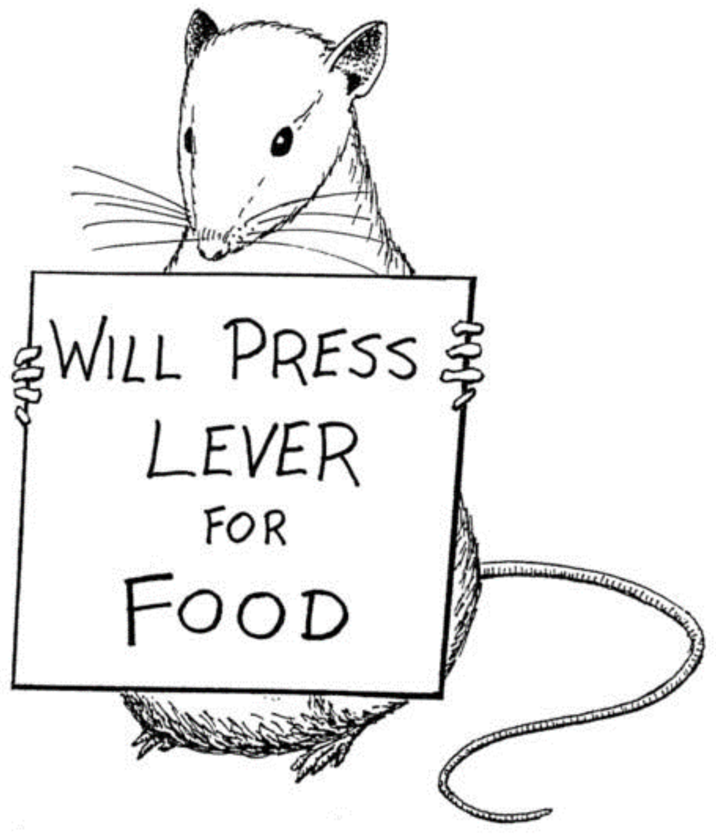
UCI University of California, Irvine

CS 277: Control and Reinforcement Learning

Winter 2022

Lecture 14: Bounded RL

Roy Fox
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine



Logistics

quizzes

- Quiz 5 is due tomorrow

assignments

- Assignment 4 to be published soon

Today's lecture

MaxEnt IRL

GAIL

Reward shaping

Bounded RL

Informational quantities: refresher

- Entropy: $\mathbb{H}[p(a)] = -\mathbb{E}_{a\sim p}[\log p(a)] = -\sum_a p(a)\log p(a)$
- Conditional entropy: $\mathbb{H}[\pi|s] = -\mathbb{E}_{a\sim\pi}[\log \pi(a|s)]$
- Expected conditional entropy: $\mathbb{H}[\pi] = \mathbb{E}_{s\sim p_\pi}[\mathbb{H}[\pi|s]] = -\mathbb{E}_{(s,a)\sim p_\pi}[\log \pi(a|s)]$
- Expected relative entropy: $\mathbb{D}[\pi\|\pi'] = \mathbb{E}_{(s,a)\sim p_\pi}\left[\log \frac{\pi(a|s)}{\pi'(a|s)}\right]$
- Expected cross entropy (aka NLL): $-\mathbb{E}_{(s,a)\sim p_\pi}[\log \pi'(a|s)]$
  - $\mathbb{D}[\pi\|\pi'] = \text{NLL} - \mathbb{H}[\pi]$

Modeling bounded teachers

- An expert teacher maximizes the value $J_\theta^{\pi*} = \sum_t \gamma^t \mathbb{E}_{s_t \sim p*}[\theta^\intercal f_{s_t}] = \mathbb{E}_{\xi \sim p*}[\theta^\intercal f_\xi]$
  - With trajectory-summed features $f_\xi = \sum_t \gamma^t f_{s_t}$
- Assume teacher has unintentional / uninformed prior policy π_0
  - Bounded rationality: cost to intentionally diverge $\mathbb{D}[\pi^* \| \pi_0]$ (with π_0 uniform: $\mathbb{H}[\pi^*]$)
  - Total cost: $\sum_t \mathbb{E}_{(s_t,a_t)\sim p*}\left[\log \frac{\pi^*(a_t | s_t)}{\pi_0(a_t | s_t)}\right] = \mathbb{E}_{\xi \sim p*}\left[\log \frac{p^*(\xi)}{p_0(\xi)}\right] = \mathbb{D}[p^*(\xi) \| p_0(\xi)]$
- Bounded optimality: $\max_{\pi*} \mathbb{E}_{\xi \sim p*}[\theta^\intercal f_\xi] - \tau \mathbb{D}[p^* \| p_0]$

Bounded optimality: naïve solution

- Bounded optimality: $\max_{\cancel{\pi^*}\, p^*} \mathbb{E}_{\xi \sim p^*}[\theta^\intercal f_\xi] - \mathbb{D}[p^* \| p_0]$
    - Naïve solution: allow any distribution p^* over trajectories
    - No need to be consistent with dynamics $p(s' \mid s, a) \Rightarrow p^*$ may be unachievable
- Add the constraint $\sum_\xi p^*(\xi) = 1$ with Lagrange multiplier λ
- Differentiate by $p^*(\xi)$ and $= 0$ to optimize

$$\theta^\intercal f_\xi - \log p^*(\xi) + \log p_0(\xi) - 1 + \lambda = 0 \Longrightarrow p^*(\xi) = \frac{p_0(\xi)\exp(\theta^\intercal f_\xi)}{\sum_{\bar{\xi}} p_0(\bar{\xi})\exp(\theta^\intercal f_{\bar{\xi}})}$$

IRL with bounded teacher

- Assume that demonstrations are distributed $p_\theta(\xi) = \frac{1}{Z_\theta} p_0(\xi) \exp(\theta^\intercal f_\xi)$
  - With partition function $Z_\theta = \mathbb{E}_{\bar{\xi} \sim p_0}[\exp(\theta^\intercal f_{\bar{\xi}})]$
- Find θ that minimizes NLL of demonstrations

$$\nabla_\theta \log p_\theta(\xi) = \nabla_\theta(\theta^\intercal f_\xi - \log Z_\theta) = f_\xi - \frac{1}{Z_\theta} \nabla_\theta Z_\theta$$

$$= f_\xi - \frac{1}{Z_\theta} \mathbb{E}_{\bar{\xi} \sim p_0}[\exp(\theta^\intercal f_{\bar{\xi}}) f_{\bar{\xi}}] = f_\xi - \mathbb{E}_{\bar{\xi} \sim p_\theta}[f_{\bar{\xi}}]$$

  - To compute gradient, we need p_θ, but how to compute Z_θ?

Computing Z_θ: backward recursion

- Partition function: $Z_\theta = \mathbb{E}_{\xi \sim p_0}[\exp(\theta^\intercal f_\xi)]$

- Compute Z_θ recursively backward: like a value function, but + becomes ·

$$Z_\theta(s_t, a_t) = \mathbb{E}_{p_0}[\exp(\theta^\intercal f_{\xi \geq t}) \,|\, s_t, a_t] = \exp(\theta^\intercal f_{s_t}) \mathbb{E}_{(s_{t+1}|s_t, a_t) \sim p}[Z_\theta(s_{t+1})]$$

$$Z_\theta(s_t) = \mathbb{E}_{p_0}[\exp(\theta^\intercal f_{\xi \geq t}) \,|\, s_t] = \mathbb{E}_{(a_t|s_t) \sim \pi_0}[Z_\theta(s_t, a_t)]$$

- How to get a policy from Z_θ?
  - Marginalize: $\pi_\theta(a_t \,|\, s_t) = \dfrac{p_\theta(\xi \,|\, s_t, a_t)}{p_\theta(\xi \,|\, s_t)} = \pi_0(a_t \,|\, s_t) \dfrac{Z_\theta(s_t, a_t)}{Z_\theta(s_t)}$

    consistent π may not even exist
  - This π_θ is not globally consistent $p_\theta(\xi) \neq p_{\pi_\theta}(\xi)$, $p_\theta(\xi)$ ignores the dynamics

MaxEnt IRL

- For each sample $\xi \sim \mathscr{D}$:
  - Compute $Z_\theta = \mathbb{E}_{\xi \sim p_0}[\exp(\theta^\intercal f_\xi)]$ recursively backward
  - Compute $\mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}}[f_{\bar{\xi}}]$ recursively forward
  - Take a gradient step to improve θ: $\nabla_\theta \log p_\theta(\xi) \approx f_\xi - \mathbb{E}_{\bar{\xi} \sim p_{\pi_\theta}}[f_{\bar{\xi}}]$
- At the optimum: feature matching $\mathbb{E}_{\xi \sim \mathscr{D}}[f_\xi] = \mathbb{E}_{\xi \sim p_{\pi_\theta}}[f_\xi]$
  - MaxEnt IRL approximates $\max_\theta \mathbb{H}[\pi_\theta]$ s.t. $\mathbb{E}_{\xi \sim \mathscr{D}}[f_\xi] = \mathbb{E}_{\xi \sim p_{\pi_\theta}}[f_\xi]$

Limitations:

- Requires dynamics p
- Assumes $p_\theta = p_{\pi_\theta}$
- Assumes $\mathscr{D} = p^*$

Today's lecture

MaxEnt IRL

GAIL

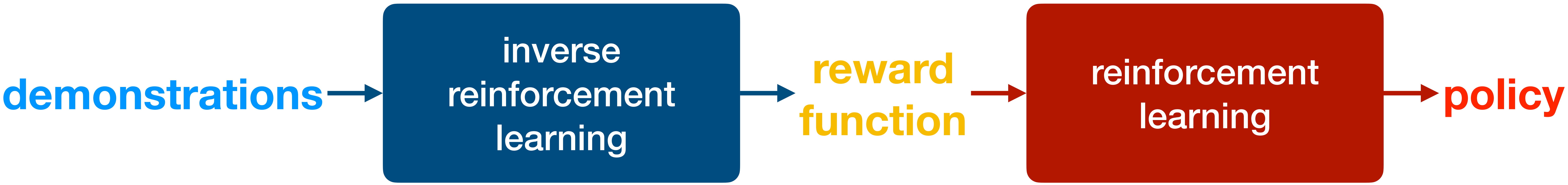
Reward shaping

Bounded RL

IRL: downstream tasks

- One IRL motivation: learn reward function for downstream tasks

...such as RL



- IL = RL ○ IRL (composition of RL on IRL)
- Some IRL algorithms already learn π as part of learning θ for $r : s \mapsto \theta^\intercal f_s$
  - Let's directly optimize IRL for the overall IL task = learn good π

IL as RL ∘ IRL

- Entropy-regularized RL: $\max_{\pi \in \Pi} \left\{ \mathbb{E}_{s \sim p_\pi}[r(s)] + \mathbb{H}[\pi] \right\}$

regularization over reward function space

- MaxEnt IRL: $\max_{r \in \mathbb{R}^{\mathcal{S}}} \left\{ \mathbb{E}_{s \sim p^*}[r(s)] - \max_{\pi \in \Pi} \left\{ \mathbb{E}_{s \sim p_\pi}[r(s)] + \mathbb{H}[\pi] \right\} \right\} - \psi(r)$

- For any π, our objective with respect to r is:

$$\hat{\psi}(p^* - p_\pi) = \max_{r \in \mathbb{R}^{\mathcal{S}}} \left\{ \langle \overbrace{p^* - p_\pi}^{\in \mathbb{R}^{\mathcal{S}}}, r \rangle - \psi(r) \right\}$$

  - This form of function $\hat{\psi} : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}$ is called the convex conjugate of ψ

Reward-function regularizers

$$\hat{\psi}(p^* - p_\pi) = \max_{r \in \mathbb{R}^{\mathcal{S}}} \left\{ \langle p^* - p_\pi, r \rangle - \psi(r) \right\}$$

- Without regularizer: $\psi = 0 \Rightarrow$ solution only exists when $p^* = p_\pi$
  - Learner achieves teacher's state distribution: perfect solution, but hard to find
- Hard linearity constraint: $\psi(r) = \begin{cases} 0 & \text{if } r(s) = \theta^\intercal f_s \\ \infty & \text{otherwise} \end{cases}$
  - Max-entropy feature matching (MaxEnt IRL)
  - Great when the reward function really is linear in f_s, otherwise no guarantees

Generative Adversarial Networks (GANs)

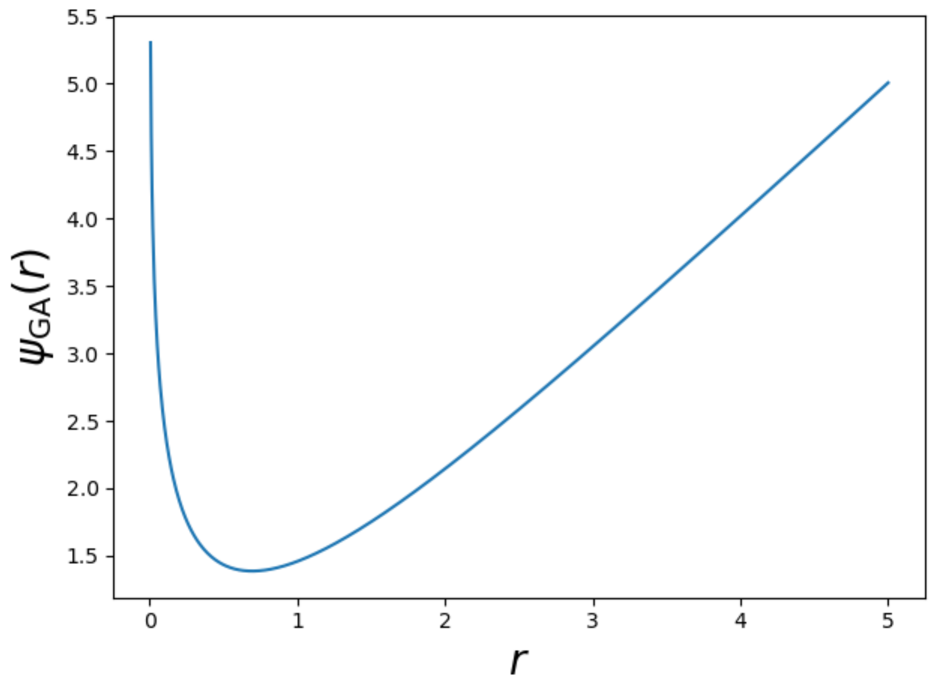
- Train generative model $p_\theta(s)$ to generate states / observations
  - Can we focus the training on failure modes?

- Also train discriminator $D_\phi(s) \in [0,1]$ to score instances
  - Kind of like a critic: are generated instances good?
- $D_\phi(s)$ predicts the probability $p(s \text{ generated by learner} \mid s) = \dfrac{p_\theta(s)}{p_\theta(s) + p^*(s)}$
  - Trained with cross-entropy loss: $\max_\phi \left\{ \mathbb{E}_{s \sim p_\theta}[\log D_\phi(s)] + \mathbb{E}_{s \sim p^*}[\log(1 - D_\phi(s))] \right\}$
- The generator tries to fool the discriminator: $\min_\theta \mathbb{E}_{s \sim p_\theta}[\log D_\phi(s)]$

Teacher-based reward-function regularizer

- Consider the regularizer

$$\psi_{\text{GA}}(r) = \mathbb{E}_{s\sim p^*}[r(s) - \log(1 - \underbrace{\exp(-r(s))}_{D(s)})]$$



- It's convex conjugate is:

$$\hat{\psi}_{\text{GA}}(p^* - p_\pi) = \max_{r\in\mathbb{R}^{\mathcal{S}}} \left\{\langle p^* - p_\pi, r\rangle - \psi_{\text{GA}}(r)\right\}$$
$$= \max_{r\in\mathbb{R}^{\mathcal{S}}} \mathbb{E}_{s\sim p^*}[r(s) - r(s) + \log(1 - D(s))] - \mathbb{E}_{s\sim p_\pi}[\overbrace{r(s)}^{-\log D(s)}]$$
$$= \max_{r\in\mathbb{R}^{\mathcal{S}}} \mathbb{E}_{s\sim p_\pi}[\log D(s)] + \mathbb{E}_{s\sim p^*}[\log(1 - D(s))]$$

  - This is a GAN: generator p_π imitating teacher p^*; discriminator $D(s) = \exp(-r(s))$

Generative Adversarial Imitation Learning (GAIL)

Algorithm GAIL

Input: demonstration dataset $\mathcal{D} \sim p^*$

Initialize policy π_θ, discriminator D_ϕ

repeat

$\xi \leftarrow$ roll out π_θ

Ascend $\mathcal{L}_\phi(\xi) = \mathbb{E}_{s\sim\xi}[\log D_\phi(s)] + \mathbb{E}_{s\sim\mathcal{D}}[\log(1 - D_\phi(s))]$

Improve π_θ with entropy-regularized PG, $r(s) = -\log D_\phi(s)$

- We've already seen entropy-regularized PG algorithms: TRPO, PPO
  - More later

Recap

- To understand behavior: infer the intentions of observed agents
- If teacher is optimal for a reward function
  - The reward function should make an optimizer imitate the teacher
  - State (or state–action) distribution of learner should match the teacher
- In this view, Inverse Reinforcement Learning (IRL) is a game:
  - Reward is optimized to show how much the teacher is better than the learner
  - Learner optimizes for the reward
  - Reward is like a discriminator (high = probably teacher); learner like a generator

Today's lecture

MaxEnt IRL

GAIL

Reward shaping

Bounded RL

Relation between RL and IL

- What makes RL harder than IL?
  - IL: teacher policy $\pi^*(a \mid s)$ indicates a good action to take in s
  - RL: $r(s, a)$ does not indicate a globally good action; $Q^*(s, a)$ does, but it's nonlocal
- But didn't we see an equivalence between RL and IL?
  - NLL loss in BC: $\mathbb{E}_{(s,a)\sim p^*}[\nabla_\theta \log \pi_\theta(a \mid s)]$
    - s and a sampled from teacher distribution, this could make IL harder than RL
  - Policy Gradient: $\mathbb{E}_{(s,a)\sim p_\theta}[R \nabla_\theta \log \pi_\theta(a \mid s)]$
    - s and a sampled from learner distribution

IL as sparse-reward RL

- NLL BC: maximize $\mathbb{E}_{(s,a)\sim p^*}[\log \pi_\theta(a\,|\,s)] = -\,\mathbb{D}[\pi^*\|\pi_\theta] - \mathbb{H}[\pi^*]$

  constant in θ

  - Experience from teacher distribution p^*
    - RL: experience from learner distribution p_θ
  - Pseudo-return $R = 1_{\text{success}}$ for successful trajectory
    - RL: $r_t = r(s_t, a_t)$ in every step
- Sparse reward = most rewards are 0 / constant ⇒ rare learning signal
  - $R = 1$ on success ⇒ very sparse; but doesn't IL provide dense learning signal?

IL as dense-reward RL

- What if instead we minimize the other relative entropy?

teacher labeling of learner states/actions
as in DAgger

$$\mathbb{D}[\pi_\theta \| \pi^*] = -\mathbb{E}_{(s,a)\sim p_\theta}[\log \pi^*(a \mid s)] - \mathbb{H}[\pi_\theta]$$

  - This is exactly the RL objective, with $r(s,a) = \log \pi^*(a \mid s)$, entropy regularizer
  - Now $r(s,a)$ does give global information on optimal action
  - In fact, with deterministic teacher, $r(s,a) = -\infty$ for any suboptimal action
- The same return can be viewed as sum of sparse rewards, or dense
  - How should we design r for easy RL?

Reward shaping

- Ideal reward: $r(s, a) = -\infty$ for any suboptimal action $\Rightarrow$ as hard to provide as π^*
    - We need supervision signal that's sufficiently easy to design + program
- Sparse reward functions may be easier to design than dense ones
    - E.g., may be easy to identify good goal states, safety violations, etc.
- Reward shaping: art of adjusting the reward function for easier RL; some tips:
    - Reward "bottleneck states": subgoals that are likely to lead to bigger goals
    - Break down long sequences of coordinated actions $\Rightarrow$ better exploration
        - E.g. reward beacons on long narrow paths, for exploration to stumble upon

Today's lecture

MaxEnt IRL

GAIL

Linearly solvable MDPs

Bounded RL

Bounded optimality

- Bounded optimizer = trades off value and divergence from prior $\pi_0(a \mid s)$

$$\max_\pi \mathbb{E}_{(s,a)\sim p_\pi}[r(s,a)] - \tau\mathbb{D}[\pi \| \pi_0] = \max_\pi \mathbb{E}_{(s,a)\sim p_\pi}\left[\beta r(s,a) - \log\frac{\pi(a \mid s)}{\pi_0(a \mid s)}\right]$$

- $\beta = \frac{1}{\tau}$ is the tradeoff coefficient between value and relative entropy
  - Similar to the inverse-temperature in thermodynamics
  - As $\beta \to 0$, the agent will fall back to the prior $\pi \to \pi_0$
  - As $\beta \to \infty$, the agent will be a perfect value optimizer $\pi \to \pi^*$
- We'll see reasons to have finite β

Simplifying assumption

- MaxEnt IRL was approximate because it violated dynamical constraints
  - $p_\pi(\xi) \propto \pi_0(\xi)\exp(R(\xi))$, regardless of trajectory feasibility
- For simplicity, let's do the same for RL
  - Suppose the environment is fully controllable $s_{t+1} = a_t$
  - Bellman equation:

$$V^*_\beta(s) = \max_\pi \mathbb{E}_{(s'|s)\sim\pi}\left[r(s) - \frac{1}{\beta}\log\frac{\pi(s'|s)}{\pi_0(s'|s)} + \gamma V^*_\beta(s')\right]$$

$$= r(s) - \frac{1}{\beta}\min_\pi \mathbb{D}\left[\pi \,\middle\|\, \frac{\pi_0(s'|s)\exp(\beta\gamma V^*_\beta(s'))}{Z'_\beta(s)}\right] + \frac{1}{\beta}\log Z'_\beta(s)$$

Linearly-Solvable MDPs (LMDPs)

- Optimal policy for $V_\beta(s) = r(s) - \frac{1}{\beta} \min_\pi \mathbb{D}\left[\pi \,\middle\|\, \frac{\pi_0(s'|s)\exp(\beta\gamma V_\beta(s'))}{Z'_\beta(s)}\right] + \frac{1}{\beta}\log Z'_\beta(s)$:
  - Soft-greedy policy: $\pi_\beta(s'|s) \propto \pi_0(s'|s)\exp(\beta\gamma V_\beta(s'))$
- Value recursion: $V_\beta(s) = r(s) + \frac{1}{\beta}\log Z'_\beta(s) = r(s) + \frac{1}{\beta}\log \mathbb{E}_{(s'|s)\sim\pi_0}[\exp(\beta\gamma V_\beta(s'))]$

$$Z_\beta(s) = \exp(\beta V_\beta(s)) = \exp(\beta r(s))Z'_\beta(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma_\beta(s')]$$

- In the undiscounted case $\gamma = 1$, with $D = \mathrm{diag}(\exp \beta r)$: $z = DP_0 z$
- We can solve for z, and therefore π, by finding a right-eigenvector of DP_0

Z-learning

$$Z(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma(s')]$$

- We can do the same model-free:
  - Given experience (s, r, s') sampled by the prior policy π_0
  - Update $Z(s) \to \exp(\beta r)Z^\gamma(s')$
- Full-controllability condition $(s_{t+1} = a_t)$ can be relaxed to allow $\pi_0(s' | s) = 0$
  - But we still allow any transition distribution $\pi(s' | s)$ over the remaining support
  - Later: the general case, $p(s' | s) = \sum_a \pi(a | s)p(s' | s, a)$

Duality between value and log prob

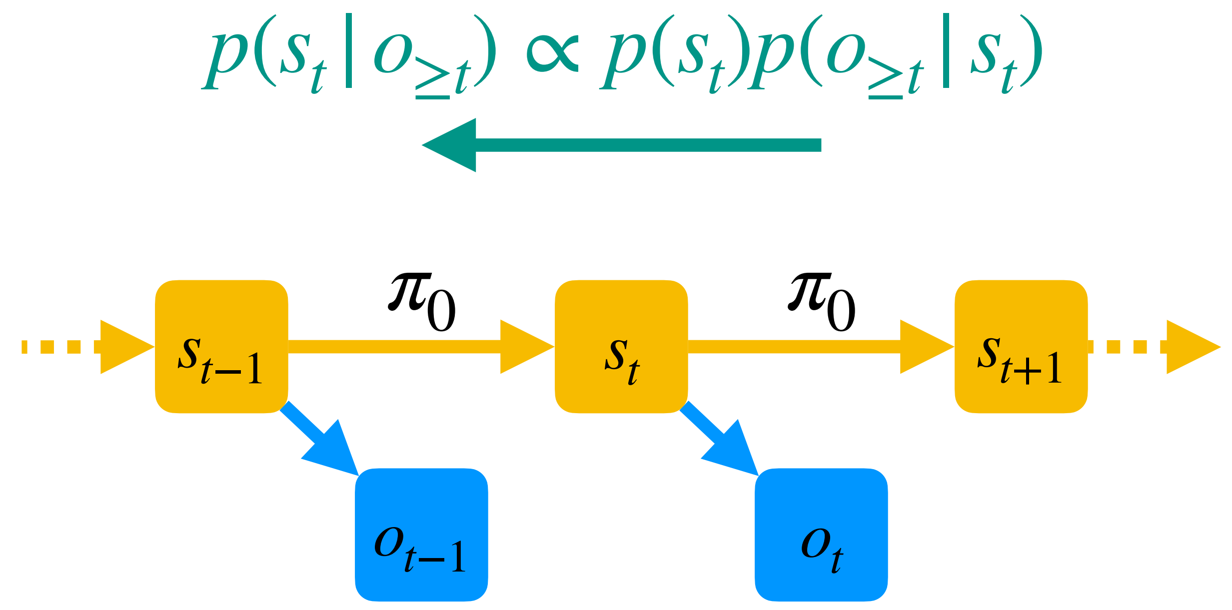
- We've seen many cases where log-probs play the role of reward / value
  - Or values the role of logits (unnormalized log-probs)
- Examples:
  - In LQG, $\log p(x \mid \hat{x}) = -\frac{1}{2} x^\intercal \Sigma x + \text{const}$; costs / values are quadratic
  - In value-based algorithms, good exploration policy: $\pi(a \mid s) = \operatorname*{softmax}_a \beta Q(s, a)$
  - Imitation Learning can be viewed as RL with $r(s, a) = \log p^*(a \mid s)$
  - In IRL, a reward function can be viewed as a discriminator $D(s) = \exp(-r(s))$

Full-controllability duality

- Bounded control in LMDP: $Z(s) = \exp(\beta r(s))\mathbb{E}_{(s'|s)\sim\pi_0}[Z^\gamma(s')]$

- Backward filtering in a partially observable system with dynamics $\pi_0(s'|s)$

$$p(o_{\geq t}|s_t) = p(o_t|s_t)\mathbb{E}_{(s_{t+1}|s_t)\sim\pi_0}[p(o_{\geq t+1}|s_{t+1})]$$



- Equivalent if $Z(s) = p(o_{\geq t}|s_t)$ and $\exp(\beta r(s)) = p(o|s)$
  - Intuition: find states that give good reward ⇔ high likelihood of observations

- Exact equivalence only in the fully-controllable case
  - Partially controllable case takes more nuanced analysis