RL Notes:

Module 2 Learning Action Values

- Sample-average method

→ The value of action is the expected reward

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a]$$

- a taken prior to t

$$Q_t(a) = \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

- Incremental update rule

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

$$= Q_n + \frac{1}{n}(R_n - Q_n)$$

$\alpha_n \in [0, 1]$

$$= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

- Greedy action selection
  - ↳ exploit current knowledge
  - ↳ explore & gain knowledge

- Exploration-Exploitation dilemma

- Estimated incrementally

- General update rule to solve non-stationary bandit prob

Trade off

- Exploration - exploitation trade off

when to?

→ Exploration - improve knowledge for long-term benefit

$q(a), N(a), q_*(a)$

→ Exploitation - exploit knowledge for short-term benefit

Optimistic Initial Values

- How optimistic initial values encourage?

$Q_{n+1} \leftarrow Q_n + \alpha (R_n - Q_n)$

- Epsilon greedy

$$A_t \leftarrow \begin{cases} \underset{a}{\text{argmax}}\, Q_t(a) & \text{with prob } 1-\varepsilon \\ a \sim \text{Uniform}(\{a_1, \dots a_k\}) & \text{with prob } \varepsilon \end{cases}$$

→ the 10-armed Testbed

Expectation across possible stochastic outcomes

⇒ To balance exploration and exploitation

- Describe limitations of OIV
  - Drive early exploration
  - Not well-suited for non-stationary problems
  - May not know max reward value

Upper-Confidence Bound (UCB), Action Selection

→ uses uncertainty in estimates to drive exploration ↗ better than ε greedy

$$A_t \doteq \arg\max_a \left[Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$Q_t(a)$: exploit; $\sqrt{\frac{\ln t}{N_t(a)}}$: explore

t: timesteps

$N_t(a)$: times action a taken

→ the Q_t values for these actions are normally distributed with mean zero and standard deviation one

- Contextual Bandits for Real World Reinforcement Learning

How to align?

→ Mind the gap:
Large simulator / reality divergence

How do you RWRL?
↳ real-world-based RL

Repeatedly:

1. Observe features x
2. Choose action $a \in A$
3. Observe reward r

Goal: Maximize reward

Key change:

No credit assignment!

⇒ Shift your priorities

Temporal Credit Assignment ↓ Generalization ↑

Control environment ↓ Environment controls ↑

Computational efficiency ↓ Statistical efficiency ↑

State ↓ Features ↑

Learning ↓ Evaluation ↑

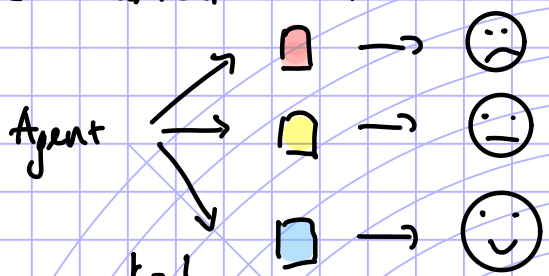
Last policy ↓ Every policy ↑

1995 EXP4 paper
2007 Epoch Greedy
2010 Personalized News
2011 Comp/Stat opt algo

2014 Better Algo
2016 Created first version of "Decision Service"
2019 First RL Service Product "Azure Cognitive Services Personalizer"

→ Summary

- The K-armed Bandit



- Action-Values

$$q_*(a) \doteq \mathbb{E}[R_t \mid A_t = a] \quad \forall a \in \{1, \dots, k\}$$

$\mathbb{E}$: Expected reward received; $A_t = a$: when taking action a

→ unknown to the agent

- Exploration vs. Exploitation

Exploration: improve knowledge for long-term benefit

Exploitation: exploit knowledge for short-term benefit

- Optimistic Initial Values

↳ OIV > $q_*(a)$, the agent will systematically explore the actions.

The optimism fades with time and the agent eventually stops exploring.

- Sample-Average Method → estimate $q_*(a)$

$$Q_t(a) \doteq \frac{\sum_{i=1}^{t-1} R_i}{t-1}$$

- Incremental update rule (α_t)

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{N_t(a)}\left(R_t - Q_t(a)\right)$$

New estimate ← oldEs + Stepsize[target − OldEs]

How do choose when to explore, when to exploit!

- Epsilon-Greedy Action Selection

$$A_t \leftarrow \begin{cases} \underset{a}{\arg\max}\, Q_t(a), & \text{prob } 1-\varepsilon \quad \text{(exploit)} \\ a \sim \text{Uniform}(\{a_1, \dots a_k\}), & \text{prob } \varepsilon \quad \text{(explore)} \end{cases}$$

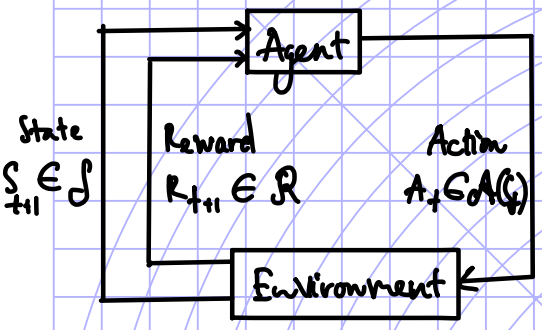
- UCB-Action Selection

$$A_t \doteq \underset{a}{\arg\max}\left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}\right]$$

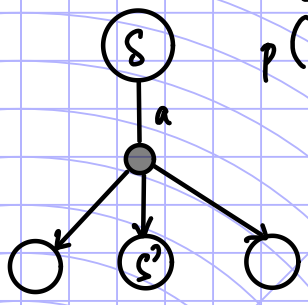
Exploit | Explore

Module 3 Finite Markov Decision Processes

- Markov Decision Processes



- How the dynamics of an MDP are defined



$p(s', r \mid s, a)$: prob distribution

→ Markov properties:

$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0,1]$

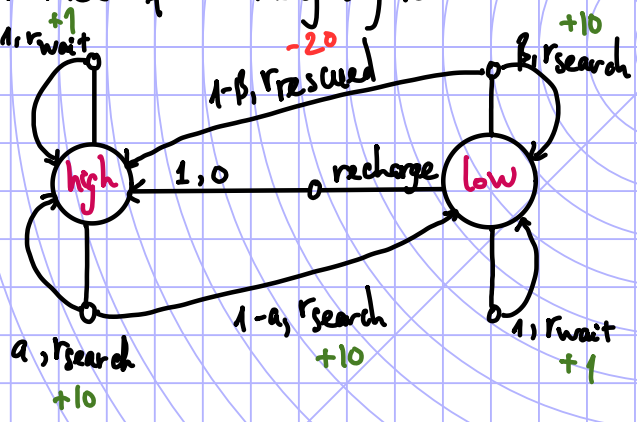
$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1$$

$\forall s \in \mathcal{S}, a \in \mathcal{A}(s)$

→ The present state contains all the info necessary to predict the future

- Examples of MDPs

↳ Dynamics of the Recycling Robot:



⇒ Sequential-making problems

→ MDP formalism is abstract and flexible

↳ Task: The goal of the robot is to pick-and-place objects

State : Latest readings of joint angles and velocities

Action : the amount of voltage of applied to each motor

Reward : +100 when an object is successfully placed

−1 for each unit of energy consumed

$G_5 = 0$ $G_4 =$

Goal of RL

→ How rewards relate to the goal of an agent

Goal of an Agent: Formal definition

return $G_t \doteq R_{t+1} + R_{t+2} + \dots$

⇒ Maximize the expected return

$$E[G_t] = E[R_{t+1} + R_{t+2} + \dots + R_T]$$

↓ final time step

The Reward Hypothesis

Whence Behavior?

→ Programming (GOFAI)
- Coding
- Human-in-the-loop

→ Examples (LfD)
- Mimic Reward
- Inverse RL

→ Optimization (RL)
- Evolutionary optimization
- Meta RL

• Challenges to H

→ Target
- risk-sensitive
- Diversity

→ High-level behavior
- Blind reward pursuer
- "purpose" over years, lifetimes

→ Identify episodic tasks

↳ In episodic tasks, the agent-env interaction breaks up into episodes

↳ Each episode begins independently of how the previous one ended

↳ At termination, the agent is reset to a start state

• Origin of the Hypothesis

Research program:

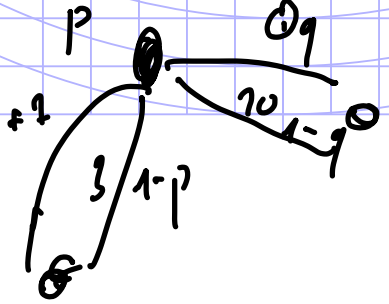
→ Identify where reward signals come from

→ Develop algos that search the space of behaviors to maximize reward signals

• Goals as Rewards

→ 1 for goal, 0 otherwise
goal-reward

→ -1 for not goal, 0 once goal reached
action-penalty



$$p + 3(1-p) = 10(1-q)$$

$$-2p + 3 = 10 - 10q$$

$\gamma = 0{,}8$. $\frac{1}{4} \cdot -21 + \frac{3}{4} \cdot (-1 + x) = x$

Continuing tasks

- Episodic vs. continuing tasks

- Interaction → episodes - Terminal state - Independent $G_t \doteq R_{t+1} + R_{t+2} + \dots + R_T$	- Goes on continually - No terminal state $G_t \doteq R_{t+1} + R_{t+2} + \dots = \infty$?

$\Rightarrow G_t = R_{t+1} + \gamma G_{t+1}$

E.g. games → Episodic tasks

Schedules → continuing tasks

↳ Discounting rate γ

↳ First → Formalize a MDP.

$G_0 = R_1 + \gamma G_1$

- Formulate returns using discounting
  - Discount the rewards in the future by γ
    $0 \leq \gamma < 1$

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots + \gamma^{k-1} R_{t+k} + \dots$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \Rightarrow \text{finite}$$

$$\leq \sum_{k=0}^{\infty} \gamma^k R_{max} = R_{max} \sum_{k=0}^{\infty} \gamma^k = R_{max} \cdot \frac{1}{1-\gamma}$$

- How returns at successive time steps are related to each other

$\gamma = 0$ ⇒ Short-sighted agent!

$G_t \doteq R_{t+1}$ (immediate reward)

$\gamma \to 1$ ⇒ Far-sighted agent!

↳ takes future rewards into account more strongly