books:
Sutton
and
Barto
RL
(intuition)

Reinforcement Learning

L1

(David Silver, UCL, DeepMind)

Szepesvari
RL
(Math,
rigor)

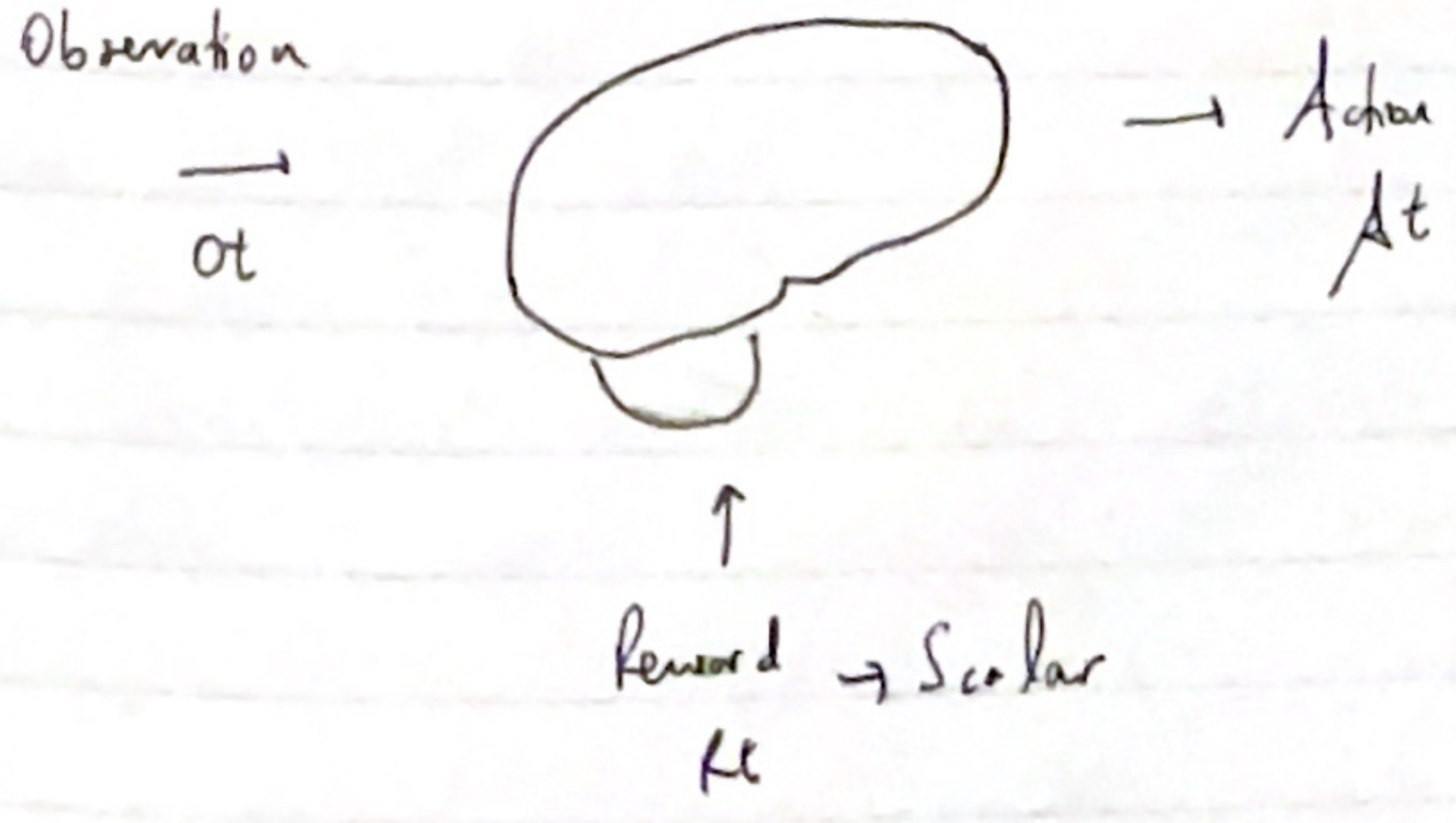
→ Science of decision making
→ other fields have other names for it

RL ≠ from other methods:
- → no supervisor, only rewards
- → feedback is delayed
- → time really matters (sequential, no i.i.d. data)
- → agent has its own actions → influencing the environment
  → influencing the data it receives –

Reward hypothesis → they look for the maximization of the cumulative reward

Goal – select actions to maximize the future reward

Actions might have long term consequences



history sequence of observations = $H_t = A_1, O_1, R_1, \dots A_t, O_t, R_t$

State → concise summary of the history

↓

environment state → agent doesn't see them but the algorithm cannot depend on it

good idea to ~~manipulate it~~ [above: have numbers to make it easier] manipulate it

Agent state → a° in our algorithm - used to pick next action

↳ our decision - what to use / what to throw away

Information State (Markov State) contains all useful information from history

A State St is Markov if and only if:

$$\mathbb{P}[S_{t+1} \mid S_t] = \mathbb{P}[S_{t+1} \mid S_1, \ldots, S_t]$$

probability of next state, conditioned the state you're in

is the same if you show all past info

You can throw away all your past info if you have your present info

"the future is independent of the past given the present"

$H_{1:t} \rightarrow S_t \rightarrow H_{t+1\infty}$

↓

State is sufficient statistic of the future

→ What we believe will

representation of state

Fully observable environment

state

$O_t = S_t$

Agent state = environment

Markov

Partial observability → indirect

(eg robot with cam

|

Partially Obs

Agent must construct its

- beliefs of enviro
- complete history
- recurrent neural ne

And RL might include

policy → agent's behaviour

value function → how

model → agent's

them but

depend on it
to make it easier
like it

used to pick next action
it to throw away

ns all useful information from

and only if:

$S_1, \dots, S_t]$

the same if you throw
past info

way all your past
have your present info

the past given the

$t_t + 100$

sufficient statistic of the

→ What we believe will happen next, depends on our representation of state (eg lever and cheese)

Fully observable environment = agent directly observes environment state

$$O_t = S_t^a = S_t^e \quad (?)$$

Agent state = environment state = information state

↓

Markov Decision Process
(MDP)

Partial observability → indirectly observes environment
(eg robot with camera)

|

Partially Observable MDP (POMDP)

Agent must construct its own state representation
- beliefs of environmental state
- complete history
- recurrent neural network

An RL might include:
policy → agent's behaviour function
value function → how good is each state and / or function
model → agent's representation of environment

59'

Policy –

map from state to action

Deterministic → $a = \pi(s)$

Stochastic $\pi(a | s) = \mathbb{P} | A = a | S = s |$

→ taken particular action given certain state

Categorizing RL agents

Value based

Used to evaluate goodness / badness of State

$$V_\pi(s) = \mathbb{E}_\pi[R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots | S_t = s]$$

Model based

predictions – P predicts the next state (dynamics)

$$P_{ss'}^a = \mathbb{P}[S' = s' | S = s, A = a]$$

rewards

↓

R predicts the next (immediate) reward

$$R_s^a = \mathbb{E}[R | S = s, A = a]$$

Actor critic – both model

value

(2)

Model free → policy and / or ~~fa~~ value function

Model based

va

Key Subproblems

→ Learning vs Planning → environment is unknown and agent learns through interaction. In planning, the environment model is fully known, and the agent performs internal look-ahead search

|

RL often combines both

|

1° learning a model

2° planning with it

→ Exploration vs Exploitation

- → Universal tradeoff unique to RL
- → exploitaton: uses current knowledge to maximize known reward
- → exploration: sacrifices immediate reward to discover better long-term options (eg trying new restaurant vs favourite restaurant)

→ Prediction vs Control

Prediction answers "how well will I do following my current policy?"

Control asks "what is the optimal policy?"

→ Solving prediction problem is typically necessary for solving control