Learning to behave via Imitation
ESSAI 2024 Course
Lecture 2/5

George Vouros

University of Piraeus, Greece

July 23, 2024

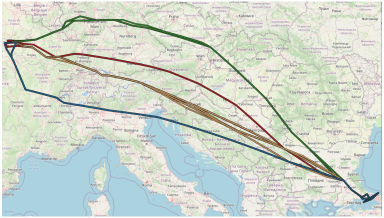
Outline

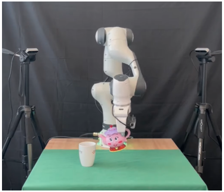
- Day 1: Motivation & Introduction to Deep Reinforcement Learning
- **Day 2: Inverse Reinforcement Learning and Connections to Probabilistic Inference**
- Day 3: Imitation Learning
- Day 4: Non-Markovian, Multimodal Imitation Learning
- Day 5: Imitating in Constrained Settings, Multiagent Imitation Learning.

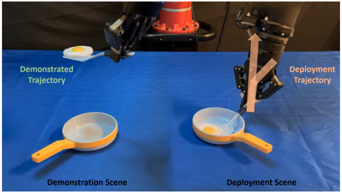
Inverse Reinforcement Learning

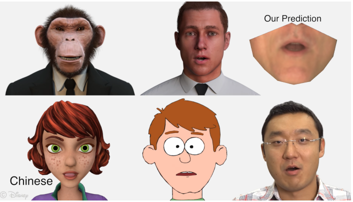
- Is it possible to always define manually the reward function, assumed known by RL algorithms?
- Learn a reward approximation from expert demonstrations
- Draw connections to probabilistic models of behavior and inference
- Review some of the practical algorithms

Why should we care about inverse reinforcement learning?









Inverse Reinforcement Learning

Question to be answered

Given that we deal with rational agents, aiming to maximize their expected utility, how to uncover this rationality given few demonstrations of executing their tasks?

Recall the Reinforcement Learning objective

Probability of τ given a policy π_θ

$p_\theta(\tau) = p(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)$

Objective: tune θ to get

$\theta^* = argmax_\theta \mathbb{E}_{\tau \sim p_\theta}[\sum_t r_t], r_t = r(s_t, a_t)$

Objective for finite time horizons

$\theta^* = argmax_\theta \sum_{t=1}^{T}\mathbb{E}_{(s_t,a_t)\sim p_\theta(s_t,a_t)}[r_t]$, where $p_\theta(s_t, a_t)$ the state, action marginal

Objective for infinite time horizons

$\theta^* = argmax_\theta \mathbb{E}_{(s,a)\sim\mu}[r(s,a)]$, where $\mu = p_\theta(s,a)$ the stationary distribution of states, actions

Inverse Reinforcement Learning

Question to be answered

Given that we deal with rational agents, aiming to maximize their expected utility, how to uncover this rationality given demonstrations of executing their tasks?

RL objective: find the most rewarding course of actions

- Deterministic case:
  $\tau = (a_1, s_1), (a_2, s_2), \ldots, (a_T, s_T) = argmax_{a_1, a_2, \ldots a_T} \sum_{t=1}^{T} r(s_t, a_t), s_{t+1} = f(s_t, a_t)$
- Stochastic case:
  $\theta^* = argmax_\theta \mathbb{E}_{\tau \sim p_\theta}[\sum_t r(s_t, a_t)]$, $a_t \sim \pi_\theta(a_t|s_t)$ and $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$

Inverse Reinforcement Learning

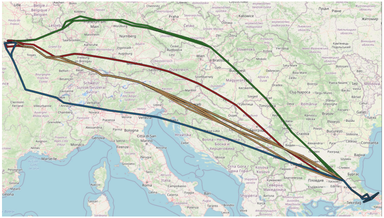
Question to be answered

Given that we deal with rational agents, aiming to maximize their expected utility, how to uncover this rationality given demonstrations of executing their tasks?

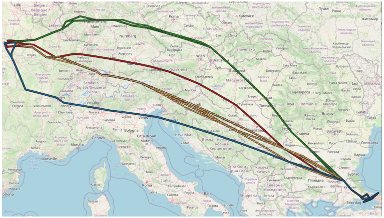
IRL Objective:

Find a reward function that explains the demonstrations.

Why should we care about inverse reinforcement learning?



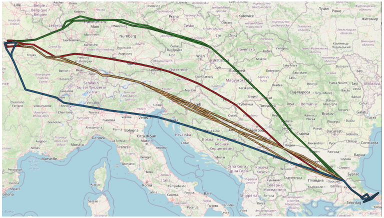
Why should we care about inverse reinforcement learning?



Imitation learning perspective:

Imitation of actions   Imitation of humans

Why should we care about inverse reinforcement learning?



Imitation learning perspective:

Imitation of actions to learn how to perform a task

Imitation of humans to learn humans' objectives or intentions

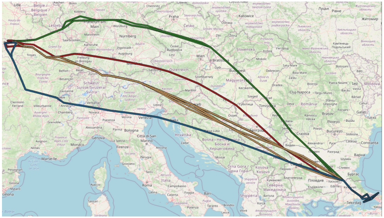
Why should we care about inverse reinforcement learning?

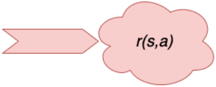
What can a reward function be?

Why should we care about inverse reinforcement learning?

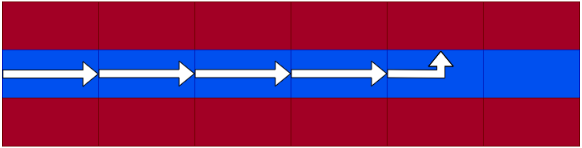
The goal of inverse reinforcement learning is to infer the reward function from demonstrations

Why should we care about inverse reinforcement learning?

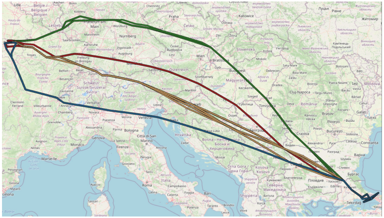


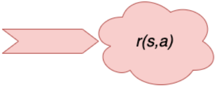


There can be many reward functions explaining the demonstrated behaviour

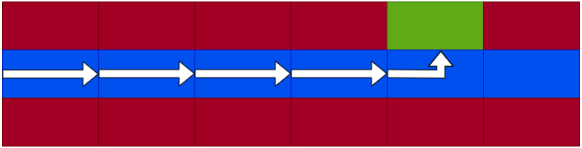


Why should we care about inverse reinforcement learning?

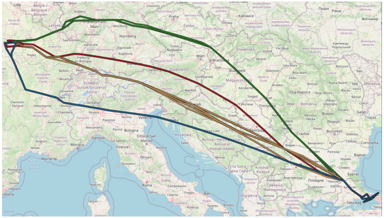


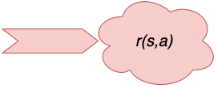


There can be many reward functions explaining the demonstrated behaviour

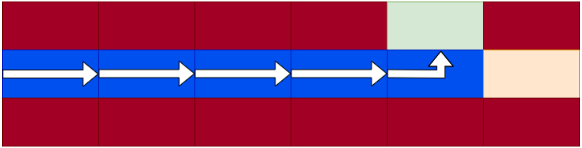


Why should we care about inverse reinforcement learning?





There can be many reward functions explaining the demonstrated behaviour



Inverse reinforcement learning

Formal constituents of the problem

- states $s \in \mathcal{S}$
- actions $a \in \mathcal{A}$
- (maybe) transition probabilities $p(s'|s, a)$
- samples τ_i sampled from (unknown) π^*
- **Goal:** learn $r_\psi(s, a)$, where ψ are function parameters

Inverse reinforcement learning

Formal constituents of the problem

- states $s \in \mathcal{S}$
- actions $a \in \mathcal{A}$
- (maybe) transition probabilities $p(s'|s, a)$
- samples τ_i sampled from π^*
- **Goal:** learn $r_\psi(s, a)$, where ψ are function parameters

and then use $r_\psi(s, a)$ to learn $\pi^{r_\psi} = \hat{\pi}^*$

Inverse reinforcement learning

Reward parameters

- Assuming a linear function, these are features' coefficients:

$$r_\psi(s,a) = \sum_i \psi_i f_i(s,a) = \psi^T \mathbf{f(s,a)}$$

  Note: Features can be of arbitrary complexity

- Or we may have a function approximator (a neural net) with parameters ψ, input (s,a) and output $r_\psi(s,a)$

Inverse reinforcement learning

Assuming a linear reward function

$$r_\psi(s,a) = \sum_i \psi_i f_i(s,a) = \psi^T \mathbf{f(s,a)}$$

we need to match features on expectation:

$$\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f(s,a)}] = \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}]$$

Again: there can be many reward functions satisfying this property.

Inverse reinforcement learning

Assuming a linear reward function

$$r_\psi(s,a) = \sum_i \psi_i f_i(s,a) = \psi^T \mathbf{f(s,a)}$$

we need to match features on expectation:

$$\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f(s,a)}] = \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}]$$

use the max margin principle:

$$max_{\psi,m} m, \text{ such that } \psi^T \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}] \geq max_{\pi \in \Pi} \psi^T \mathbb{E}_\pi[\mathbf{f(s,a)}] + m$$

Inverse reinforcement learning

Assuming a linear reward function

$$r_\psi(s,a) = \sum_i \psi_i f_i(s,a) = \psi^T \mathbf{f(s,a)}$$

we need to match features on expectation:

$$\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f(s,a)}] = \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}]$$

use the max margin principle:

$$max_{\psi,m} m, \textit{ such that } \psi^T \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}] \geq max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[\mathbf{f(s,a)}] + m$$

or by "weighting" policies using a kind of similarity to π^*:

$$min_\psi \frac{1}{2} ||\psi||^2, \textit{ such that } \psi^T \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}] \geq max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[\mathbf{f(s,a)}] + D(\pi^*, \pi)$$

Inverse reinforcement learning

Assuming a linear reward function

we need to match features on expectation:

$$\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f(s,a)}] = \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}]$$

use the max margin principle:

$$max_{\psi,m} m, \text{ such that } \psi^T \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}] \geq max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[\mathbf{f(s,a)}] + m$$

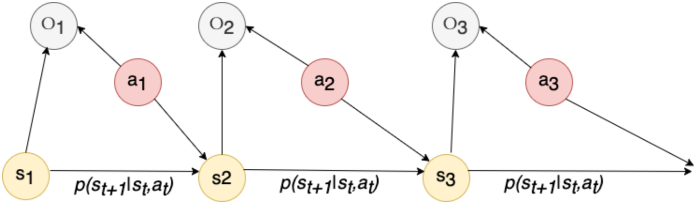
or (re-statement):

$$min_{\psi} \frac{1}{2} ||\psi||^2, \text{ such that } \psi^T \mathbb{E}_{\pi^*}[\mathbf{f(s,a)}] \geq max_{\pi \in \Pi} \psi^T \mathbb{E}_{\pi}[\mathbf{f(s,a)}] + D(\pi^*, \pi)$$

However, an arbitrary heuristic, and does not account for sub-optimality of the expert.

Inverse reinforcement learning: connection to probabilistic models

Probabilistic graphical models

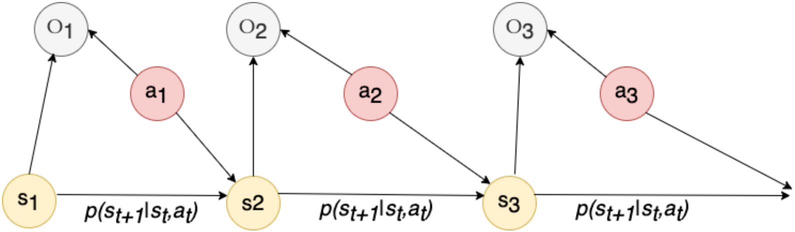


$$p(\mathcal{O}_t|s_t, a_t) = exp(r(s_t, a_t)),$$

$$\begin{aligned} p(\tau|\mathcal{O}_{1:T}) &= \frac{p(\tau, \mathcal{O}_{1:T})}{p(\mathcal{O}_{1:T})} \\ &\propto p(\tau) \prod_t exp(r(s_t, a_t)) \\ &= p(\tau) exp(\sum_t r(s_t, a_t)) \end{aligned}$$

Inverse reinforcement learning: connection to probabilistic models

Probabilistic graphical models



The Inverse problem:
Given the demonstrated trajectories, can we infer a reward function so that the demonstrated trajectories are most likely?

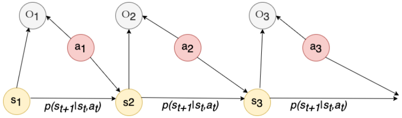
Inverse reinforcement learning: connection to probabilistic models

Question to be answered:

Given, a set of demonstrations τ_i,
what is the reward function $r_\psi(s, a)$ that maximizes the likelihood of demonstrated trajectories to be inferred from the probabilistic graphical model, given $\pi^{r_\psi}(\tau)$?

Inverse reinforcement learning: connection to probabilistic models

Probabilistic graphical models

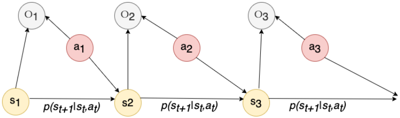


$$p(\mathcal{O}_t|s_t, a_t, \psi) = exp(r_\psi(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{1:T}, \psi) \propto p(\tau)exp(\sum_t r_\psi(s_t, a_t))$$

Inverse reinforcement learning: connection to probabilistic models

Probabilistic graphical models



$$p(\mathcal{O}_t|s_t, a_t, \psi) = exp(r_\psi(s_t, a_t))$$

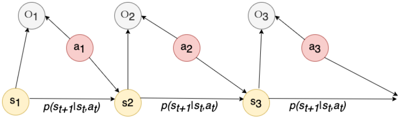
$$p(\tau|\mathcal{O}_{1:T}, \psi) \propto p(\tau)exp(\sum_t r_\psi(s_t, a_t))$$

maximum likelihood learning:

$$\mathcal{L} = max_\psi \frac{1}{N}\sum_{i=1}^{N} logp(\tau_i|\mathcal{O}_{1:T}, \psi)$$

Inverse reinforcement learning: connection to probabilistic models

Probabilistic graphical models



$$p(\mathcal{O}_t|s_t, a_t, \psi) = exp(r_\psi(s_t, a_t))$$

$$p(\tau|\mathcal{O}_{1:T}, \psi) \propto p(\tau)exp(\sum_t r_\psi(s_t, a_t))$$

maximum likelihood learning:

$$\mathcal{L} = max_\psi \frac{1}{N} \sum_{i=1}^{N} logp(\tau_i|\mathcal{O}_{1:T}, \psi) = max_\psi \frac{1}{N} \sum_{i=1}^{N} r_\psi(\tau_i) - logZ$$

Inverse reinforcement learning: connection to probabilistic models

$$p(\tau|\mathcal{O}_{1:T}) \propto p(\tau) exp(\sum_t r_\psi(s_t, a_t))$$

maximum likelihood learning:

$$\mathcal{L} = max_\psi \frac{1}{N} \sum_{i=1}^{N} logp(\tau_i|\mathcal{O}_{1:T}, \psi) = max_\psi \frac{1}{N} \sum_{i=1}^{N} r_\psi(\tau_i) - logZ$$

, where

$$Z = \int p(\tau) exp(r_\psi(\tau))\, d\tau$$

and it turns out that

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau \sim \pi^*(\tau)}[\nabla_\psi r_\psi(\tau)] - \mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T}, \psi)}[\nabla_\psi r_\psi(\tau)]$$

Inverse reinforcement learning: connection to probabilistic models

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau \sim \pi^*(\tau)}[\nabla_\psi r_\psi(\tau)] - \mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T},\psi)}[\nabla_\psi r_\psi(\tau)]$$

for the 2^{nd} term

$$\mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T},\psi)}[\nabla_\psi r_\psi(\tau)] =$$

$$\mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T},\psi)}[\nabla_\psi \sum_{t=1}^{T} r_\psi(s_t, a_t)] =$$

$$\mathbb{E}_{(s_t,a_t) \sim p(s_t,a_t|\mathcal{O}_{1:T},\psi)}[\nabla_\psi \sum_{t=1}^{T} r_\psi(s_t, a_t)]$$

Inverse reinforcement learning: connection to probabilistic models

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau\sim\pi^*(\tau)}[\nabla_\psi r_\psi(\tau)] - \mathbb{E}_{(s_t,a_t)\sim p(s_t,a_t|\mathcal{O}_{1:T},\psi)}[\nabla_\psi \sum_{t=1}^{T} r_\psi(s_t,a_t)]$$

and

$$\begin{aligned}
&p(s_t,a_t|\mathcal{O}_{1:T},\psi) = \\
&p(a_t|s_t,\mathcal{O}_{1:T},\psi)p(s_t|\mathcal{O}_{1:T},\psi) \propto \\
&\frac{\beta(s_t,a_t)}{\beta(s_t)}\alpha(s_t)\beta(s_t) = \beta(s_t,a_t)\alpha(s_t) \\
&\text{let } \mu(s_t,a_t) \propto \beta(s_t,a_t)\alpha(s_t)
\end{aligned}$$

i.e., a product of backward and forward messages.

Inverse reinforcement learning: connection to probabilistic models

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau\sim\pi^*(\tau)}\left[\nabla_\psi r_\psi(\tau)\right] - \mathbb{E}_{(s_t,a_t)\sim p(s_t,a_t|\mathcal{O}_{1:T},\psi)}\left[\nabla_\psi \sum_{t=1}^{T} r_\psi(s_t,a_t)\right] =$$

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\psi r_\psi(s_{i,t},a_{i,t}) - \sum_{t=1}^{T}\int\int \mu_t(s_t,a_t)\nabla_\psi r_\psi(s_t,a_t)\,ds_t\,da_t$$

Inverse reinforcement learning: connection to probabilistic models

MaxEnt IRL algorithm[1]

Iterate over the following steps:

1. Given ψ, compute backward message $\beta(s_t, a_t)$
2. Given ψ, compute forward message $\alpha(s_t)$
3. Compute $\mu(s_t, a_t) \propto \beta(s_t, a_t)\alpha(s_t)$
4. Evaluate $\nabla_\psi \mathcal{L} =$
$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\psi r_\psi(s_{i,t}, a_{i,t}) - \sum_{t=1}^{T}\int\int \mu_t(s_t, a_t)\nabla_\psi r_\psi(s_t, a_t)\, ds_t\, da_t$
5. $\psi \leftarrow \psi + \eta\nabla_\psi \mathcal{L}$

[1] Ziebart et al., (2018) "Maximum Entropy Inverse Reinforcement Learning"

Inverse reinforcement learning: connection to probabilistic models

MaxEnt IRL algorithm

Iterate over the following steps:

1. Given ψ, compute backward message $\beta(s_t, a_t)$
2. Given ψ, compute forward message $\alpha(s_t)$
3. Compute $\mu(s_t, a_t) \propto \beta(s_t, a_t)\alpha(s_t)$
4. Evaluate $\nabla_\psi \mathcal{L} =$
$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\psi r_\psi(s_{i,t}, a_{i,t}) - \sum_{t=1}^{T}\int\int \mu_t(s_t, a_t)\nabla_\psi r_\psi(s_t, a_t)\, ds_t\, da_t$
5. $\psi \leftarrow \psi + \eta\nabla_\psi \mathcal{L}$

So what?

1. In the case of linearity $r_\psi(s, a) = \psi^T \mathbf{f(s,a)}$ it is shown that it optimizes $max_\psi \mathcal{H}(\pi^{r_\psi})$ such that $\mathbb{E}_{\pi^{r_\psi}}[\mathbf{f}] = \mathbb{E}_{\pi^*}[\mathbf{f}]]$, **but**
2. In low dimensional spaces of actions and states with known dynamics

Inverse reinforcement learning: connection to probabilistic models

What we need to apply IRL to practical problem settings:

1. Handle large and continuous state and action spaces
2. Obtain states only via sampling
3. Solve in cases of unknown dynamics

Inverse reinforcement learning

Recall that

$$\nabla_\psi \mathcal{L} = \mathbb{E}_{\tau \sim \pi^*(\tau)}[\nabla_\psi r_\psi(\tau)] - \mathbb{E}_{\tau \sim p(\tau|\mathcal{O}_{1:T},\psi)}[\nabla_\psi r_\psi(\tau)]$$

Obtain the dynamics by sampling (as in standard RL)

Learn $p(a_t|s_t, \mathcal{O}_{1:T}, \psi)$ using a MaxEnt RL algorithm maximizing the following objective and run the policy to sample.

$$J(\theta) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\pi(s_t,a_t)}[r(s_t,a_t)] + \mathbb{E}_{s_t\sim\pi(s_t)}\mathcal{H}(\pi(a_t|s_t))$$

Then,

$$\nabla_\psi \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi r_\psi(\tau_i) - \frac{1}{M}\sum_{i=1}^{M} \nabla_\psi r_\psi(\tau_j)$$

Inverse reinforcement learning

Obtain the dynamics by sampling (as in standard RL)

Learn $p(a_t|s_t, \mathcal{O}_{1:T}, \psi)$ using a MaxEnt RL algorithm and run the policy to sample, at any gradient step for the reward function.

$$\nabla_\psi \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi r_\psi(\tau_i) - \frac{1}{M}\sum_{j=1}^{M} \nabla_\psi r_\psi(\tau_j)$$

Too expensive!
Try to build on previous policy approximations, using samples generated by them using importance sampling to fight bias.

$$\nabla_\psi \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\psi r_\psi(\tau_i) - \frac{1}{\sum_j w_j}\sum_{j=1}^{M} w_j \nabla_\psi r_\psi(\tau_j)$$

Inverse reinforcement learning

Obtain the dynamics by sampling (as in standard RL)

Build on previous policy approximations, using samples generated by them using importance sampling to fight bias.

$$\nabla_\psi \mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \nabla_\psi r_\psi(\tau_i) - \frac{1}{\sum_j w_j} \sum_{j=1}^{M} w_j \nabla_\psi r_\psi(\tau_j)$$

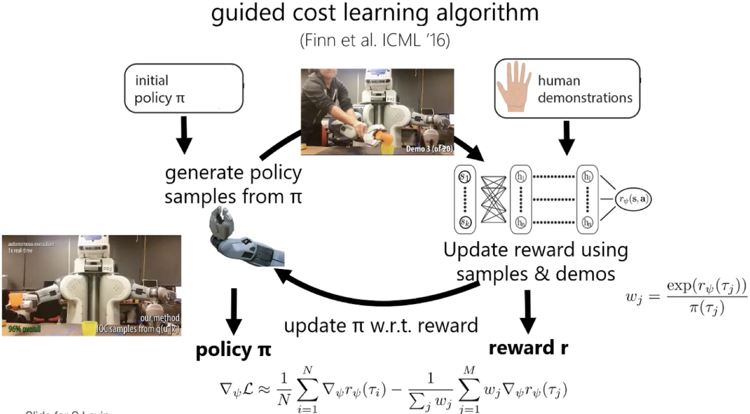
with (as in importance sampling)

$$w_j = \frac{p(\tau_j) exp(r_\psi(\tau_j))}{\pi(\tau_j)} =$$

$$\frac{p(s_1) \prod_t p(s_{t+1}|s_t, a_t) exp(r_\psi(s_t, a_t))}{p(s_1) \prod_t p(s_{t+1}|s_t, a_t) \pi(a_t|s_t)} = \frac{exp(\sum_t r_\psi(s_t, a_t))}{\prod_t \pi(a_t|s_t)}$$

Each policy update given an r_ψ, brings these importance weights closer to 1

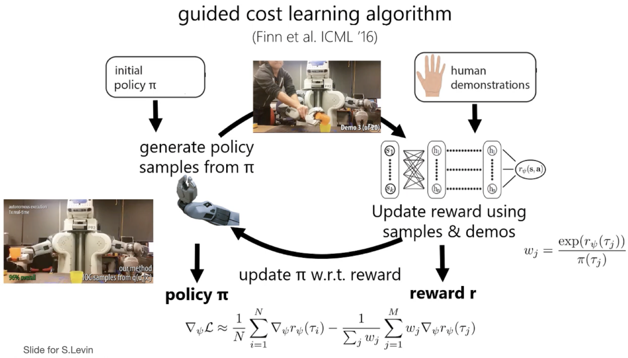
Inverse reinforcement learning



Slide for S.Levin

Inverse reinforcement learning: connection to probabilistic models

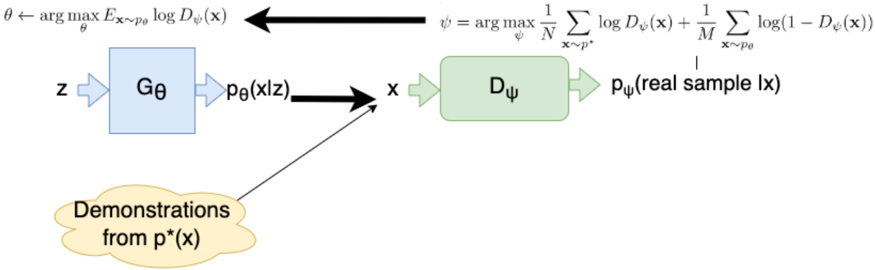
Isn't like an adversarial game?



The policy improves to get high reward, while the reward is updated to distinguish expert samples from those of the policy.

Inverse reinforcement learning

Can you see the analogy to GANs?



The generator improves itself to foul the discriminator, while the discriminator is updated to distinguish expert samples from those of the generator, as better as possible.

Inverse reinforcement learning

What can the discriminator be?

At convergence, for GANs the optimal discriminator represents the density ratio between p^* and p_θ:

$$D^*(x) = \frac{p^*(X)}{p_\theta(x) + p^*(x)}$$

For IRL, given that $\pi_\theta(\tau) \propto p(\tau) exp(r(\tau))$

$$D_\psi(\tau) = \frac{p(\tau)(1/Z)exp(r_\psi(\tau))}{p_\theta(\tau) + p(\tau)(1/Z)exp(r_\psi(\tau))} = \frac{(1/Z)exp(r_\psi(\tau))}{\prod_t \pi_\theta(a_t|s_t) + (1/Z)exp(r_\psi(\tau))}$$

Inverse reinforcement learning

What can the discriminator be?

For IRL, given that $\pi_\theta(\tau) \propto p(\tau)exp(r(\tau))$

$$D_\psi(\tau) = \frac{(1/Z)exp(r_\psi(\tau))}{\prod_t \pi_\theta(a_t|s_t) + (1/Z)exp(r_\psi(\tau))}$$

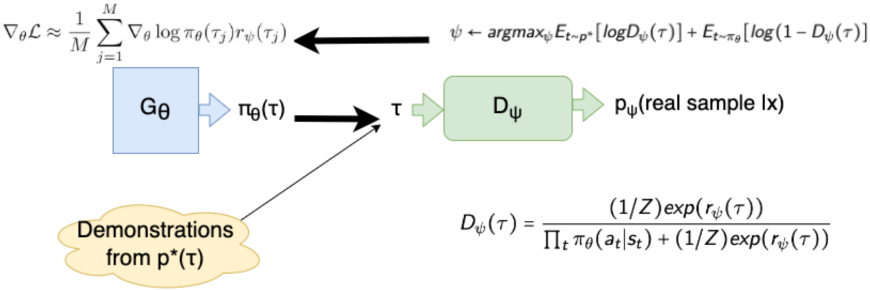
We optimize the reward (and Z) with the objective[2]

$$\psi \leftarrow argmax_\psi \mathbb{E}_{\tau \sim p^*}[logD_\psi(\tau)] + \mathbb{E}_{\tau \sim \pi_\theta}[log(1 - D_\psi(\tau)]$$

[2] Finn et al., (2016) "A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models"

Inverse reinforcement learning

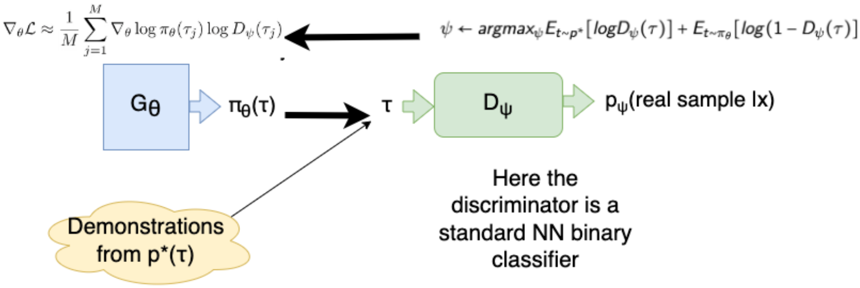
IRL as a GAN



The generator improves itself to foul the discriminator, while the discriminator is updated to distinguish expert samples from those of the generator, as better as possible.

Inverse reinforcement learning

We will return to this style of learning while talking on imitation learning



The generator improves itself to foul the discriminator, while the discriminator is updated to distinguish expert samples from those of the generator, as better as possible.