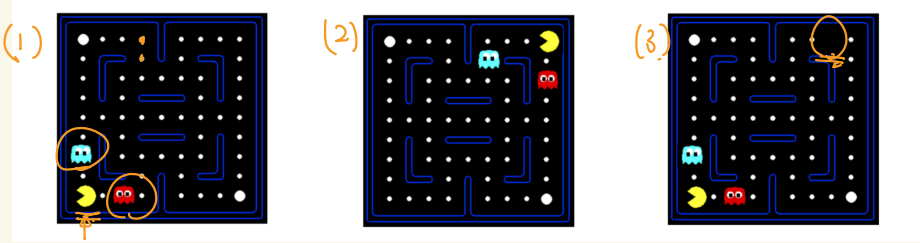
OFU ↔ PFU

Sample complexity 样本复杂度

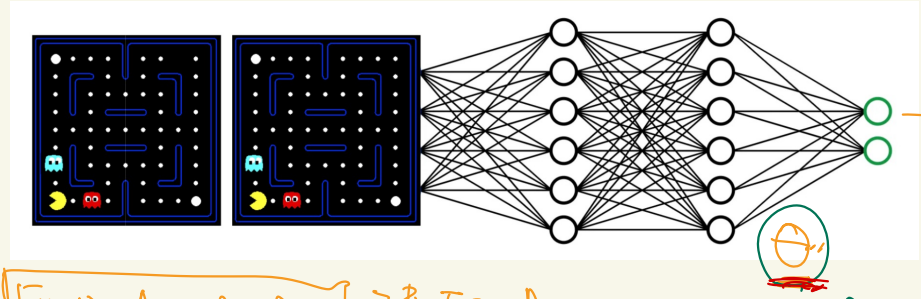
non-asymptotic

Large State Space

$|S|$



e.g.



Function Approximation 函数逼近 ☆

features

learn parameters

data

(generalization)

• Low rank MDP (linear MDP)

Value-Targeted Regression

A Machine Learning Perspective

ML ← SL, USL, RL

Supervised Learning 监督学习

Given dataset $\boxed{\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}}$ learn $f: x \to y$.

$\|f(x) - y\| < \varepsilon$

online (streaming) version:

for round $t = 1, 2, \cdots$, the learner:

- observe $x^{(t)}$
- predict $\hat{y}^{(t)}$
- receive $\boxed{y^{(t)}}$

$\Rightarrow$ want to maximize $\#\{y^{(t)} = \hat{y}^{(t)}\}$

loss $\boxed{\ell(y, \hat{y}) = \mathbb{1}\{\hat{y} \neq y\}}$

① Dataset: fixed

② "Full information" 完全信息

Core challenge: generalization 泛化

ERM $\Downarrow$ generalization bound

Contextual Bandits 上下文赌博机

arm 摇臂

no $x^{(t)}$

for round $t = 1, 2, \cdots$, the learner:

- observe $x^{(t)}$ (state)
- choose from a set of actions $a^{(t)} \in A$. (arm)
- receive reward $r^{(t)} \sim R(x^{(t)}, a^{(t)})$

Explore-then-Commit

MAB (Multi-Armed Bandits)

Bandits Algorithms by Lattimore & ...

$\Rightarrow$ want to maximize total reward $\sum_{t=1}^{T} r^{(t)}$

or: minimize total regret 遗憾 $T \max_{a \in A} R(x^{(t)}, a) - \boxed{\sum_{t=1}^{T} r^{(t)}}$.

e.g. 1. for an image, the learner guesses a label, and is told whether it's correct or not (reward = 1 if correct and 0 o.w).

△ Do not know what's the true label.

e.g. 2. for an user, the website recommends a movie, and observes whether the user likes it or not.

△ Do not know what movies the user really want to see.

① You can generate your own dataset

$\{(x^{(t)}, a^{(t)}, r^{(t)})\}_{t=1}^{T}$

② "Partial Information Setting"

$l(a_t)$ 值 ✓

$l(\cdot)$ 函数 ×

Reinforcement Learning: Multi-Step Bandits

Finite-horizon

episode

for round $t = 1, 2, \cdots$:

horizon (时域)

轮 (回合)

for time step $h = 1, 2, \cdots, H$, the learner:

步 (阶段)

stage

contextual bandits

- Observe $x_h^{(t)}$
- choose $a_h^{(t)} \in A$
- receives $r_h^{(t)} \sim R(x_h^{(t)}, a_h^{(t)})$
- Next $x_{h+1}^{(t)}$ is generated as a function of $x_h^{(t)}$, $a_h^{(t)}$.

deterministic

stochastic

$\mathbb{P}$: transition dynamics

(or sometimes: all previous x's and a's within round t)

MDP (Markov Decision Process) → non-Markovian

episodic RL

continuous, discounting RL

planning : model-based

learning : model-free

Why Statistical RL ?

There're two types of scenarios in RL research: ???

① Solving a large planning problem using a learning approach.

e.g. AlphaGo, video game playing, simulated robotics.

Dyna

- Transition dynamics (Go rules) known, but too many states.
- Run the simulator to collect data.

simulated model

simulated experiences

RL { reward, learner, environment, model }

② Solving a learning problem.

e.g. adaptive medical treatment.

- Transition dynamics unknown (and too many states)
- Interact with the environment to collect data.