MS Thesis - I
Inverse Constrained Reinforcement Learning

Usman Anwar
MSDS19001

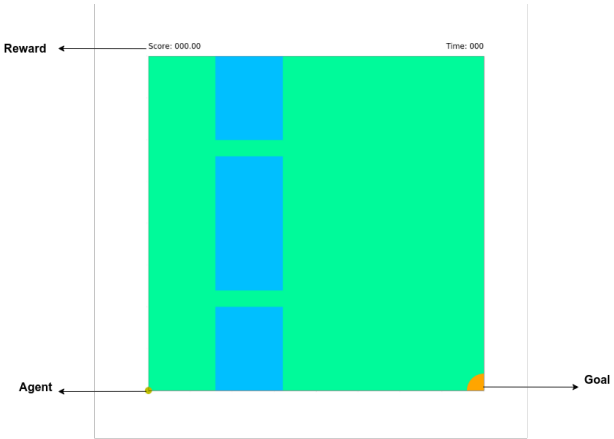
Advisor: Dr. Ali Ahmed.

Information Technology University, Lahore.

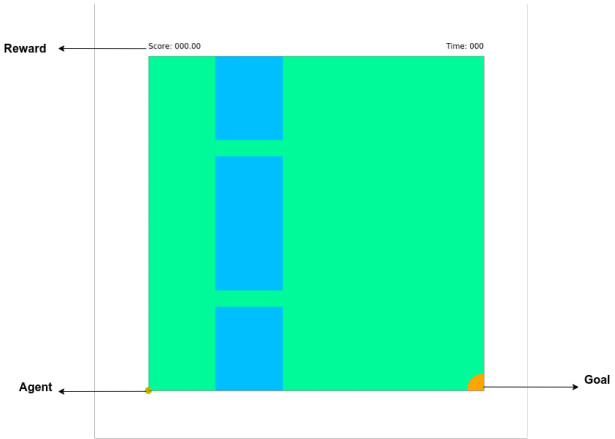
February 24, 2021

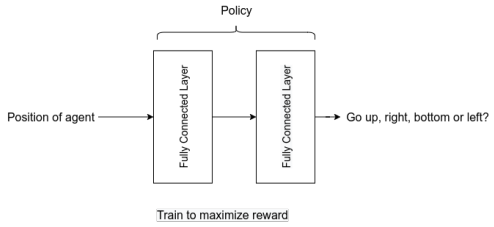
Motivation & Introduction

Reinforcement Learning (RL): Basic Setup



Reinforcement Learning (RL): Basic Setup





Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)

Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)
- A = actions (up, down, right, left)

Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)
- A = actions (up, down, right, left)
- r = reward (score)

Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)
- A = actions (up, down, right, left)
- r = reward (score)
- $\mathcal{T}$ = transition function (next state given current state and action)

Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)
- A = actions (up, down, right, left)
- r = reward (score)
- $\mathcal{T}$ = transition function (next state given current state and action)

Total reward: $r = \sum_{t=1}^{T} r(s_t, a_t)$.

Reinforcement Learning (RL): Notation & Formal Definition

A Markov Decision Process (MDP) contains

- S = states ((x, y) coordinates)
- A = actions (up, down, right, left)
- r = reward (score)
- $\mathcal{T}$ = transition function (next state given current state and action)

Total reward: $r = \sum_{t=1}^{T} r(s_t, a_t)$.

Goal is to find policy $\pi : S \mapsto A$ that maximizes r.

Reward Specification Is Hard!

https://youtu.be/tlOIHko8ySg

Observation 1

Reward must provide two types of signals:

- Must ‘encourage’ good or desired behaviour.
- Must ‘discourage’ bad or undesired behaviours.

Observation 1

Reward must provide two types of signals:

- Must ‘encourage’ good or desired behaviour.
- Must ‘discourage’ bad or undesired behaviours.
  - We can model these as constraints.

Constrained RL

Augment MDP with cost function c and budget α and solve:

$$\underset{\pi}{\text{maximize}}\ \mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t r(s_t,a_t)\right]\quad \text{subject to}\ \mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t c(s_t,a_t)\right]\leq\alpha$$

Constrained RL

Augment MDP with cost function c and budget α and solve:

$$\underset{\pi}{\text{maximize}}\ \mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t r(s_t,a_t)\right]\quad\text{subject to}\ \mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t c(s_t,a_t)\right]\leq\alpha$$

Algorithm:

Primal-Dual Optimization [1]: Construct Lagrangian and solve via dual gradient descent:

$$\min_{\lambda\geq 0}\max_{\pi}\mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t r(s_t,a_t)\right]-\lambda\left(\mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t c(s_t,a_t)\right]-\alpha\right)$$

[1] C. Tessler, D. J. Mankowitz, and S. Mannor. "Reward Constrained Policy Optimization". In: *International Conference on Learning Representations*. 2019

Observation 1 Continued

Reward must provide two types of signals:

- Must ‘encourage’ good or desired behaviour.
- Must ‘discourage’ bad or undesired behaviours.

Observation 1 Continued

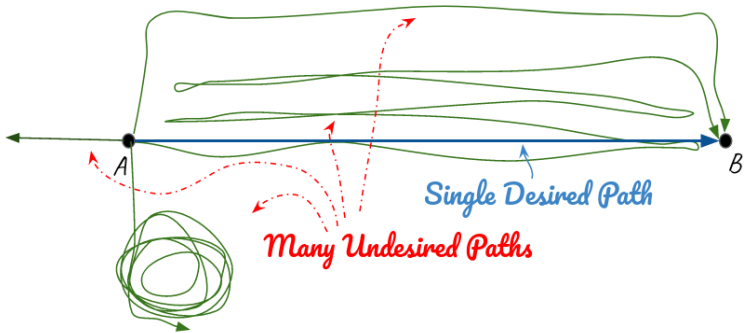
Reward must provide two types of signals:

- Must 'encourage' good or desired behaviour. ⟵ Few such behaviours.
- Must 'discourage' bad or undesired behaviours.

Observation 1 Continued

Reward must provide two types of signals:

- Must 'encourage' good or desired behaviour. ⟵ Few such behaviours.
- Must 'discourage' bad or undesired behaviours. ⟵ *Many* such behaviours.



Proposal: Inverse Constrained Reinforcement Learning

Reward must provide two types of signals:

- Must 'encourage' good or desired behaviour. $\longleftarrow$ Loosely specified by designer.
- Must 'discourage' bad or undesired behaviours. $\longleftarrow$ Learn from data.

Literature Review

Maximum Likelihood Constraint Inference[2]

- Formulates the problem of constraint inference in context of maximum entropy inverse reinforcement learning.
- Presents a greedy algorithm with bounded suboptimality.

Limitations

- Tabular settings.
- Model based appraoch.

[2] D. R. Scobee and S. S. Sastry. "Maximum Likelihood Constraint Inference for Inverse Reinforcement Learning". In: *International Conference on Learning Representations*. 2020

Imitation Learning

Learn to imitate expert agent by observing demonstrations of expert behaviour.

Imitation Learning

Learn to imitate expert agent by observing demonstrations of expert behaviour.

Generative Adversarial Imitation Learning[3]

- Casts the imitation learning problem in the context of adversarial learning framework.
- Provides guarantees on perfectly matching expert agent policy in infinite data regime.
- Most dominant IL framework currently.

[3] J. Ho and S. Ermon. "Generative Adversarial Imitation Learning". In: *Advances in Neural Information Processing Systems*. 2016

Imitation Learning

Learn to imitate expert agent by observing demonstrations of expert behaviour.

Generative Adversarial Imitation Learning

- Casts the imitation learning problem in the context of adversarial learning framework.
- Provides guarantees on perfectly matching expert agent policy in infinite data regime.
- Most dominant IL framework currently.

Guided Cost Learning [3]

- Inverse reinforcement learning method; infer a reward from demonstrations and then learn a policy under that reward function.
- Restricts (learned) reward function to be Lipschitz smooth and monotonically increasing.

[3] C. Finn, S. Levine, and P. Abbeel. "Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization". In: *International Conference on Machine Learning*. 2016

Reward Modelling

Inverse Reward Design [4]

- Posits that given reward function is only representative of what designer wants inside the training MDP.
- Uses Bayesian IRL to combine information from the given reward function and training MDP to learn a *true* reward fuction which may transfer better to other MDPs.

[4] D. Hadfield-Menell et al. "Inverse Reward Design". In: *Advances in Neural Information Processing Systems*. 2017

Reward Modelling

Inverse Reward Design

- Posits that given reward function is only representative of what designer wants inside the training MDP.
- Uses Bayesian IRL to combine information from the given reward function and training MDP to learn a *true* reward fuction which may transfer better to other MDPs.

Preferences Implicit In The State Of The World[4] Uses initiate state of the robot (final state of the demonstrator) as a prior and attempts to make the reward function consistent with this prior.

[4] P. F. Christiano et al. "Deep Reinforcement Learning from Human Preferences". In: *Advances in Neural Information Processing Systems*. 2017

Reward Modelling

Inverse Reward Design

- Posits that given reward function is only representative of what designer wants inside the training MDP.
- Uses Bayesian IRL to combine information from the given reward function and training MDP to learn a *true* reward fuction which may transfer better to other MDPs.

Preferences Implicit In The State Of The World Uses initiate state of the robot (final state of the demonstrator) as a prior and attempts to make the reward function consistent with this prior.

Learning Human Objectives by Evaluating Hypothetical Behavior [4] Simulates the extreme behaviours induced by reward function and generates queries by active learning to find *true* human preferences.

[4] S. Reddy et al. "Learning Human Objectives by Evaluating Hypothetical Behavior". In: *International Conference on Machine Learning*. 2020

Formulation & Algorithm

Problem Statement

Given a nominal MDP $\mathcal{M} =< S, A, \mathcal{T}, r >$ and demonstrations $\mathcal{D}$ from an optimal policy $\pi_{\mathcal{M}^{\mathcal{C}*}}$ operating in a constrained MDP $\mathcal{M}^{\mathcal{C}^*} = \mathcal{M} \cup \mathcal{C}^*$, recover a constraint set $\mathcal{C}$ such that optimal policy in $\mathcal{M}^{\mathcal{C}} = \mathcal{M} \cup \mathcal{C}$ is behaviourially equivalent to $\pi_{\mathcal{M}^{\mathcal{C}}}$.

Overview Of Work

- Formulates the constraint learning problem.
- Model free approach.
- Developed approach scales well to high dimensional continuous settings.
- Learned constraints are transferable across dynamics & agents.

Maximum Likelihood Constraint Inference

Asumptions

- All constraint sets are equally likely i.e. $p(\mathcal{C})$ is same for all $\mathcal{C}$.

Maximum Likelihood Constraint Inference

Asumptions

- All constraint sets are equally likely i.e. $p(\mathcal{C})$ is same for all $\mathcal{C}$.
- Trajectories in the dataset $\mathcal{D}$ are identically and independently distributed.

Maximum Likelihood Constraint Inference

Asumptions

- All constraint sets are equally likely i.e. $p(\mathcal{C})$ is same for all $\mathcal{C}$.
- Trajectories in the dataset $\mathcal{D}$ are identically and independently distributed.
- Each trajectory in the MDP $\mathcal{M}$ is distributed according to maximum entropy model[5].

$$\pi_{\mathcal{M}}(\tau) = \frac{\exp(\beta r(\tau))}{Z_{\mathcal{M}}} \mathbb{1}^{\mathcal{M}}(\tau). \tag{1}$$

where

- $Z_{\mathcal{M}} = \int \exp(\beta r(\tau)) \mathbb{1}^{\mathcal{M}}(\tau)$ is the partition function.
- $\beta \in [0, \text{inf})$ is the parameter controlloing the optimality of demonstration set (lower β means greater randomness in trajectories).
- $\mathbb{1}^{\mathcal{M}}$ is an indicator function which is 0 for all constrained trajectories.

[5] B. D. Ziebart et al. "Maximum Entropy Inverse Reinforcement Learning". In: *23rd National Conference on Artificial Intelligence (AAAI)*. 2008

Maximum Likelihood Constraint Inference

Asumptions

- All constraint sets are equally likely i.e. $p(\mathcal{C})$ is same for all $\mathcal{C}$.
- Trajectories in the dataset $\mathcal{D}$ are identically and independently distributed.
- Each trajectory in the MDP $\mathcal{M}$ is distributed according to maximum entropy model[5].

$$\pi_{\mathcal{M}}(\tau) = \frac{\exp(\beta r(\tau))}{Z_{\mathcal{M}}} \mathbb{1}^{\mathcal{M}}(\tau). \tag{1}$$

- Indicator function distributes over individual state action pairs i.e. $\mathbb{1}^{\mathcal{M}}(\tau) = \prod_{i=1}^{T} \mathbb{1}^{\mathcal{M}}(s_t, a_t)$.

[5] Ziebart et al., "Maximum Entropy Inverse Reinforcement Learning"

Maximum Likelihood Constraint Inference

Objective: Find the constraint set which best explains the demonstrations and nominal MDP M.

$$\mathcal{C}^* \leftarrow \underset{\mathcal{C}}{\operatorname{argmax}}\, p_{\mathcal{M}}(\mathcal{D}|\mathcal{C}). \tag{2}$$

Under the assumptions listed in the previous slide, we can write.

$$p(\mathcal{D}|\mathcal{C}) = \frac{1}{(Z_{\mathcal{M}^{\mathcal{C}}})^N} \prod_{i=1}^{N} \exp(\beta r(\tau^{(i)})) \mathbb{1}^{\mathcal{M}^{\mathcal{C}}}(\tau^{(i)}). \tag{3}$$

where N is the number of demonstartions in dataset $\mathcal{D}$.

Sample Based Approximation

$$\begin{aligned}\mathcal{L}(\mathcal{C}) &= \frac{1}{N}\log p(\mathcal{D}|\mathcal{C}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\beta r(\tau^{(i)}) + \log \mathbb{1}^{\mathcal{M}^{\mathcal{C}}}(\tau^{(i)})\right] - \log Z_{\mathcal{M}^{\mathcal{C}}} \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[\beta r(\tau^{(i)}) + \log \mathbb{1}^{\mathcal{M}^{\mathcal{C}}}(\tau^{(i)})\right] - \log \int \exp(\beta r(\tau))\mathbb{1}^{\mathcal{M}^{\mathcal{C}}}(\tau)d\tau.\end{aligned} \tag{4}$$

Key Idea

Use a classifier ζ_θ parametrized by θ to approximate the indicator function $\mathbb{1}^{\mathcal{M}^{\mathcal{C}}}(\tau)$.

$$\mathcal{L}(\mathcal{C}) \approx \mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\beta r(\tau^{(i)}) + \log \zeta_\theta(\tau^{(i)})\right] - \log \int \exp(\beta r(\tau))\zeta_\theta(\tau)d\tau. \tag{5}$$

Sample Based Approximation

By differentiating the eq (5) with repect to θ, we get[6]:

$$\nabla_\theta \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \zeta_\theta(\tau^{(i)}) - \mathbb{E}_{\tau \sim \pi_{\mathcal{M}^{\zeta_\theta}}} \left[\nabla_\theta \log \zeta_\theta(\tau) \right]. \tag{6}$$

Using a sample-based approximation for the right-hand term we can rewrite the gradient as

$$\nabla_\theta \mathcal{L}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \zeta_\theta(\tau^{(i)}) - \frac{1}{M} \sum_{j=1}^{M} \nabla_\theta \log \zeta_\theta(\hat{\tau}^{(j)}), \tag{7}$$

Intuition

Find a ζ_θ such that both expert demonstrations and nominal agent trajectories have an equal cost on average.

[6] Complete derivation in appendix on slide 36.

Regularizer

$$R(\theta) = \delta \sum_{\tau \sim \{\mathcal{D}, \mathcal{S}\}} [\zeta_\theta(\tau) - 1] \tag{8}$$

where

δ is a hyperparameter controlling the magnitude of regularizer loss.

$\mathcal{S}$ is set of trajecotries from $\pi_{\mathcal{M}^{\bar{\zeta}_\theta}}$.

Intuition

Motivates ζ_θ to *only* contrain states and actions when absolutely necessary and prevents trivial solution of outputting 0 everywhere (i.e. constraining everything).

Forward Step

To get samples from $\pi_{\bar{\zeta}_\theta}$, we parametrize it as a neural network with parameters ϕ and solve the forward constrained RL problem with using $\bar{\zeta}_\theta = 1 - \zeta_\theta$ as the cost estimate.

$$\min_{\lambda \geq 0} \max_{\phi} J(\pi^\phi) + \frac{1}{\beta}\mathcal{H}(\pi^\phi) - \lambda(\mathbb{E}_{\tau \sim \pi^\phi}[\bar{\zeta}_\theta(\tau)] - \alpha) \tag{9}$$

Forward Step

To get samples from $\pi_{\bar{\zeta}_\theta}$, we parametrize it as a neural network with parameters ϕ and solve the forward constrained RL problem with using $\bar{\zeta}_\theta = 1 - \zeta_\theta$ as the cost estimate.

$$\min_{\lambda \geq 0} \max_{\phi} J(\pi^\phi) + \frac{1}{\beta}\mathcal{H}(\pi^\phi) - \lambda(\mathbb{E}_{\tau \sim \pi^\phi}[\bar{\zeta}_\theta(\tau)] - \alpha) \tag{9}$$

Forward step is expensive!

Forward Step

To get samples from $\pi_{\bar{\zeta}_\theta}$, we parametrize it as a neural network with parameters ϕ and solve the forward constrained RL problem with using $\bar{\zeta}_\theta = 1 - \zeta_\theta$ as the cost estimate.

$$\min_{\lambda \geq 0} \max_{\phi} J(\pi^\phi) + \frac{1}{\beta}\mathcal{H}(\pi^\phi) - \lambda(\mathbb{E}_{\tau \sim \pi^\phi}[\bar{\zeta}_\theta(\tau)] - \alpha) \tag{9}$$

Forward step is expensive!

Key Idea 2

Do multiple ζ_θ updates for one forward step and use importance sampling to correct for the bias.

Importance Sampling

By leveraging the fact that constraints are assumed to be *Markovian*, we are able to derive IS weights that are only dependent on current timestep of trajectory[7].

$$\omega(s_t, a_t) = \frac{\zeta_\theta(s_t, a_t)}{\zeta_{\bar{\theta}}(s_t, a_t)}. \tag{10}$$

Final Objective

$$\nabla_\theta \mathcal{L}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{s_t, a_t \in \tau^{(i)}} \nabla_\theta \log \zeta_\theta(s_t, a_t) - \frac{1}{M} \sum_{j=1}^{M} \sum_{\hat{s}_t, \hat{a}_t \in \tau^{(j)}} \omega(\hat{s}_t, \hat{a}_t) \nabla_\theta \log \zeta_\theta(\hat{s}_t, \hat{a}_t), \tag{11}$$

where $\{\tau^{(j)}\}_{j=1}^{M}$ are sampled from $\pi_{\bar{\theta}}$.

[7] IS Weights derivation on slide 37.

KL Based Early Stopping

To control for the variance introduced by the importance sampling estimate, we use KL-based stopping criterions to prevent bad updates to ζ_θ [8].

$$
\begin{aligned}
D_{\mathrm{KL}}(\pi_{\tilde{\theta}} || \pi_\theta) &\leq 2 \log \bar{\omega} \\
D_{KL}(\pi_\theta || \pi_{\tilde{\theta}}) &\leq \frac{\mathbb{E}_{\tau \sim \pi_{\tilde{\theta}}} \left[(\omega(\tau) - \bar{\omega}) \log \omega(\tau) \right]}{\bar{\omega}}.
\end{aligned} \tag{12}
$$

[8] Forward KL bound derivation on slide 39. Reverse KL bound derivation on slide 40.

Algorithm

Algorithm 1 ICRL

Input: Expert trajectories $\mathcal{D}$, iterations N, backward iterations B, maximum allowable KLs ϵ_F and ϵ_R
Initialize θ and ϕ randomly
for $i = 1$ **to** N **do**
  Learn π^ϕ by solving (9) using current ζ_θ
  for $j = 1$ **to** B **do**
    Sample set of trajectories $\mathcal{S} = \{\tau^{(k)}\}_{k=1}^{M}$ from π^ϕ
    Compute importance sampling weights $w(\tau^{(k)})$ using (10) for $k = 1, \ldots, M$
    Use $\mathcal{S}$ and $\mathcal{D}$ to update θ via SGD by using the gradient in (11)
    Compute forward and reverse KLs using (12)
    if forward KL $\geq \epsilon_F$ **or** reverse KL $\geq \epsilon_R$: **then**
      break
    end if
  end for
end for

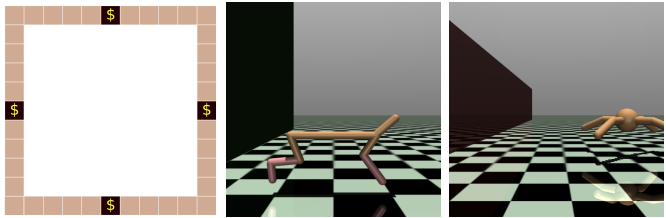
Experiments

Baselines

- **Binary Classifier (BC):** Train a binary classifier (using the cross-entropy loss) to classify between nominal and expert trajectories.
- **GAIL-Constraint (GC):**
  - Based on Generative Adversarial Imitation Learning (GAIL). [9].
  - Assume a reward structure of the form $\bar{r}(s,a) := r(s,a) + \log \zeta(s,a)$ and learns $\zeta(s,a)$ using GAIL's algorithm.

[9] Ho and Ermon, "Generative Adversarial Imitation Learning"

Results: Learning Constraints



(a) LapGridWorld (b) HalfCheetah (c) Ant

Figure: The environments used in the experiments for learning constraints.

Results: Learning Constraints

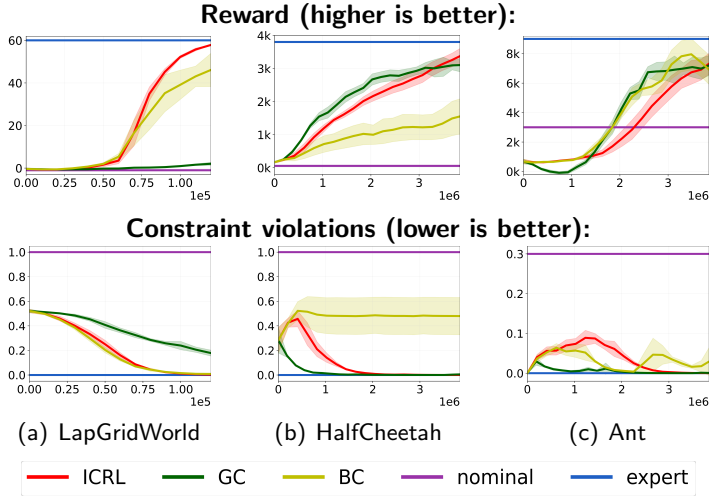
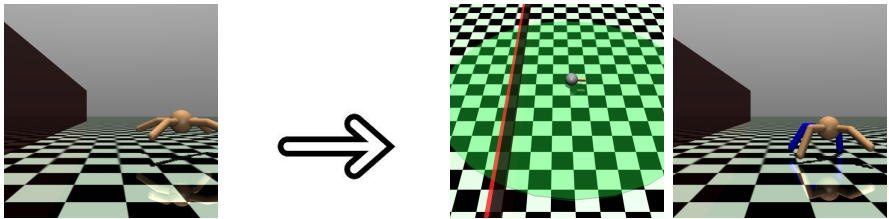


Figure: Performance of agents during training over several seeds (5 in LapGridWorld, 10 in others). The x-axis is the number of timesteps taken in the environment. The shaded regions correspond to the standard error.

Results: Transferring Constraints



(a) Ant (b) Point (c) AntBroken

Figure: Constraints learned in ant environment were transferred to point and ant broken environments.

Results: Transferring Constraints

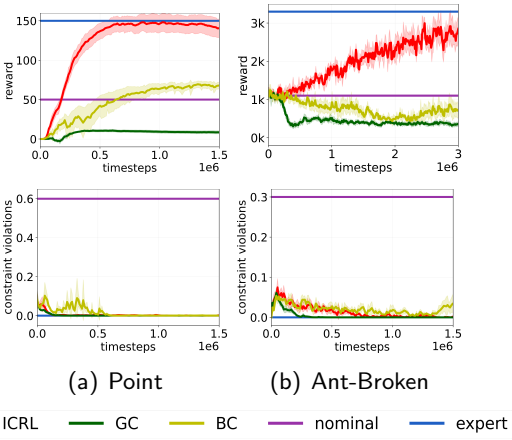


Figure: Transferring constraints. The x-axis is the number of timesteps taken in the environment. All plots were smoothed and averaged over 5 seeds. The shaded regions correspond to the standard error.

Ablation Studies

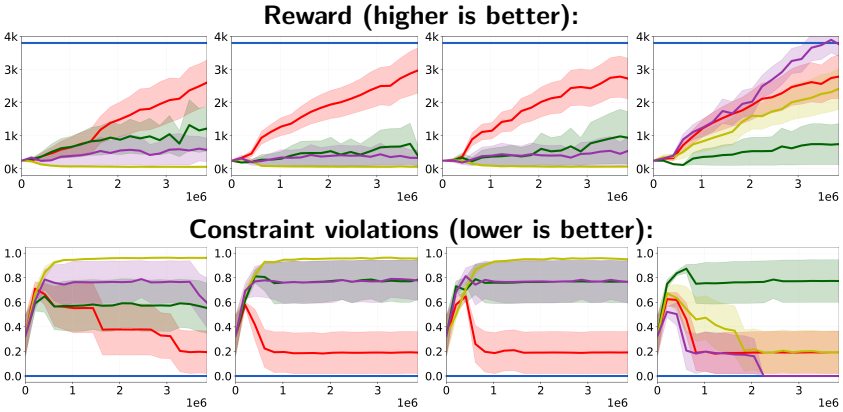


Figure: Ablation studies on the HalfCheetah environment. All plots were averaged over 5 seeds. IS refers to importance sampling and ES to early stopping. The x-axis corresponds to the number of timesteps the agent takes in the environment. Shaded regions correspond to the standard error.

Limitations & Future Work

- Maximum Causal Entropy & stochastic MDPs.
- Soft Constraints.
- Off-policy constraint learning.
- Robust imitation learning.

Acknowledgement

Thanks to Shehryar Malik, Dr. Ali Ahmed and Dr. Alireza Aghasi for their support and contribution in this work.

Thank You.

Appendix

Gradient Of Log Likelihood

The gradient of (5) is

$$
\begin{aligned}
\nabla_\theta \mathcal{L}(\theta) &= \frac{1}{N}\sum_{i=1}^{M}\left[0 + \nabla_\theta \log \zeta_\theta(\tau^{(i)})\right] - \frac{1}{\int \exp(\beta r(\tau))\zeta_\theta(\tau)d\tau}\int \exp(\beta r(\tau))\nabla_\theta \zeta_\theta(\tau)d\tau \\
&= \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \log \zeta_\theta(\tau^{(i)}) - \int \frac{\exp(\beta r(\tau))\zeta_\theta(\tau)}{\int \exp(\beta r(\tau'))\zeta_\theta(\tau')d\tau'}\nabla_\theta \log \zeta_\theta(\tau)d\tau \\
&= \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \log \zeta_\theta(\tau^{(i)}) - \int p_{\mathcal{M}^{\zeta_\theta}}(\tau)\nabla_\theta \log \zeta_\theta(\tau)d\tau \\
&= \frac{1}{N}\sum_{i=1}^{N}\nabla_\theta \log \zeta_\theta(\tau^{(i)}) - \mathbb{E}_{\tau\sim\pi_{\mathcal{M}^{\zeta_\theta}}}\left[\nabla_\theta \log \zeta_\theta(\tau)\right],
\end{aligned}
\tag{13}
$$

Derivation Of IS Weights - Sketch

Goal: Estimate $\nabla_\theta \log Z_\theta$ using samples from policy $\pi_{\bar{\zeta}_{\bar{\theta}}}$ corresponding to an older $\zeta_{\bar{\theta}}$.

$$Z_\theta = Z_{\bar{\theta}} \cdot \mathbb{E}_{\tau \sim \pi_{\zeta_{\bar{\theta}}}} \left[\frac{\zeta_\theta(\tau)}{\zeta_{\bar{\theta}}(\tau)} \right]. \tag{14}$$

Therefore

$$\begin{aligned} \nabla_\theta \log Z_\theta &= \frac{1}{Z_\theta} \nabla_\theta Z_\theta \\ &= \frac{1}{\mathbb{E}_{\tau \sim \pi_{\zeta_{\bar{\theta}}}} \left[\frac{\zeta_\theta(\tau)}{\zeta_{\bar{\theta}}(\tau)} \right]} \left[\mathbb{E}_{\tau \sim \pi_{\zeta_{\bar{\theta}}}} \left[\frac{\zeta_\theta(\tau)}{\zeta_{\bar{\theta}}(\tau)} \nabla_\theta \log \zeta_\theta(\tau) \right] \right]. \end{aligned} \tag{15}$$

Derivation Of IS Weights - Sketch

Note that $\mathbb{E}_{\tau\sim\pi_{\zeta_{\bar{\theta}}}}\left[\frac{\zeta_\theta(\tau)}{\zeta_{\bar{\theta}}(\tau)}\right] = \int \pi_{\zeta_\theta}(\tau)d\tau = 1.$

$$\begin{aligned}
\nabla_\theta \log Z_\theta &= \mathbb{E}_{\pi_{\zeta_{\bar{\theta}}}}\left[\frac{\zeta_\theta(\tau)}{\zeta_{\bar{\theta}}(\tau)}\nabla_\theta \log \zeta_\theta(\tau)\right] \\
&= \mathbb{E}_{\pi_{\zeta_{\bar{\theta}}}}\left[\prod_{t=1}^{T}\frac{\zeta_\theta(s_t,a_t)}{\zeta_{\bar{\theta}}(s_t,a_t)}\nabla_\theta \log \prod_{t'=1}^{T}\zeta_\theta(s_{t'},a_{t'})\right] \\
&= \sum_{t'=1}^{T}\mathbb{E}_{\tau/(s_{t'},a_{t'})\sim\pi_{\zeta_{\bar{\theta}}}}\left[\prod_{\substack{t=1\\t\neq t'}}^{T}\frac{\zeta_\theta(s_t,a_t)}{\zeta_{\bar{\theta}}(s_t,a_t)}\right]\mathbb{E}_{s_{t'},a_{t'}\sim\pi_{\zeta_{\bar{\theta}}}}\left[\frac{\zeta_\theta(s_{t'},a_{t'})}{\zeta_{\bar{\theta}}(s_{t'},a_{t'})}\nabla_\theta \log \zeta_\theta(s_{t'},a_{t'})\right] \\
&= \sum_{t'=1}^{T}\left(\frac{Z_\theta}{Z_{\bar{\theta}}}\right)\cdot\mathbb{E}_{\pi_{\zeta_{\bar{\theta}}}}\left[\frac{\zeta_\theta(s_{t'},a_{t'})}{\zeta_{\bar{\theta}}(s_{t'},a_{t'})}\nabla_\theta \log \zeta_\theta(s_{t'},a_{t'})\right] \\
&\approx \sum_{t'=1}^{T}\mathbb{E}_{\pi_{\zeta_{\bar{\theta}}}}\left[\frac{\zeta_\theta(s_{t'},a_{t'})}{\zeta_{\bar{\theta}}(s_{t'},a_{t'})}\nabla_\theta \log \zeta_\theta(s_{t'},a_{t'})\right].
\end{aligned}$$

Forward KL Expression

$$
\begin{aligned}
D_{KL}(\pi_{\bar{\theta}} || \pi_\theta) &= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\log \frac{\pi_{\bar{\theta}}(\tau)}{\pi_\theta(\tau)} \right] \\
&= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\log \frac{\zeta_{\bar{\theta}}(\tau)}{\zeta_\theta(\tau)} \right] + \log \frac{Z_\theta}{Z_{\bar{\theta}}}.
\end{aligned} \tag{17}
$$

Let $\omega(\tau)$ denote $\zeta_{\bar{\theta}}(\tau)/\zeta_\theta(\tau)$. Plugging in the expression for Z_θ from (14) and using Jensen's inequality gives

$$
\begin{aligned}
D_{KL}(\pi_{\bar{\theta}} || \pi_\theta) &= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} [\log \omega(\tau)] + \log \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} [\omega(\tau)] \\
&\leq 2 \log \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} [\omega(\tau)] .
\end{aligned} \tag{18}
$$

Reverse KL Expression

$$
\begin{aligned}
D_{KL}(\pi_\theta || \pi_{\bar{\theta}}) &= \mathbb{E}_{\tau \sim \pi_\theta} \left[\log \frac{\pi_\theta(\tau)}{\pi_{\bar{\theta}}(\tau)} \right] \\
&= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\frac{\pi_\theta(\tau)}{\pi_{\bar{\theta}}(\tau)} \log \frac{\pi_\theta(\tau)}{\pi_{\bar{\theta}}(\tau)} \right] \\
&= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\omega(\tau) \frac{Z_{\bar{\theta}}}{Z_\theta} \log \omega(\tau) \frac{Z_{\bar{\theta}}}{Z_\theta} \right] \\
&= \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\omega(\tau) \log \omega(\tau) \right] \frac{Z_{\bar{\theta}}}{Z_\theta} + \mathbb{E}_{\tau \sim \pi_{\bar{\theta}}} \left[\omega(\tau) \right] \frac{Z_{\bar{\theta}}}{Z_\theta} \log \frac{Z_{\bar{\theta}}}{Z_\theta}.
\end{aligned}
\tag{19}
$$

Reverse KL Expression

From (14) we know that $Z_{\bar{\theta}}/Z_{\theta} = 1/\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}\omega(\tau)$. Using Jensen's inequality we have

$$\begin{aligned} D_{KL}(\pi_\theta||\pi_{\bar{\theta}}) &= \frac{1}{\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)]}\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)\log\omega(\tau)] - \log\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)] \\ &\leq \frac{1}{\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)]}\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)\log\omega(\tau)] - \mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\log\omega(\tau)]\,. \end{aligned} \tag{20}$$

Letting $\bar{\omega}$ denote $\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[\omega(\tau)]$ gives us

$$D_{KL}(\pi_\theta||\pi_{\bar{\theta}}) \leq \frac{\mathbb{E}_{\tau\sim\pi_{\bar{\theta}}}[(\omega(\tau)-\bar{\omega})\log\omega(\tau)]}{\bar{\omega}}. \tag{21}$$