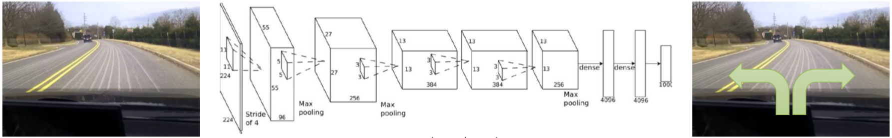
Generative Adversarial Imitation Learning

Jonathan Ho, Stefano Ermon - NIPS 2016

Stanford University

Presenter: Jinsung Yoon

- Learning to perform a task from **demonstrations (Expert trajectories) without interaction** with the expert or access to reinforcement signal
- Applications: When the **rewards are hard to define**



$\mathbf{o}_t$ $\pi_\theta(\mathbf{u}_t|\mathbf{o}_t)$ $\mathbf{u}_t$

Examples:

- **Autonomous vehicles:** `https://www.youtube.com/watch?v=cFtnflNe5fM`
- **Medicine**

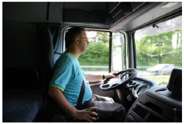
Problem: Imitation Learning

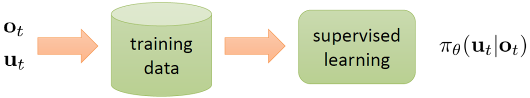
Two main approaches

- **Behavioral cloning (BC)**
  - A supervised learning problem that maps state/action pairs to policy. **(State: feature, action: label)**
  - Requires a large number of expert trajectories **(high sample complexity)** - due to **compounding error** caused by covariate shift
  - Copies unnecessary actions as well
- **Inverse Reinforcement Learning (IRL)**
  - Learns the **reward function** from expert trajectories that **prioritizes** entire trajectories over others, then derives the optimal policy
  - Expensive to run **(Inner loop has RL)**
  - Indirectly learns optimal policy from the reward function **(Using RL)**

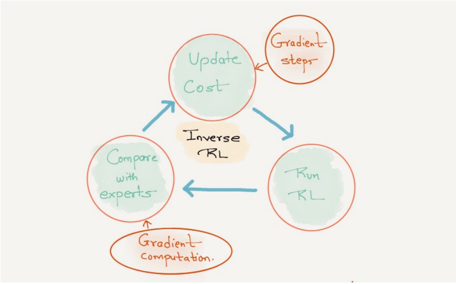
Problem: Imitation Learning

- **Behavioral cloning (BC)**





- **Inverse Reinforcement Learning (IRL)**



Proposed Model

Generative Adversarial Imitation Learning (GAIL)

- **Directly extracting a policy** from data as if it were obtained by RL following IRL.
- **Bypassing** any intermediate IRL step
- Draws an analogy between **imitation learning** and **generative adversarial networks (GAN)**
- Derive a model-free imitation learning algorithm with **significant performance improvement** with low sample and computational complexity.

Background: Preliminaries on RL

- $\mathcal{S}$: Finite state space
- $\mathcal{A}$: Finite action space
- Π : the set of all stochastic policies. Take action $\in \mathcal{A}$ given state $\in \mathcal{S}$
- $P(s'|s,a)$: Model dynamics
- $\pi \in \Pi$: A policy
- γ-discounted infinite horizon setting

$$\mathbb{E}_\pi[c(s,a)] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)]$$

- where $s_0 \sim p_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)$
- π_E: Expert policy

Inverse Reinforcement Learning (IRL)

- **Maximum causal entropy IRL**

$$\max_{c \in \mathcal{C}} \left[\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)] \right] - \mathbb{E}_{\pi_E}[c(s,a)]$$

- where $H(\pi) = [E]_{\pi}[-\log \pi(a|s)]$
- Try to find a cost function $c \in \mathcal{C}$ that assigns **low cost** to the expert policy π_E and **high cost** to other poilcies (π)
- Using **RL procedure**, we can find the expert policy based on the cost c

$$RL(c) = \arg\min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)]$$

- Inner loop has **RL**; thus, slow

Proposed Framework

- Use the **largest possible set of cost functions** $\mathcal{C} = \mathbb{R}^{\mathcal{S}\times\mathcal{A}} = \{c : \mathcal{S}\times\mathcal{A} \to \mathbb{R}\}$
- Use Gaussian processes or **Neural networks** to find the best cost function c among large cost function class $\mathcal{C}$
- Avoid overfitting, we use a **"convex" cost function regularizer** $\psi : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R} \cup \infty$
- With ψ, IRL procedure can be written as

$$\mathrm{IRL}_\psi(\pi_E) = \arg\max_{c \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}} -\psi(c) + \left[\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_\pi[c(s,a)]\right] - \mathbb{E}_{\pi_E}[c(s,a)]$$

- Let $\tilde{c} \in \mathrm{IRL}_\psi(\pi_E)$, we are interested in $\pi = \mathrm{RL}(\tilde{c})$

Occupancy Measure

- For a policy $\pi \in \Pi$, define its occupancy measure $\rho_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as

$$\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi)$$

- In words, occupancy measure is the **distribution** of state-action pairs with policy π

$$\mathbb{E}_\pi[c(s, a)] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)] = \sum_{s,a} \rho_\pi(s, a) c(s, a)$$

- The set of valid occupancy measures $\mathcal{D} = \{\rho_\pi : \pi \in \Pi\}$ can be written as

$$\mathcal{D} = \{\rho : \rho \geq 0 \& \sum_{a} \rho(s, a) = p_0(s) + \gamma \sum_{s', a} P(s|s', a)\rho(s', a) \forall s \in \mathcal{S}\}$$

- Note that there is **1-1 correspondence** between Π and $\mathcal{D}$

Occupancy Measure

- π_ρ to denote the **unique policy for an occupancy measure** ρ
- **Convex conjugate:** for a function $f : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}\cup\infty$, its convex conjugate $f^* : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}\cup\infty$ is

$$f^*(x) = \sup_{y\in\mathbb{R}^{\mathcal{S}\times\mathcal{A}}} x^T y - f(y)$$

- Then, $\mathrm{RL}(\tilde{c})$ can be written as

$$\mathrm{RL}\odot\mathrm{IRL}_\psi(\pi_E) = \arg\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

- Proof:

$$\begin{aligned}&\arg\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})\\ =&\arg\min_{\pi\in\Pi}\max_{c} -H(\pi) - \psi(c) + \sum_{s,a}\rho(s,a)c(s,a) - \sum_{s,a}\rho_{\pi_E}(s,a)c(s,a)\end{aligned}$$

Occupancy Measure

- The above theorem said that ψ-regularized IRL, implicitly seeks a **policy whose occupancy measure is close to the expert's** as measured by the convex function ψ^*
- It shows that various settings of ψ lead to various imitation learning algorithms that **directly solve the optimization problem.** (without IRL and RL iteration)
- Therefore we can deduce the following things
  - IRL is a dual of an **occupancy measure matching problem**
  - The induced optimal policy is the **primal optimum.**
- Therefore, the traditional IRL definition (**finding a cost function that the expert poilicy is uniquely optimal**) changes to (**finding a policy that matches the expert's occupancy measure**)

- Proposed ψ

$$\psi_{GA}(c) = \begin{cases} \mathbb{E}_{\pi_E}[g(c(s,a))], & \text{if } c < 0. \\ \infty, & \text{otherwise.} \end{cases}$$

where

$$g(x) = \begin{cases} -x - \log(1 - e^x), & \text{if } x < 0. \\ \infty, & \text{otherwise.} \end{cases}$$

- **Low penalty** when x is far from 0. **High penalty** when x is close to 0.

ψ_{GA}

- The regularization function ψ_{GA} is **motivated by the following fact.**

$$\psi^*_{GA}(\rho_\pi - \rho_{\pi_E}) = \max_{D \in (0,1)^{\mathcal{S} \times \mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]$$

- where the **maximum ranges over discriminative classifiers** is $D : \mathcal{S} \times \mathcal{A} \to (0,1)$
- The above equation is the **optimal negative log loss** of the binary classification problem of distinguishing between state-action pairs of π and π_E.
- Therefore, the optimal loss is the **Jensen-Shannon divergence**

$$D_{JS}(\rho_\pi, \rho_{\pi_E}) = D_{KL}(\rho_\pi || (\rho_\pi + \rho_E)/2) + D_{KL}(\rho_{\pi_E} || (\rho_\pi + \rho_{\pi_E})/2)$$

Proposed Optimization Problem

$$\min_{\pi} \psi^*_{GA}(\rho_\pi - \rho_{\pi_E}) - \lambda H(\pi) = D_{JS}(\rho_\pi, \rho_{\pi_E}) - \lambda H(\pi)$$

- It finds a policy whose **occupancy measure minimizes Jensen-Shannon divergence to the expert's.**

Proposed Algorithm

$$\min_{\pi} \max_{D \in (0,1)^{\mathcal{S} \times \mathcal{A}}} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

- We find a saddle point (π, D)

Algorithm

$$\min_{\pi} \max_{D \in (0,1)^{\mathcal{S} \times \mathcal{A}}} \mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$$

- Initialize the **policy** π_θ, and a **discriminator** $D_w : \mathcal{S} \times \mathcal{A} \to (0,1)$
- **Alternatively update** θ and w
  - **Adam for gradient step on** w to increase $\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]$
  - **TPRO step on** θ to decrease $\mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))] - \lambda H(\pi)$
- Discriminator network is a **local cost function** providing learning signal to the policy.
- Taking a policy step that **decreases expected cost** w.r.t $c(s,a) = \log D(s,a)$

TPRO

Try to find the policy $\pi \in \Pi$ that minimizes the cost function $c(s, a)$

$$L(\theta) = \mathbb{E}_{(s,a)\sim p}[c(s, a)]$$

Therefore, its gradient is

$$\begin{aligned}
\nabla_\theta L(\theta) &= \nabla_\theta \int_{s,a} c(s, a)p(s, a) \\
&= \nabla_\theta \int_{s,a} c(s, a)\pi_\theta(a|s)p(s) \\
&= \int_{s,a} c(s, a)\nabla_\theta \pi_\theta(a|s)p(s) \\
&= \int_{s,a} c(s, a)\frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\pi_\theta(a|s)p(s) \\
&= \int_{s,a} c(s, a)\nabla_\theta \log(\pi_\theta(a|s))\pi_\theta(a|s)p(s) \\
&= \mathbb{E}_{(s,a)\sim p}[c(s, a)\nabla_\theta \log(\pi_\theta(a|s))]
\end{aligned}$$

Algorithm

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \tag{17}$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\begin{aligned} &\hat{\mathbb{E}}_{\tau_i}\left[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)\right] - \lambda \nabla_\theta H(\pi_\theta), \\ &\text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \end{aligned} \tag{18}$$

6: **end for**

Experiments

Experimental Settings (Run on OpenAI Gym)

- **Low-dimensional** control tasks: (e.g. Cartpole, Acrobot)
- **High-dimensional** tasks: (e.g. 3D humanoid locomotion)

Procedures

- Generate expert behavior for these tasks by **running TRPO** on the **true cost functions** to create expert policies.
- Run GAIL and other benchmarks on the **generated expert policies**.
- Evaluate imitation performance w.r.t **sample complexity of expert data**.

Benchmarks

- Behavior Cloning
- Feature expectation matching (FEM): with **linear** cost function
- Game-theoretic apprenticeship learning (GTAL): with **convex** cost function

Results

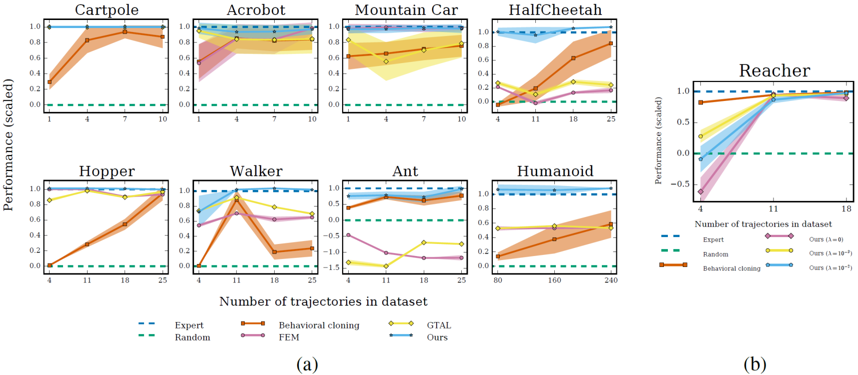


Figure 1: (a) Performance of learned policies. The y-axis is negative cost, scaled so that the expert achieves 1 and a random policy achieves 0. (b) Causal entropy regularization λ on Reacher.

Results

Table 3: Learned policy performance

Task	Dataset size	Behavioral cloning	FEM	GTAL	Ours
Cartpole	1	72.02 ± 35.82	200.00 ± 0.00	200.00 ± 0.00	200.00 ± 0.00
	4	169.18 ± 59.81	200.00 ± 0.00	200.00 ± 0.00	200.00 ± 0.00
	7	188.60 ± 29.61	200.00 ± 0.00	199.94 ± 1.14	200.00 ± 0.00
	10	177.19 ± 52.83	199.75 ± 3.50	200.00 ± 0.00	200.00 ± 0.00
Acrobot	1	-130.60 ± 55.08	-133.14 ± 60.80	-81.35 ± 22.40	-77.26 ± 18.03
	4	-93.20 ± 32.58	-94.21 ± 47.20	-94.80 ± 46.08	-83.12 ± 23.31
	7	-96.92 ± 34.51	-95.08 ± 46.67	-95.75 ± 46.57	-82.56 ± 20.95
	10	-95.09 ± 33.33	-77.22 ± 18.51	-94.32 ± 46.51	-78.91 ± 15.76
Mountain Car	1	-136.76 ± 34.44	-100.97 ± 12.54	-115.48 ± 36.35	-101.55 ± 10.32
	4	-133.25 ± 29.97	-99.29 ± 8.33	-143.58 ± 50.08	-101.35 ± 10.63
	7	-127.34 ± 29.15	-100.65 ± 9.36	-128.96 ± 46.13	-99.90 ± 7.97
	10	-123.14 ± 28.26	-100.48 ± 8.14	-120.05 ± 36.66	-100.83 ± 11.40
HalfCheetah	4	-493.62 ± 246.58	734.01 ± 84.59	1008.14 ± 280.42	4515.70 ± 549.49
	11	637.57 ± 1708.10	-375.22 ± 291.13	226.06 ± 307.87	4280.65 ± 1119.93
	18	2705.01 ± 2273.00	343.58 ± 159.66	1084.26 ± 317.02	4749.43 ± 149.04
	25	3718.58 ± 1856.22	502.29 ± 375.78	869.55 ± 447.90	4840.07 ± 95.36
Hopper	4	50.57 ± 0.95	3571.98 ± 6.35	3065.21 ± 147.79	3614.22 ± 7.17
	11	1025.84 ± 266.86	3572.30 ± 12.03	3502.71 ± 14.54	3615.00 ± 4.32
	18	1949.09 ± 500.61	3230.68 ± 4.58	3201.05 ± 6.74	3600.70 ± 4.24
	25	3383.96 ± 657.61	3331.05 ± 3.55	3458.82 ± 5.40	3560.85 ± 3.09
Walker	4	32.18 ± 1.25	3648.17 ± 327.41	4945.90 ± 65.97	4877.98 ± 2848.37
	11	5946.81 ± 1733.73	4723.44 ± 117.18	6139.29 ± 91.48	6850.27 ± 39.19
	18	1263.82 ± 1347.74	4184.34 ± 485.54	5288.68 ± 37.29	6964.68 ± 46.30
	25	1599.36 ± 1456.59	4368.15 ± 267.17	4687.80 ± 186.22	6832.01 ± 254.64
Ant	4	1611.75 ± 359.54	-2052.51 ± 49.41	-5743.81 ± 723.48	3186.80 ± 903.57
	11	3065.59 ± 635.19	-4462.70 ± 53.84	-6252.19 ± 409.42	3306.67 ± 988.39
	18	2597.22 ± 1366.57	-5148.62 ± 37.80	-3067.07 ± 177.20	3033.87 ± 1460.96
	25	3235.73 ± 1186.38	-5122.12 ± 703.19	-3271.37 ± 226.66	4132.90 ± 878.67
Humanoid	80	1397.06 ± 1057.84	5093.12 ± 583.11	5096.43 ± 24.96	10200.73 ± 1324.47
	160	3655.14 ± 3714.28	5120.52 ± 17.07	5412.47 ± 19.53	10119.80 ± 1254.73
	240	5660.53 ± 3600.70	5192.34 ± 24.59	5145.94 ± 21.13	10361.94 ± 61.28
Task	**Dataset size**	**Behavioral cloning**	**Ours ($\lambda = 0$)**	**Ours ($\lambda = 10^{-3}$)**	**Ours ($\lambda = 10^{-2}$)**
Reacher	4	-10.97 ± 7.07	-67.23 ± 88.99	-32.37 ± 39.81	-46.72 ± 82.88
	11	-6.23 ± 3.29	-6.06 ± 5.36	-6.61 ± 5.11	-9.26 ± 21.88
	18	-4.76 ± 2.31	-8.25 ± 21.99	-5.66 ± 3.15	-5.04 ± 2.22

References

- **Paper Link:** https://arxiv.org/pdf/1606.03476.pdf - **2016 NIPS paper**
- **Useful blog links:**
  - https://medium.com/@sanketgujar95/generative-adversarial-imitation-learning-266f45634e60
  - https://hollygrimm.com/rl_gail
- **Code links:**
  - https://github.com/hollygrimm/gail-mujoco
  - https://github.com/andrewliao11/gail-tf
- **Youtube links:**
  - **Imitation learning tutorial:** https://www.youtube.com/watch?v=6rZTaboSY4k
  - **Author presentation:** https://www.youtube.com/watch?v=bcnCo9RxhB8