CS234: Reinforcement learning – [2019]

20/12/2020

Lecture 1 : Intro

Reinforcement Learning aims to learn to make good sequences of decisions

Learn to make good sequence of decisions

Learn: don't know in advance how the world works

good: learn to make good sequences of decisions

sequence of decisions: repeated interaction with world

- fundamental challenge in artificial intelligence and machine learning is learning to make good decisions under uncertainty.

RL, behavior & Intelligence

Yael Niv

Childhood : primitive brain & eye, swims around, attaches to a rock

Adulthood : digests brain and sits

→ Brain is helping guide decision [no more decisions, no need for brain?]

DeepMind Nature, 2015 (Atari Learning)

Robotics

Educational Games

~~I~~ Health care

NLP, Vision

<u>RL involves</u>

Optimization — Goal is to find an optimal way to make decisions — Or at least a very good strategy

Delayed consequences — Decisions now can impact things much later (Saving for retirement) — Introduces two challenges (when planning and when learning)

Exploration

Generalization

<u>Exploration:</u>

- Learning about the world by making decisions
- Censored data (Reward is the only way)
- Decisions impact what we learn about

Finding key of ~~Montezuma~~ ~~revenge~~ Montezuma revenge

<u>Policy</u> is mapping from past experience to action

Why we can't preprogram it? (Not possible in big cases)

Reinforcement Learning 2019 Stanford

24/12/2020

Lecture 1: Intro

RL

O - Optimization
G - Generalization
E - Exploration
D - Delayed Consequency

AI Planning vs RL

O	G	E	D
✓	✓	✗	✓

Superised ML vs RL

O	G	E	D
✓	✓	✗	✗

Learns from experience

Unsupervise ML vs RL

O	G	E	D
✓	✓	✗	✗

Imitation ~~Ree~~ Learning [popularised by Andrew Ng]

O	G	E	D
✓	✓	✗	✓

Learns from experience of others.

Assumes input demos of good policies.

- Reduces RL to supervized learning

Benifits

- Great tools for supervized learning
- Avoids exploration problem

• Limitations

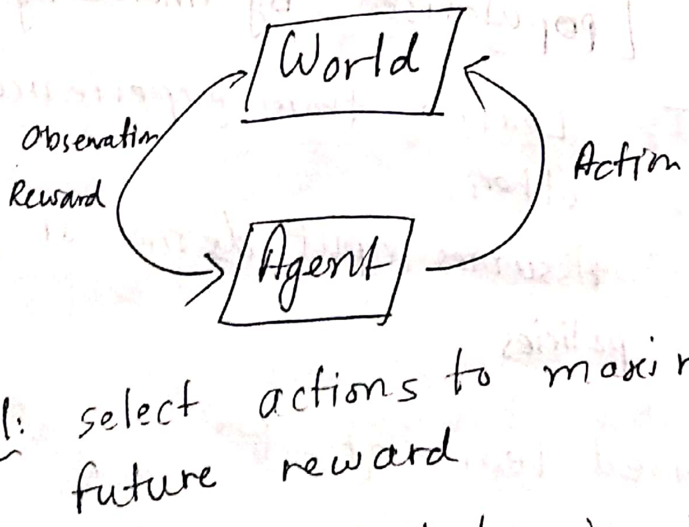
  • Can be expensive to capture
  • Limited by data collected.

Imitation ~~Rea~~ Learning + RL promising

How do we proceed?

- Explore the world
- Use experience to guide future decisions

• Introduction to sequential decision making under uncertainty.



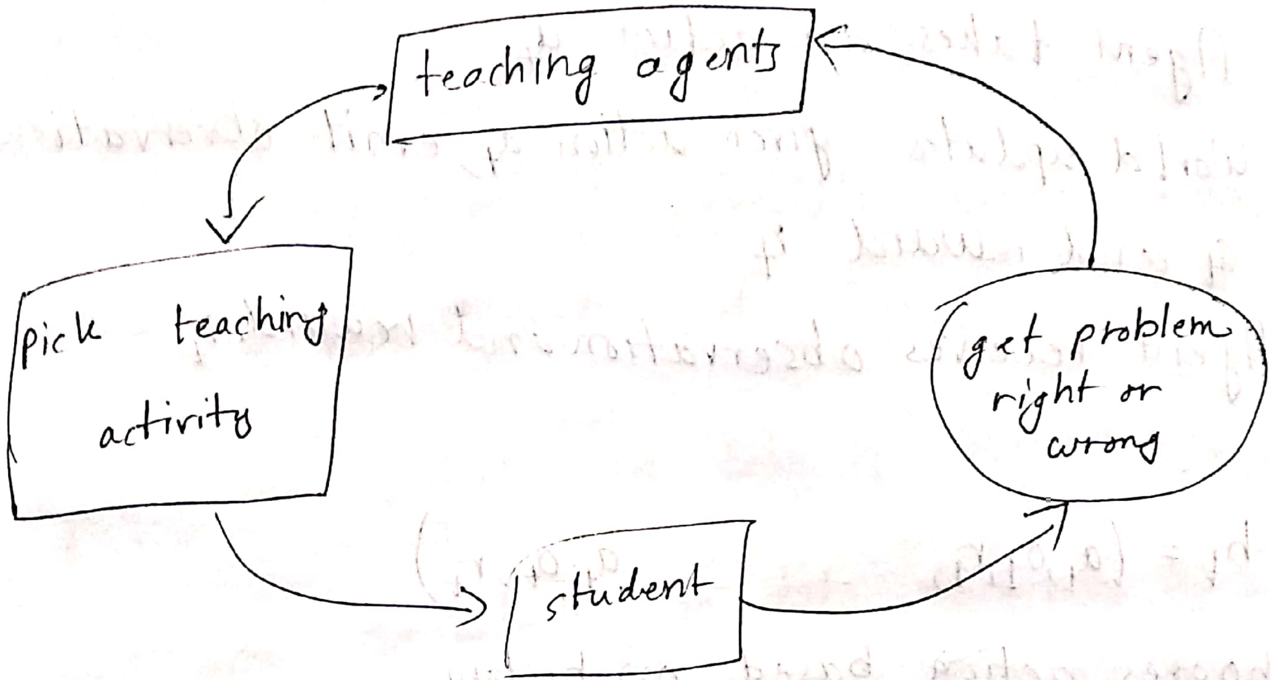
Example: Web Advertising (i)
Robot unloading (ii) Dishwasher
Blood Pressure control (i) (ii)

Goal: select actions to maximize total expected future reward

(i) • May require balancing intermediate & long-term ~~result~~ rewards

(ii) • May require strategic behavior to achieve high rewards

Artificial Tutor:



+1 if ~~they~~ student get the problem right

-1 if they get it wrong

what activity would agent choose to get max $\sum$ rewards?

Bev Wolf, 2000

The Student initially doesn't know addition (easier) nor substraction (harder)

Reward hacking → give easy problems first then hard problems

- Machine teaching

Each time step t:

- Agent takes an action a_t
- World updates given action a_t, emit observation o_t and reward r_t
- Agent receives observation and reward. r_t

History $h_t = (a_1, o_1, r_1, \ldots \ldots a_t, o_t, r_t)$

Agent chooses action based on history

State is information ~~&~~ assumed to determine what happens next. $s_t = (h_t)$

World State
- This is the true state of the world used to determine how world generates next observation and reward
- Often hidden and unknown to agent
- Even if ~~g~~ known may contain information not needed by agent.

Agent state / What the agent (algorithm uses to make decisions about how to act $s_t = f(h_t)$

Generally including meta info like state of algo. or decision process

Markov Assumption

- Information state: sufficient static of history
- State s_t is Markov if and only if

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|h_t, a_t)$$

- Future is independent of past given present

Hypertension control: let state be current blood pressure, and action be whether to take ~~decisions~~ of medications or not. Is the system markov? No

Website shopping: state is current product viewed by customer and action is what other product to recomment. Is the system Markov? No

Why is it popular?

- Can always be satisfied
  [Setting state as history always Markov; $s_t = h_t$]
- In practive $s_t = o_t$

State representation has big implication for

- Computational complexity
- Data required
- Resulting performance

Full Observability (MDP)

$S_t = O_t$

Partial Observability (POMDP)

Agent state is not the ~~st~~ same as world state

Agent constructs its own state

· Use history $s_t = h_t$ or beliefs of world state

Poker healthcare

Types of Sequential Decision Processes

Bandits : actions have no influence on next observations

- No delayed rewards.

How world changes

Deterministic (single observation and reward)

Stochastic (many " " ")

RL algo components

Model → representation of how the world changes

Policy → function mapping agent's states to action

$\pi(s) = a$ $\pi(a|s) = Pr(a_t = a | s_t = a)$

Value function : Future rewards from being in a state and/or action when following a particular policy

Value function : Expected discounted sum of future rewards under a particular ~~poli~~ policy π

$$v^{\pi}(s_t = s) = E_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \mid s_t = s]$$

Discount factor γ weights immediate vs future rewards $0 \leftrightarrow 1$

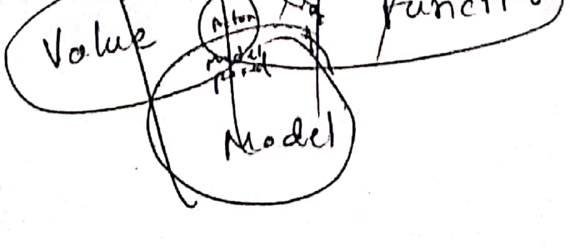
~~Can be quantifies goodne~~

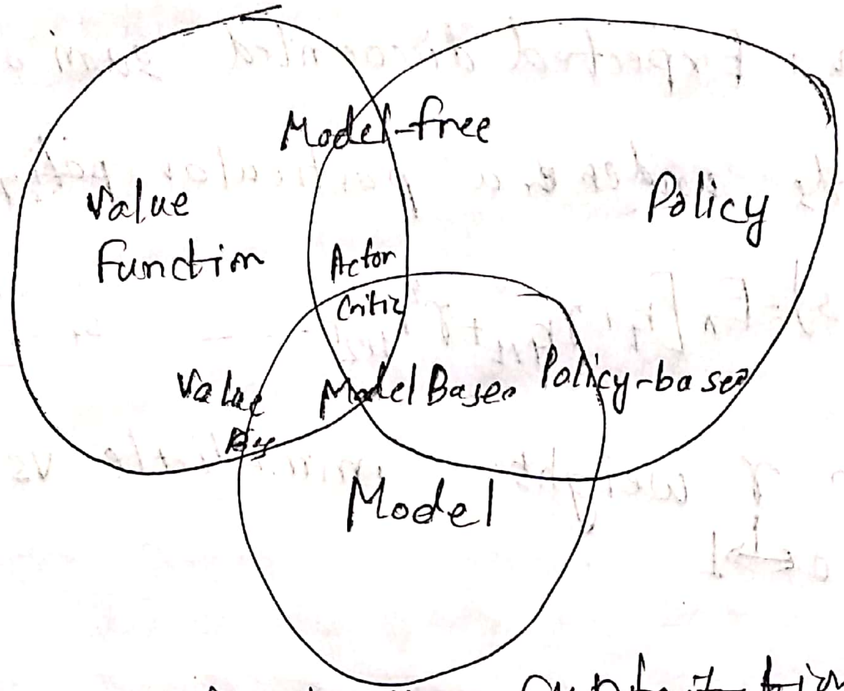
Can be used to quantify goodness/badnes

Decide how to act ~~compating~~ comparing policies.

Types of RL agents

- Model-based : Explicit: Mode
  - May or maynot have policy and/or value func
- Model-free · Explicit: Value function and/or policy function
  - No model





Exploration an ~~Exploration~~ ~~Exploitation~~

Exploration and Exploitation:

→ Choosing actions that ~~are~~ expected to yield good reward given past experience

→ trying new things that might enable the agent ~~to~~ make better decisions in the future

· May have to sacrifice reward in order to explore & learn about potentially better policy