Imitation Learning

Why learn rewards?
→ to optimize what the person wants

otherwise: just copy the human
✓ → robot does things in human-like way
✓ → robot does thing when we don't know R/U
✓ → robot has model of π_H, the human policy

$\xi_D \sim$ rollout from π_D $\quad s_D^0, a_D^0, s_D^1, a_D^1, \ldots$

Behavior(al) Cloning: train π s.t. $\approx \pi_\theta(s) = \pi_D(s)$
parametrize π by θ, e.g. ANN

$$\max_\theta \pi_\theta(a_D \mid \xi_D) \Leftrightarrow \max_\theta \sum_i \log \pi_\theta(a_D^i \mid s_D^i)$$

$$\min_\theta \mathbb{E}_{s \sim \pi_D} [KL(\pi_D(a \mid s) \,\|\, \pi_\theta(a \mid s))]$$

$$\Leftrightarrow \min_\theta \sum_i - \sum_a \pi_D(a \mid s_D^i) \cdot \log \left(\frac{\pi_\theta(a \mid s_D^i)}{\pi_D(a \mid s_D^i)} \right)$$

↳ 1 for a_D^i? $\quad$ ↳ $\log \frac{x}{y} = \log x - \log y$

$$\approx \max_\theta \sum_i \log \pi_\theta(a_D^i \mid s_D^i)$$

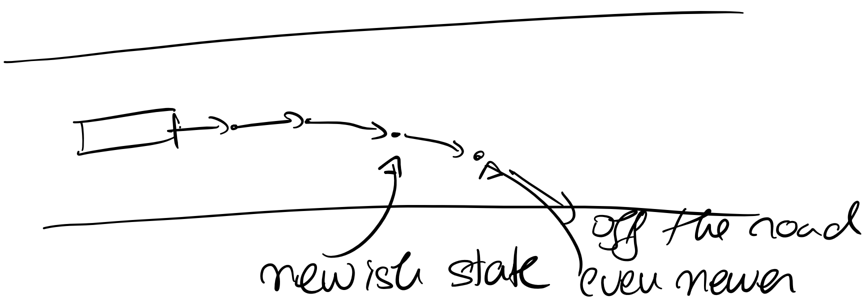
what's wrong w. BC?

↳ assumes samples are IID
but we are in a sequential domain!

⇓

error accumulation

ALVIN (CMU):



theoretical argument:

supervised learning has chance of ε error
if iid; T times: $T\varepsilon$ errors
BC is not iid: errors cumulate $\rightarrow T^2\varepsilon$

↳ once error $\rightarrow$ cost of 1 @ each remaining timestep

$$\varepsilon \cdot (T-1) + \varepsilon(T-2) + \dots + \varepsilon \cdot 1$$

fixes:

1) 2011 DAgger (Dataset Aggregation)

- fit π_θ on $\mathcal{D}$
- roll it out, collect induced states $s_{t=0}^{T}$
- ask for action labels $a_{t=0}^{T}$

  ↳ can people give you this?

- $\mathcal{D} \leftarrow \mathcal{D} \cup (s, a)$

get "on policy" labels

(turns out injecting noise in the demonstrator's actions to get them to go off and recover is probably enough (Dart '17)

2) practice with RL to "get back on" or "recover".

2016 GAIL (generative adversarial imi. lear.)

think of IRL as

$$\max_{c} \min_{\pi} \left(\mathbb{E}_{\pi}[c(s,a)] - \lambda H(\pi) \right) - \mathbb{E}_{\pi_D}[c(s,a)]$$

w/out regularization of $c \rightarrow \pi$ matches demonst. state-action occupancy

idea: search for a π that does that more directly

$D(s,a) = 0$ if (s,a) came from π_D, 1 else

$\log D(s,a) = -\infty$ if (s,a) came from π_D, 0 else

train π to minimize D; train D to be low given π_D

but high otherwise:

$$\max_D \min_\pi \left(\mathbb{E}_\pi [\log D(s,a)] - \lambda H(\pi) \right) - \mathbb{E}_{\pi_D} [\log(1 - D(s,a))]$$

iterate between gradient on D

and RL update on π