Intro to RL + MDPs

- HW 2 should be out tonight.
- Haifeng is around.

What is a "finite horizon $\underline{H}$
episodic
Markov Decision Process"?

$H=1 \longleftrightarrow$ roughly contextual bandit.

What is such an MDP? H = horizon

$S = \{\text{set of states}\}$ $\Delta(S) = \{$set of probability distributions over $S\}$

$A = \{\text{set of actions}\}$

Time steps $h = 1$ to H.

$$P_h : S \times A \to \Delta(S)$$

$$R_h : S \times A \to \Delta(\mathbb{R})$$

$d_1 \in \Delta(S)$ distribution over initial states.

How do we "play" a single episode of the MDP?

User Selects a policy

$$\pi = (\pi_1, \dots, \pi_H)$$

$$\pi_h : S \longrightarrow \Delta(A)$$

from 1 to H,

Then: $s_1 \sim d_1$

We select action $a_1 \sim \pi_1(s_1)$

observe $r_1 \sim R_1(s_1, a_1)$

$s_2 \sim P_1(s_1, a_1)$

for $h = 2$ to H:

$a_h \sim \pi_h(s_h)$

$r_h \sim R_h(s_h, a_h)$

$s_{h+1} \sim P_h(s_h, a_h)$

Total reward in one episode is $\sum_{h=1}^{H} r_h$.

Play episodes $t=1, \ldots, T$.

For $t=1$ to T:

- User selects a policy π^t based off past experience.
- Play policy π^t in the MDP.
- Gain reward $\sum_{h=1}^{H} r_h^t$ ← episode t.

Total Reward $= \sum_{t=1}^{T} \sum_{h=1}^{H} r_h^t$ (goal: maximize this.)

Regret = Our total reward − $\mathbb{E}[$reward of $\pi^*]$

best policy

What is the "best" π^*?

$f(\pi) = \mathbb{E}$ reward under policy π (in one episode) $= \mathbb{E}_{\pi} \sum_{h=1}^{H} r_h$

$s_1 \sim d_1$
for $h = 1$ to H:
  $a_h \sim \pi_h(s_h)$
  $r_h \sim R_h(s_h, a_h)$
  $s_{h+1} \sim P_h(s_h, a_h)$.

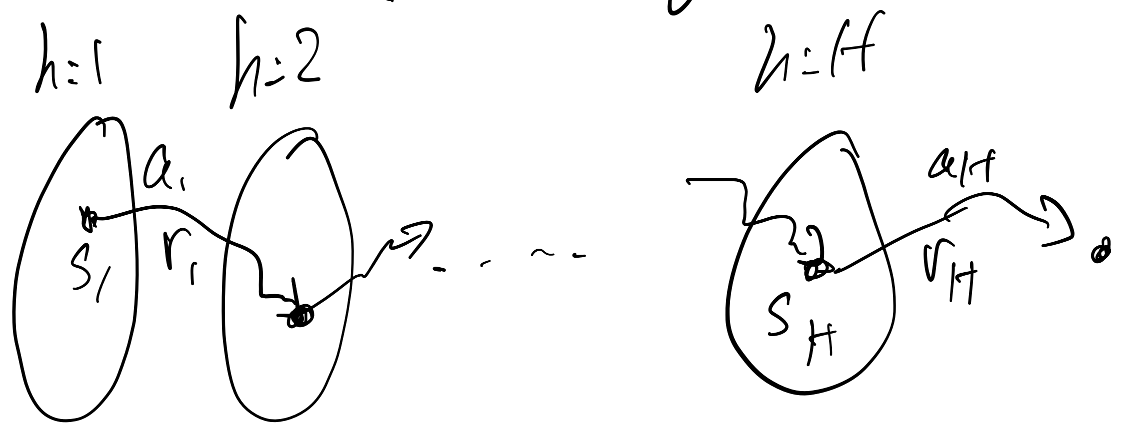
$\pi^* = \arg\max_{\pi} f(\pi)$.

Remark: Getting $o(T)$ regret "Online RL"

$\Updownarrow$

Finding a nearly optimal policy. "PAC-RL"

Next: Bellman equations / Dynamic Programming



Def Value function under policy π

$$V_h^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{\ell=h}^{H} r_\ell \,\middle|\, s_h = s\right]$$

Q-function under policy π

$$Q_h^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{\ell=h}^{H} r_\ell \,\middle|\, s_h = s, a_h = a\right]$$

Def: $V_h^*(s) = V_h^{\pi^*}(s)$

$Q_h^*(s,a) = Q_h^{\pi^*}(s,a)$.

$$Q_H^*(s,a) = \mathbb{E}[r_H \mid s_H = s, a_H = a]$$

$$V_H^*(s) = \mathbb{E}_{\pi^*}[r_H \mid s_H = s]$$

$$= \max_a \mathbb{E}[r_H \mid s_H = s, a_H = a]$$

$$= \max_a Q_H^*(s,a).$$

$$\pi_h^*(s) = \underset{a}{\operatorname{argmax}}\ Q_h^*(s,a)$$

$$V_h^*(s) = \max_a\ Q_h^*(s,a)$$

$$Q_h^*(s,a) = \mathop{\mathbb{E}}_{\pi^*}\left[\sum_{\ell=h}^{H} r_\ell \,\middle|\, s_h=s, a_h=a\right]$$

$$= \mathbb{E}\left[\begin{matrix} V_{h+1}^*(s_{h+1}) \\ + r_h \end{matrix} \,\middle|\, s_h=s, a_h=a\right]$$

"Bellman Equations"

Dynamic Programming algorithm to compute π^*, V^*, and Q^*:

("Value iteration")

(1) $$Q_H^*(s,a) = \mathbb{E}[r_H \mid s_H = s, a_H = a]$$

$$V_H^*(s) = \max_a Q_H^*(s,a)$$

$$\pi_H^*(s) = \arg\max_a Q_H^*(s,a)$$

(2) Use Bellman equations to define

$$Q^*_{H-1}, V^*_{H-1}, \pi^*_{H-1}$$

Then $Q^*_{H-2}, V^*_{H-2}, \pi^*_{H-2}, \ldots\ Q^*_1, V^*_1, \pi^*_1$.

II Failure of naive exploration.

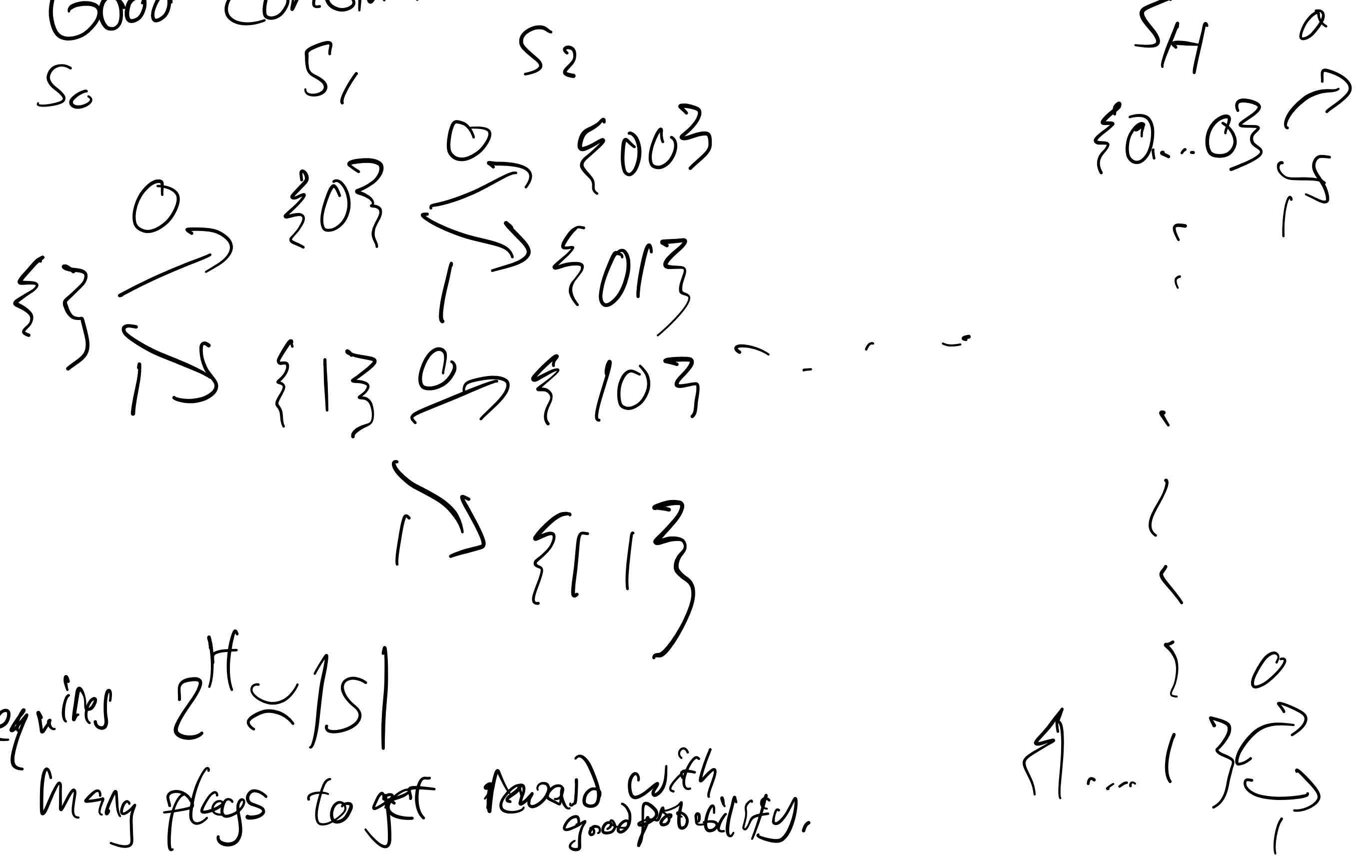
I Difficulty of large state spaces.

$r_h = 0$ $h < H$ password

$r_H = 1$ if you played 00110011 . . .

(or some other fixed string)

"Good combination lock"



Requires $2^H \simeq |S|$

many plays to get reward with good probability.

II "Bad combination lock"