Imitation Learning as Game-Solving

Gokul Swamy

Outline for Today

1. Why do we need interaction in imitation learning?

2. What else do we need to tell which mistakes matter?

3. How do we learn a policy that recovers from mistakes that matter if we don’t know what the reward function is?

Outline for Today

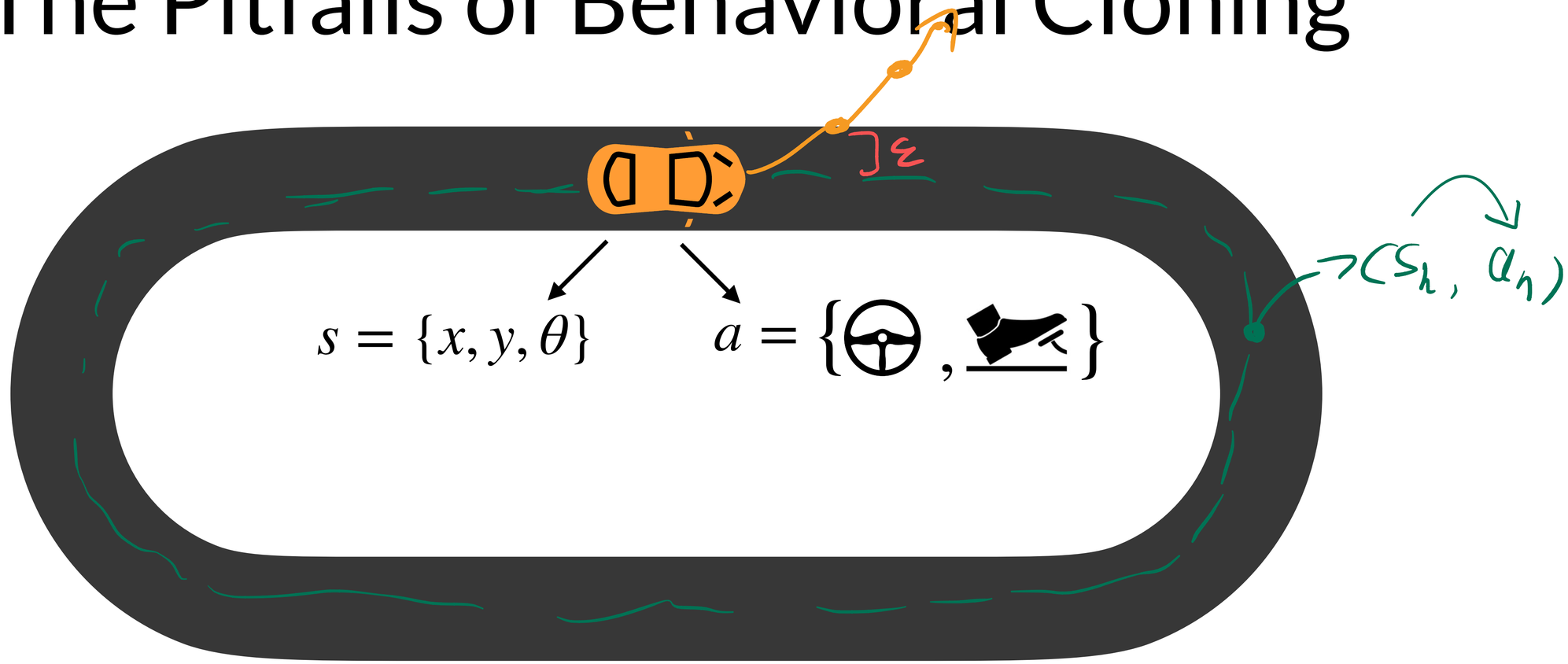
1. Why do we need interaction in imitation learning?

A: to be able to tell that we've made a mistake that compounds.

2. What else do we need to tell which mistakes matter?

3. How do we learn a policy that recovers from mistakes that matter if we don't know what the reward function is?

The Pitfalls of Behavioral Cloning



trajectory-level MSE

Gaussian policies => MSE

$$\arg\min_{\pi \in \Pi} \mathbb{E}_{\xi \sim \pi_E} \left[-\log \left(\prod_h^H \pi(a_h \mid s_h) \right) \right] = \arg\min_{\pi \in \Pi} \sum_h^H \mathbb{E}_{s_h, a_h \sim \pi_E} [-\log \pi(a_h \mid s_h)]$$

What went wrong?

Train Time: $\ell_{BC}(\pi) = \mathbb{E}_{s_h, a_h \sim \pi_E}\left[-\log \pi(a_h \mid s_h)\right]$

avg. over timesteps

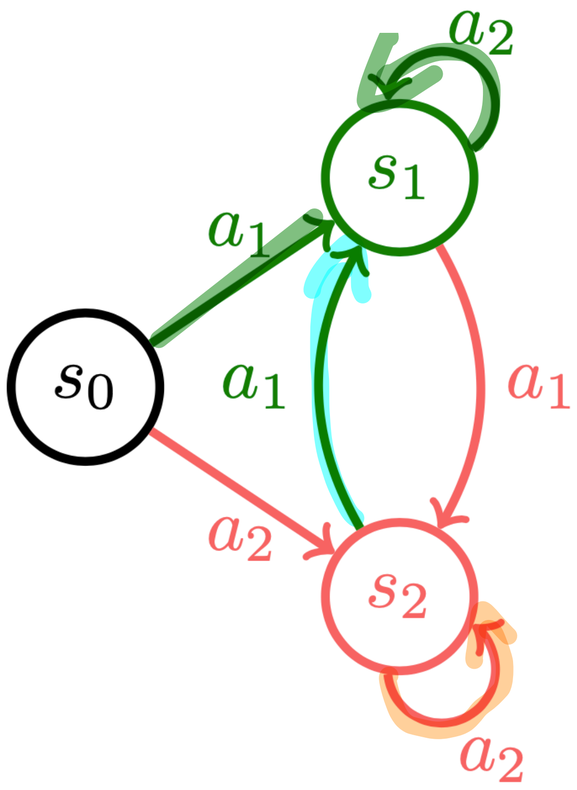
Test Time:

$\mathbb{E}_{s_h \sim \pi} \neq \mathbb{E}_{s_h \sim \pi_E}$

$P_{test}(x) \neq P_{train}(x)$

"covariate shift"

Covariate Shift ⇒ Compounding Errors



$$r(s,a) = \mathbf{1}[s = s_1]$$

$$\pi_1(s_0) = \pi_2(s_0) = [1,0]$$

$$\pi_1(s_1) = \pi_2(s_1) = [\varepsilon, 1-\varepsilon]$$

$$\pi_1(s_2) = [1,0] \quad \pi_2(s_2) = [0,1]$$

recover

doesn't recover

$$\ell_{BC}(\pi_1) = \ell_{BC}(\pi_2)$$

$$J(\pi_E, r) - J(\pi_1, r) = \varepsilon \cdot H$$

$$J(\pi_E, r) - J(\pi_2, r) = \varepsilon \cdot H^2$$

No offline IL algorithm can tell the difference between π_1 and π_2

What we talk about when we talk about ε

1. *Finite-sample error*: limited number of expert demos.

A: Get more data.

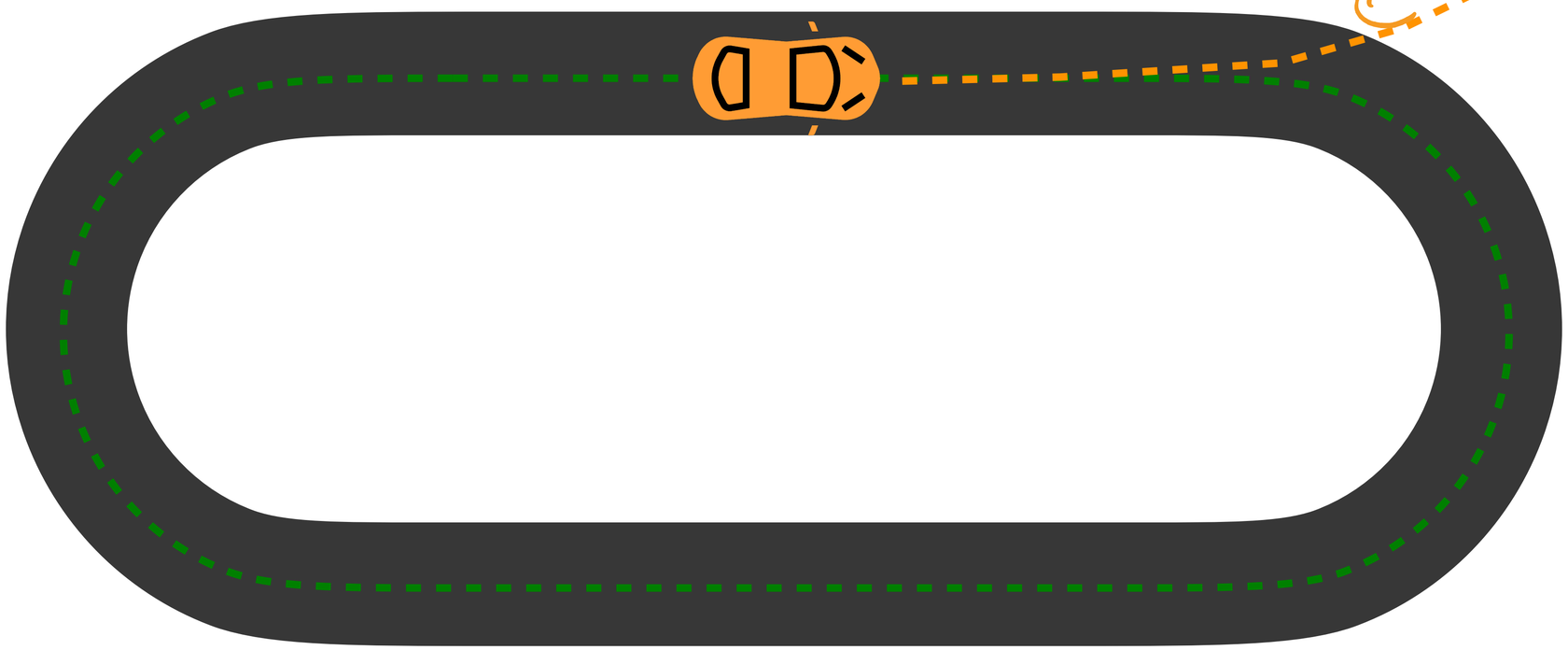
2. *Optimization error*: imperfect search over policy class.

A: Use more compute.

3. *Misspecification error:* irreducible error from $\pi_E \notin \Pi$.

A: Use an interactive algorithm.

Interaction Generates Samples from the Test Distribution



Outline for Today

1. Why do we need interaction in imitation learning?

A: to be able to tell that we've made a mistake that compounds.

2. What else do we need to tell which mistakes matter?

3. How do we learn a policy that recovers from mistakes that matter if we don't know what the reward function is?

Outline for Today

1. Why do we need interaction in imitation learning?

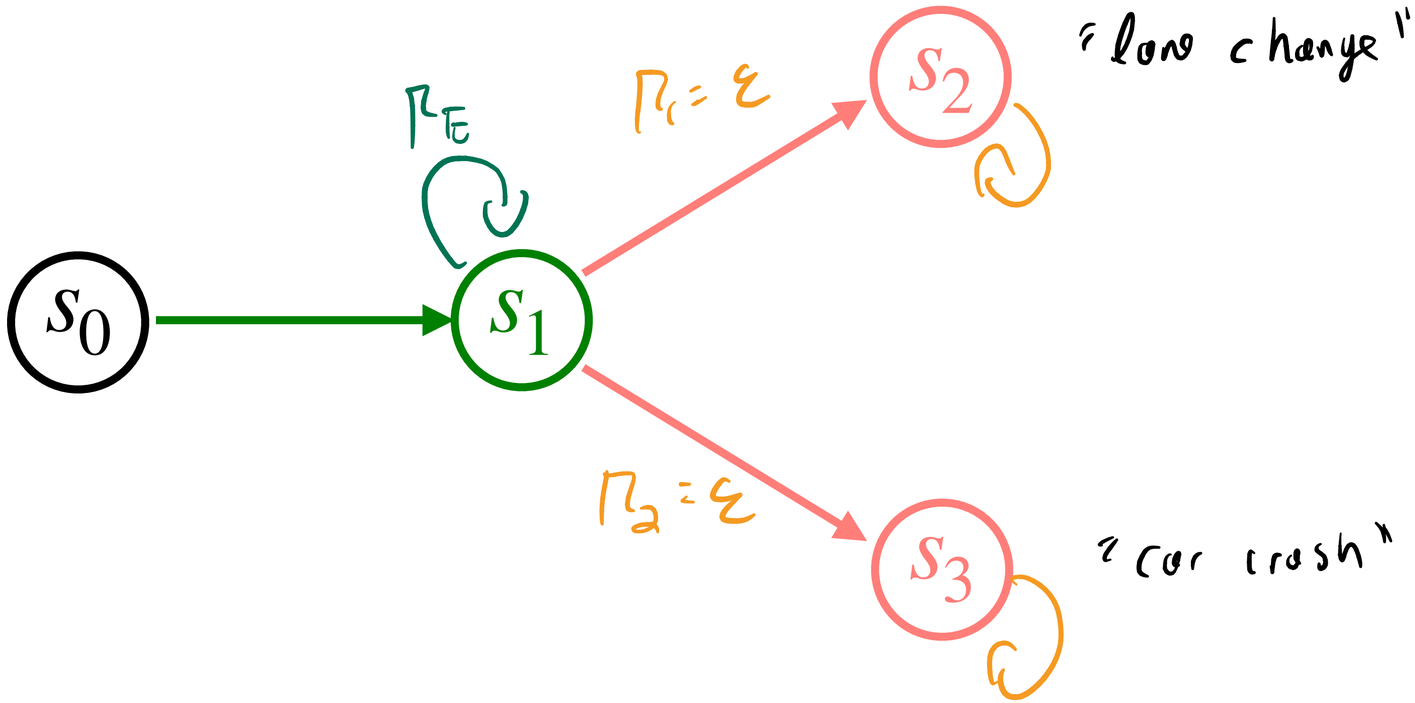
***A**: to be able to tell that we've made a mistake that compounds.*

2. What else do we need to tell which mistakes matter?

***A**: information about the **set** of rewards we could be judged on.*

3. How do we learn a policy that recovers from mistakes that matter if we don't know what the reward function is?

Not All Mistakes are Made Equal



We need to be able to tell which mistakes cost us performance.

Moments in Imitation Learning

$$\mathscr{R} = \{$$

- distance to nearest car
- distance to center of lane
- distance from edge of road
- speed / speed limit
- avoid difficult to drive on areas
- distance from nearest person
- distance to the goal

$$\}$$

Ok … but which $f \in \mathscr{R}$???

I'll even tell you that $r \in \mathscr{R}$.

🔑 Idea: Be good under *all* $f \in \mathscr{R}$!

Outline for Today

1. Why do we need interaction in imitation learning?

***A**: to be able to tell that we've made a mistake that compounds.*

2. What else do we need to tell which mistakes matter?

***A**: information about the **set** of rewards we could be judged on.*

3. How do we learn a policy that recovers from mistakes that matter if we don't know what the reward function is?

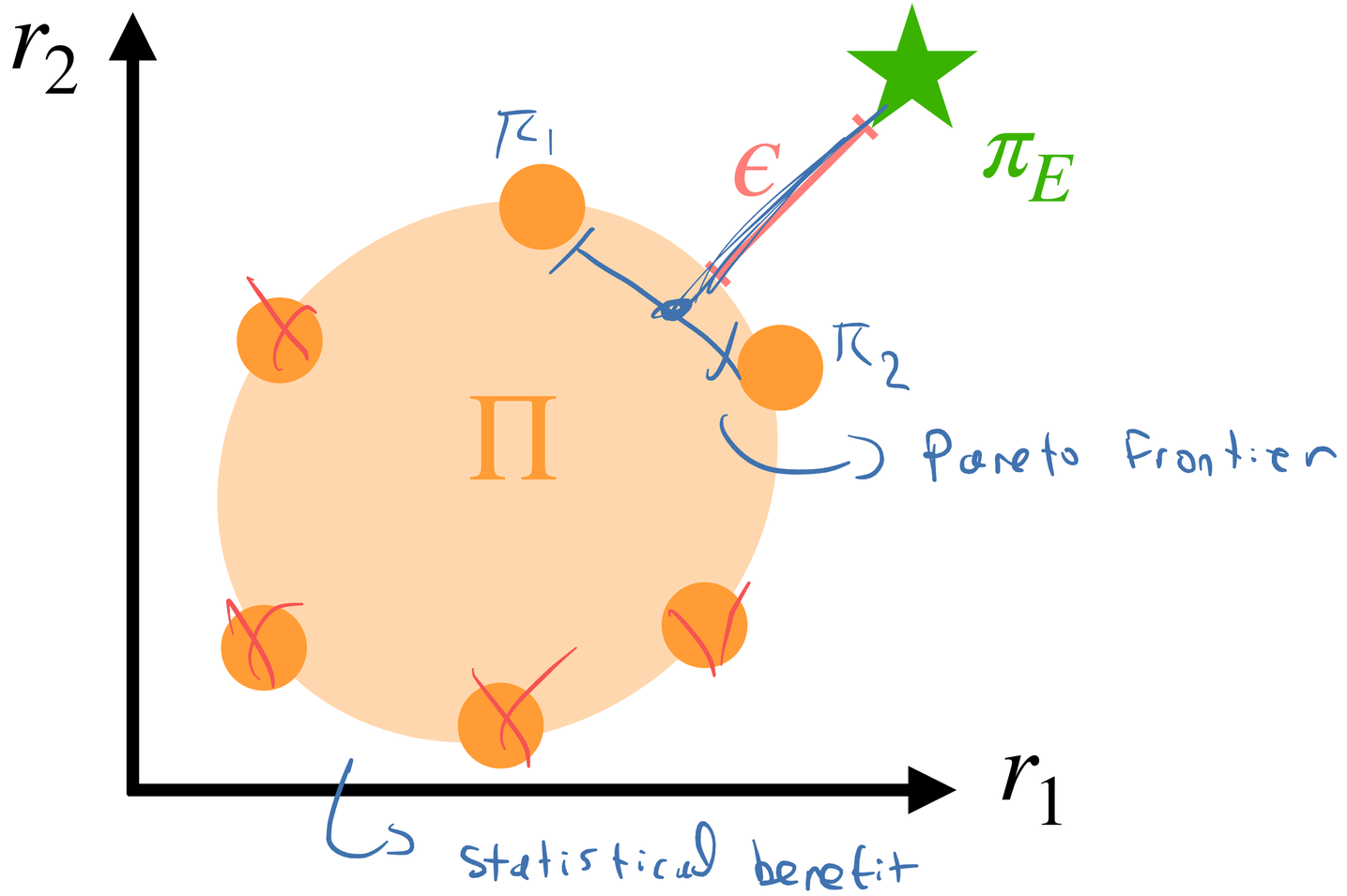
***A**: Find the policy that is the least distinguishable from the expert's under **any** reward function in the moment set $\mathscr{R}$.*

Inverse RL as Game-Solving

$$\max_{\pi \in \Pi} \min_{f \in \mathscr{R}} J(\pi, f) - J(\pi_E, f)$$

where $J(\pi, f) \triangleq \mathbb{E}_{\xi \sim \pi}\left[\sum_{h}^{H} f(s_h, a_h)\right].$

Game-Solving Searches the Pareto Frontier



Approx. Equilibria of IRL Game

Lemma: Assume $\hat{\pi}$ is an ε-approximate equilibria for the IRL game and for simplicity assume $\pi_E \in \Pi$. Then,

$$J(\pi_E, r) - J(\pi, r) \leq \mathcal{O}(\epsilon H)$$

$$\min_{f \in \mathscr{R}} \frac{J(\hat{\pi}, f) - J(\pi_E, f)}{H} \geq \varepsilon \quad \Longrightarrow \quad J(\pi_E, r) - J(\pi, r) \leq \mathcal{O}(\epsilon H)$$

$r \in R$

Inverse RL as Game-Solving

1. Inverse RL lets avoid compounding errors without needing access to extra expert interaction.
2. Inverse RL reduces the search space of policies to just those that are on the Pareto frontier.
3. Inverse RL isn't merely picking a reward that makes the expert look optimal — it is fundamentally game-theoretic.

How do we solve the IRL Game?

$$\max_{\pi \in \Pi} \min_{f \in \mathcal{R}} J(\pi, f) - J(\pi_E, f)$$

classifier

BR vs. NR = "Dual"

NR vs. NR = Primal

Policy Update:	RL	PG step
Reward Update:	OGD e.g. MaxEnt IRL	OGD e.g. GAIL

This is really just a GAN in the space of trajectories!

Take-aways from Today

1. Why do we need interaction in imitation learning?

A: to be able to tell that we've made a mistake that compounds.

2. What else do we need to tell which mistakes matter?

*A: information about the **set** of rewards we could be judged on.*

3. How do we learn a policy that recovers from mistakes that matter if we don't know what the reward function is?

*A: Find the policy that is the least distinguishable from the expert's under **any** reward function in the moment set $\mathscr{R}$.*

If ⏱️: MaxEnt Inverse RL

$$\min_{\pi} -\mathbb{H}(\pi)$$

$$\text{s.t. } \forall f \in \mathscr{R}, \quad \mathbb{E}_{\xi\sim\pi}\left[\sum_h^H f(s_h, a_h)\right] = \mathbb{E}_{\xi\sim\pi_E}\left[\sum_h^H f(s_h, a_h)\right]$$

"causal entropy"

"Augmented Lagrangian"

$$\max_{\lambda\in\mathbb{R}^{|\mathscr{R}|}} \min_{\pi} \mathbb{E}_{\xi\sim\pi}\left[\sum_h^H \log \pi(a_h \mid s_h)\right] + \sum_{f\in\mathscr{R}} \lambda^f (J(\pi, f) - J(\pi_E, f))$$

$-\lambda_t^2$

For some fixed λ_t, we can write the best-response over π as:

$$\min_{\pi} \mathbb{E}_{\xi\sim\pi}\left[\sum_{h}^{H} \log \pi(a_h \,|\, s_h)\right] + J\left(\pi, \sum_{f\in\mathscr{R}} \lambda_t^f f\right)$$

$$= \min_{\pi} \mathbb{E}_{\xi\sim\pi}\left[\sum_{h}^{H} \log \pi(a_h \,|\, s_h) + \sum_{f\in\mathscr{R}} \lambda_t^f f(s_h, a_h)\right]$$

$$r_t(s_h, a_h) \triangleq \log \pi(a_h \,|\, s_h) + \sum_{f\in\mathscr{R}} \lambda_t^f f(s_h, a_h)$$

Let us proceed by backwards-in-time induction over h:

Base Case ($h = H$):

$$V_t^\star(s_H) \triangleq 0$$

Inductive Step ($h \in [0, H-1]$):

$$\pi_t^\star(\,\cdot\,|\,s_h) = \min_{p \in \Delta(\mathscr{A})} \mathbb{E}_p \left[\log p(a) + \sum_{f \in \mathscr{R}} \lambda_t^f f(s_h, a) + \mathbb{E}_{T(s_h,a)}[V_t^\star(s_{h+1})] \right]$$

This is a single-step, action-level maximum entropy problem!

Recall that MaxEnt problems of the form :

$$\min_{p \in \Delta(\mathcal{X})} \; -\mathbb{H}(p) + \mathbb{E}_p[m(x)]$$

Have solutions of the form :

$$p^\star(x) = \frac{\exp(m(x))}{\sum_{x' \in \mathcal{X}} \exp(m(x'))}$$

Here, m is just :

$$m(a) = \sum_{f \in \mathcal{R}} \lambda_t^f f(s_h, a) + \mathbb{E}_{T(s_h,a)}[V_t^\star(s_{h+1})]$$

$$\pi_t^\star(a_h \mid s_h) = \frac{\exp\left(\sum_{f\in\mathscr{R}} \lambda_t^f f(s_h, a_h) + \mathbb{E}_{T(s_h,a_h)}[V_t^\star(s_{h+1})]\right)}{\sum_{a\in\mathscr{A}} \exp\left(\sum_{f\in\mathscr{R}} \lambda_t^f f(s_h, a) + \mathbb{E}_{T(s_h,a)}[V_t^\star(s_{h+1})]\right)}$$

$$V_t^\star(s_h) = \mathbb{E}_{a_h\sim\pi_t^\star(s_h)}[\log \pi_t^\star(a_h \mid s_h) + \lambda_t^f f(s_h, a_h) + \mathbb{E}_{T(s_h,a_h)}[V_t^\star(s_{h+1})]$$

Can solve for $\pi_t^\star$ via "soft" policy / value iteration!

Closely connected to Natural Policy Gradient and Hedge!