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** \( c(s, a) \)
- **Reinforcement Learning (RL)**
- **Environment (MDP)**

\( C : S \times A \rightarrow R \)

\( RL \) needs cost signal

**Optimal policy** \( \pi \)

- States \( S \)
- Actions \( A \)
- Transitions: \( P(s' | s, a) \)
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)

Environment (MDP)

Inverse Reinforcement Learning (IRL)

Optimal policy $\pi$

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

Cost Function $c(s)$

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

(Ziebart et al., 2010; Rust 1987)

Everything else has high cost

Expert has small cost
Problem setup

Cost Function $c(s)$

Reinforcement Learning (RL)

Environment (MDP)

Inverse Reinforcement Learning (IRL)

Optimal policy $\pi$

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\(\circ\)IRL

Reinforcement Learning (RL) \(\rightarrow\) Optimal policy \(\pi\) \(\approx\) (similar w.r.t. \(\psi\))

\(\psi\)-regularized Inverse Reinforcement Learning (IRL) \(\leftarrow\) Expert’s Trajectories \(s_0, s_1, s_2, \ldots\)

\(\rho_\pi\) = occupancy measure = distribution of state-action pairs encountered when navigating the environment with the policy

\(\rho_{\pi E}\) = Expert’s occupancy measure

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

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

\[\psi(c) = \begin{cases} \infty & \text{Linear in features} \\ 0 & \text{All cost functions} \end{cases}\]
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 \( \{(s_0^i, a_0^i, s_1^i, a_1^i, \ldots)\}_{i=1}^n \sim \pi_E \)

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

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

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
  
• $RL \circ IRL_\psi(\pi_E)$ 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
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

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

Figure from Goodfellow et al, 2014
GAIL

Differentiable function $D$ tries to output 0 when sampled from the expert. Differentiable function $D$ also tries to output 1 when sampled from the model. The generator $G$ is used to generate samples for training. The black box simulator is used to simulate the environment.

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)

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

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

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

• Solution: trust region policy optimization
Results
Results

Input: driving demonstrations (Torcs)

Output policy: Li et al, 2017. InfoGAIL: Interpretable Imitation Learning from Visual Demonstrations

From raw visual inputs
Experimental results

![Graphs showing performance vs. number of trajectories for different environments: Cartpole, Acrobot, Mountain Car, HalfCheetah, Hopper, Walker, Ant, and Humanoid. The graphs depict performance (scaled) on the y-axis and the number of trajectories in the dataset on the x-axis. The performance curves are color-coded to represent different methods: Expert, Random, Behavioral cloning, FEM, GTAL, and GAIL (ours).]
Latent structure in demonstrations

Human model

Latent variables $z$ → Policy → Environment → Observed Behavior

Semantically meaningful latent structure?
InfoGAIL

Latent structure

<table>
<thead>
<tr>
<th>Latent variables</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td></td>
</tr>
</tbody>
</table>

Hou et al.

Maximize mutual information

Environment

Observed data

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

Demonstrations

GAIL

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

InfoGAIL model

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

model

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 \( \pi_K \)

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0,0</td>
<td>10,10</td>
</tr>
<tr>
<td>L</td>
<td>10,10</td>
<td>0,0</td>
</tr>
</tbody>
</table>
Problem setup

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


depends on

MA Reinforcement Learning (MARL)

Environment (Markov Game)

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

Inverse Reinforcement Learning (MAIRL)

Optimal policies \( \pi \)

\[ \approx \text{(similar wrt } \psi) \]

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

Generator G

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

Diff. function $D_1$ tries to output 0

Diff. function $D_2$ tries to output 0

... Diff. function $D_N$ tries to output 0

Policy Agent 1

Black box simulator

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

Diff. function $D_1$ tries to output 1

Diff. function $D_2$ tries to output 1

... Diff. function $D_N$ tries to output 1

Policy Agent N

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