Generative Adversarial Imitation Learning

Stefano Ermon
Joint work with Jayesh Gupta, Jonathan Ho, Yunzhu Li, Hongyu Ren, and Jiaming Song

Stanford University
Reinforcement Learning

- Goal: Learn policies
- High-dimensional, raw observations
Reinforcement Learning

- **MDP**: Model for (stochastic) *sequential decision making* problems
  
- **States** $S$
- **Actions** $A$
- **Cost** function (immediate): $C: S \times A \rightarrow R$
- **Transition Probabilities**: $P(s' | s, a)$

- **Policy**: mapping from states to actions
  - E.g., $(S_0 \rightarrow a_1, S_1 \rightarrow a_0, S_2 \rightarrow a_0)$

- **Reinforcement learning**: minimize total (expected, discounted) cost
  \[
  \sum_{t=0}^{T-1} c(S_t)
  \]
Reinforcement Learning

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

- **Cost Function** \(c(s,a)\)
- **Environment (MDP)**
  - States \(S\)
  - Actions \(A\)
  - Transitions: \(P(s'|s,a)\)
- **Policy**: mapping from states to actions
  - E.g., \((S_0 \rightarrow a_1, S_1 \rightarrow a_0, S_2 \rightarrow a_0)\)
- **Reinforcement Learning (RL)**
- **Optimal policy** \(\pi\)

**Cost**

RL needs cost signal
Imitation

Input: expert behavior generated by $\pi_E$

$$\{(s^i_0, a^i_0, s^i_1, a^i_1, \ldots )\}_{i=1}^n \sim \pi_E$$

Goal: learn cost function (reward) or policy

(Ng and Russell, 2000), (Abbeel and Ng, 2004; Syed and Schapire, 2007), (Ratliff et al., 2006), (Ziebart et al., 2008), (Kolter et al., 2008), (Finn et al., 2016), etc.
Behavioral Cloning

- Small errors compound over time (*cascading errors*)
- *Decisions are purposeful* (*require planning*)
Inverse RL

- An approach to imitation
- Learns a cost $c$ such that

$$\pi_E = \arg\min_{\pi \in \Pi} \mathbb{E}_\pi[c(s, a)]$$
Problem setup

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

**Cost Function** $c(s)$ → **Reinforcement Learning (RL)** → **Optimal policy** $\pi$

**Environment (MDP)** → **Inverse Reinforcement Learning (IRL)** → **Expert’s Trajectories** $s_0, s_1, s_2, \ldots$

**Cost Function** $c(s)$

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

(Ziebart et al., 2010; Rust 1987)

Everything else has high cost

Expert has small cost
Problem setup

Reinforcement Learning (RL)

Environment (MDP)

Inverse Reinforcement Learning (IRL)

Optimal policy $\pi$

Cost Function $c(s)$

Expert’s Trajectories $s_0, s_1, s_2, \ldots$

Convex cost regularizer

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

Theorem: ψ-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert's, as measured by ψ* (convex conjugate of ψ)

$$RL \circ IRL_\psi(\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$
Theorem: \( \psi \)-regularized inverse reinforcement learning, implicitly, seeks a policy whose occupancy measure is close to the expert’s, as measured by \( \psi^* \)

- Typical IRL definition: finding a cost function \( c \) such that the expert policy is uniquely optimal w.r.t. \( c \)

- Alternative view: IRL as a procedure that tries to induce a policy that matches the expert’s occupancy measure (generative model)
Special cases

\[ \text{RL} \circ \text{IRL}_\psi (\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E}) \]

• If \( \psi(c) = \text{constant} \), then \( \rho_{\tilde{\pi}} = \rho_{\pi_E} \)
  – Not a useful algorithm. In practice, we only have sampled trajectories

• **Overfitting**: Too much flexibility in choosing the cost function (and the policy)
Towards Apprenticeship learning

• Solution: use features $f_{s,a}$
• Cost $c(s,a) = \theta \cdot f_{s,a}$

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

Only these “simple” cost functions are allowed

$\psi(c) = \infty$

Linear in features

$\psi(c) = 0$

All cost functions
Apprenticeship learning

- For that choice of $\psi$, $RL \circ IRL_\psi$ framework gives apprenticeship learning

$$RL \circ IRL_\psi(\pi_E) = \arg\min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E})$$

- Apprenticeship learning: find $\pi$ performing better than $\pi_E$ over costs linear in the features
  - Abbeel and Ng (2004)
  - Syed and Schapire (2007)
Apprenticeship learning

• Given \( \{(s_i^0, a_i^0, s_i^1, a_i^1, \ldots)\}_{i=1}^n \sim \pi_E \)

• Goal: find \( \pi \) performing better than \( \pi_E \) over a class of costs

\[
\minimize \max_{\pi} \max_{c \in C} \mathbb{E}_\pi[c(s, a)] - \mathbb{E}_{\pi_E}[c(s, a)]
\]

Approximated using demonstrations
Issues with Apprenticeship learning

• Need to craft features very carefully
  – unless the true expert cost function (assuming it exists) lies in C, there is no guarantee that AL will recover the expert policy

• $\text{RL} \circ \text{IRL}_\psi(\pi_E)$ is “encoding” the expert behavior as a cost function in C.
  – it might not be possible to decode it back if C is too simple

\[\pi_E \xrightarrow{\text{IRL}} \pi_p \xrightarrow{\text{RL}} \pi_{\text{opt}}\]
Generative Adversarial Imitation Learning

- **Solution**: use a more expressive class of cost functions

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

where \( g(x) = \begin{cases} -x - \log(1 - e^x) & \text{if } x < 0 \\ +\infty & \text{otherwise} \end{cases} \)
Generative Adversarial Imitation Learning

- $\psi^* = \text{optimal negative log-loss of the binary classification problem of distinguishing between state-action pairs of } \pi \text{ and } \pi_E$

$$
\psi^*_A(\rho_\pi - \rho_{\pi_E}) = \sup_{D \in (0,1)^S \times A} \mathbb{E}_\pi[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a))]
$$
Generative Adversarial Networks

Figure from Goodfellow et al, 2014
GAIL

Differentiable function $D$

D tries to output 0
Sample from expert

Differentiable function $D$

D tries to output 1
Sample from model

Black box simulator
Differentiable function $P$

Ho and Ermon, *Generative Adversarial Imitation Learning*
How to optimize the objective

• Previous Apprenticeship learning work:
  – Full dynamics model
  – Small environment
  – Repeated RL

• We propose: gradient descent over policy parameters (and discriminator)

J. Ho, J. K. Gupta, and S. Ermon. Model-free imitation learning with policy optimization. ICML 2016.
Properties

• Inherits pros of policy gradient
  – Convergence to local minima
  – Can be model free

• Inherits cons of policy gradient
  – High variance
  – Small steps required
Properties

• Inherit pros of policy gradient
  – Convergence to local minima
  – Can be model free

• Inherit cons of policy gradient
  – High variance
  – Small steps required

• Solution: trust region policy optimization
Results
Results

Input: driving demonstrations (Torcs)

Output policy: From raw visual inputs

Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations
Experimental results

[Graphs showing performance metrics for different tasks such as Cartpole, Acrobot, Mountain Car, HalfCheetah, Hopper, Walker, Ant, Humanoid. Each graph plots performance against the number of trajectories in the dataset, with different lines representing expert, behavioral cloning, GAIL (ours), Random, and FEM.]
Latent structure in demonstrations

Human model

Latent variables $z$  
Policy

Environment

Observed Behavior

Semantically meaningful latent structure?
Latent variables $z$ for InfoGAIL

- Add Smiling
- Remove Smiling
- Add Eyeglass
- Remove Eyeglass

Environment
Policy

Maximize mutual information

Infer structure
Hou el al.

Observed data
Observed Behavior
InfoGAIL

\[ L_I(\pi_\theta, Q_\psi) = \mathbb{E}_{c \sim p(c), a \sim \pi_\theta(\cdot | s, c)} \left[ \log Q_\psi(c | s, a) \right] + H(c) \leq I(c; s, a) \]
Synthetic Experiment

Demonstrations

GAIL

Info-GAIL
Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

InfoGAIL

Latent variables $z$

Policy

Environment

Trajectories

Pass left ($z=0$)

Pass right ($z=1$)
Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

InfoGAIL

Latent variables $z$

Policy

Environment

Trajectories

Turn inside ($z=0$)

Turn outside ($z=1$)
Multi-agent environments

What are the goals of these 4 agents?
Problem setup

Cost Functions
\[ c_1(s,a_1) \]
\[ \ldots \]
\[ c_N(s,a_N) \]

MA Reinforcement Learning (MARL)

Environment (Markov Game)

Optimal policies \( \pi_1 \)

Optimal policies \( \pi_K \)

|   | R   | L   |
|---|-----|-----|
| R | 0,0 | 10,10 |
| L | 10,10 | 0,0 |

DRIVE ON LEFT

DRIVE ON RIGHT
Problem setup

Cost Functions

\[ c_1(s, a_1) \]
\[ \ldots \]
\[ c_N(s, a_N) \]

MA Reinforcement Learning (MARL)

Optimal policies \( \pi \)

Environment

(Markov Game)

Inverse Reinforcement Learning (MAIRL)

≈

(s similar wrt \( \psi \))

Cost Functions

\[ c_1(s, a_1) \]
\[ \ldots \]
\[ c_N(s, a_N) \]

Expert’s Trajectories

\( (s_0, a_{01}, \ldots, a_{0N}) \)
\( (s_1, a_{11}, \ldots, a_{1N}) \)
\[ \ldots \]

\[
MIM_\psi(\pi_E) = \arg \max_{\pi \in \Pi} \max_v \min_{r \in \mathbb{R}^{S \times A}} \mathcal{L}_\psi(\pi_E, v)
\]

\[
\mathcal{L}_\psi(\pi_E, v) = -f_r(\pi, v) + f_r(\pi_E, v) + \psi(r)
\]

\[ r \in \text{MAIRL}(\pi_E) \]
Sample from expert $(s,a_1,a_2,...,a_N)$

Sample from model $(s,a_1,a_2,...,a_N)$

Black box simulator

Policy Agent 1

Policy Agent N

Generator $G$

Song, Ren, Sadigh, Ermon, Multi-Agent Generative Adversarial Imitation Learning
Environments

Demonstrations

MAGAIL
Environments

Demonstrations

MAGAIL
Suboptimal demos

Expert

MAGAIL

lighter plank + bumps on ground
Conclusions

• IRL is a dual of an occupancy measure matching problem (generative modeling)

• Might need flexible cost functions
  – GAN style approach

• Policy gradient approach
  – Scales to high dimensional settings

• Towards unsupervised learning of latent structure from demonstrations