Generative Adversarial Imitation Learning

Jonathan Ho and Stefano Ermon

Topic: Inverse RL
Presenter: Albert Hsueh
Problem Addressed

- Imitation learning for large and maybe complex environments is intractable
- Maybe cost function cannot capture the actual task or is too expensive to evaluate
Limitations of Prior work

- Behaviour cloning require lots of data
- Inverse RL is computationally expensive
- Thus imitation learning for large scale learning environments remain unsolved research challenge
GAIL’s Contribution

1. New algorithm using particular choice of regularizer, $\Psi$
2. Proved mathematically that the new algorithm is similar to GAN
3. State of the art performance -- beat 9 environment baselines and demonstrated feasibility of algo on increasingly larger environments
Background Concepts
Important Background Concepts

- The vanilla learning process: RL(IRL(\pi_E))
- IRL’s cross entropy formulation
- Regularization & model capacity
- Occupancy measure
RL(IRL(pi_E))
RL(IRL(pi_E))
$RL(IRL(pi_E))$

\[\pi E \rightarrow \text{IRL \hspace{1cm} Cost function} \rightarrow \text{RL} \rightarrow \pi\]

Slow, and perhaps not expressive enough
RL(IRL(\pi_E))
Update Cost

Inverse RL

Compare with experts

Gradient computation.

Gradient steps

Run RL

https://medium.com/@sanketgajar95/generative-adversarial-imitation-learning-266f45634e60
RL(IRL($\pi_E$))

One Step?
Important Background Concepts

- The vanilla learning process: RL(IRL(\pi_E))
- IRL’s cross entropy formulation
- Regularization & model capacity
- Occupancy measure
IRL’s cross entropy formulation

- Key point here is that the H entropy term add noise to the system and enable exploration
Important Background Concepts

- The vanilla learning process: $RL(\text{IRL}(\pi_E))$
- IRL’s cross-entropy formulation
- Regularization & model capacity
- Occupancy measure
Regularization & model capacity

- Use expressive, high capacity models (DNN) to learn cost functions may overfit easily given finite dataset (particularly since expert dataset usually small) so we need to regularize
  - High capacity models necessary for high us to rationalize expert behavior without hand crafting features

- Turns out that regularizer dictates what algorithm class gets recovered
Regularization & model capacity

- Use expressive models. They may overfit easily on small, noisy dataset usually.
  - High capacity models can learn the expert behavior without hand-tuning cost functions.
  - Often, overfitting is not a problem since expert behavior can be recovered.

- Turns out that model capacity gets recovered.
Important Background Concepts

- The vanilla learning process: \( RL(\text{IRL}(\pi_E)) \)
- IRL's cross-entropy formulation
- Regularization & model capacity
- Occupancy measure
Occupancy Measure

\[ E_\pi[c(s, a)] = \sum_{s,a} \rho_\pi(s, a)c(s, a) \] for any cost function \( c \).

- “Distribution of \((s,a)\) that agent encounters” given its policy

**Proposition 3.1** (Theorem 2 of Syed et al. [29]). If \( \rho \in D \), then \( \rho \) is the occupancy measure for

\[ \pi_\rho(a|s) \triangleq \frac{\rho(s, a)}{\sum_{a'} \rho(s, a')} \], and \( \pi_\rho \) is the only policy whose occupancy measure is \( \rho \).
Algorithm Design Method (key takeaways)

- Proposition 3.1 + lemma 3.1 -> duality of IRL/occupancy matching
- Choice of regularizer gives rise to different known algorithm classes
- Lead to GAN loss function
Derivation
\[
\text{maximize } \left( \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)] \right) - \mathbb{E}_{\pi_E}[c(s, a)]
\] (1)
• Outer optimization wants to find the cost function that can distinguish between the expert and some “worst-case”/“most like expert” policy $\pi^*$ that there is
Inner optimization finds our \( \pi^* \) from all possible \( \pi \) by selecting the \( \pi \) that generate trajectories with the lowest average cost.
Inner optimization finds our $\pi^*$ from all possible $\pi$ by selecting the $\pi$ that generate trajectories with the lowest average cost.

By doing so we essentially are doing RL in the “inner loop”

$$RL(c) = \arg \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi [c(s, a)]$$  \hspace{1cm} (2)
\[
\max_{\pi \in \Pi} \left( \min_{c \in C} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)] \right) - \mathbb{E}_{\pi_E}[c(s, a)]
\] (1)

\[
\text{IRL}_\psi(\pi_E) = \arg \max_{c \in \mathbb{R}^{S \times A}} -\psi(c) + \left( \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s, a)] \right) - \mathbb{E}_{\pi_E}[c(s, a)]
\] (3)

Add regularization to prevent overfitting
maximize \left( \min_{\pi \in \Pi} -H(\pi) + \mathbb{E}_\pi [c(s, a)] \right) - \mathbb{E}_{\pi_E} [c(s, a)] \tag{1}

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)] \tag{3}

RL \circ IRL_\psi(\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E}) \tag{4}

By Proposition 3.2’s proof in the Appendix
Interlude -- Regularizer design

- Low penalty on functions that give negative cost to expert trajectories
- Heavily penalize otherwise

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

where \( g(x) = \begin{cases} 
-x - \log(1 - e^x) & \text{if } x < 0 \\
+\infty & \text{otherwise}
\end{cases} \)
Interlude -- Regularizer design

- Low penalty on functions that give negative cost to expert trajectories
- Heavily penalize all

\[
\psi_{GA}(c) = \begin{cases} 
\mathbb{E}_{\pi_E}[g(c)] & \text{if } c \neq +\infty \\
+\infty & \text{otherwise}
\end{cases}
\]

\[
y(x) = \begin{cases} 
1 - e^x & \text{if } x < 0 \\
+\infty & \text{otherwise}
\end{cases}
\]
Interlude -- Regularizer design

- Constant regularizer -> Exact Occupancy Matching
  - large environments and few expert samples makes it intractable
Interlude -- Regularization

- Constant regularization
  - large environment
    - makes it intractable

http://www dspguide com/ch3/2.htm
Interlude -- Regularizer design

- Constant regularizer -> Exact Occupancy Matching
  - large environments and few expert samples makes it intractable
- Indicator regularizer -> apprenticeship learning
  - Scales well to large environments but require careful tuning to exact match
\[ RL \circ IRL_\psi(\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E}) \]
\[
\text{One Step [done, eqn. 4]}
\]

\[
\begin{align*}
\text{RL} \circ \text{IRL}_\psi (\pi_E) &= \arg \min_{\pi \in \Pi} -H(\pi) + \psi^* (\rho_\pi - \rho_{\pi_E}) \\
\psi^*_\text{GA} (\rho_\pi - \rho_{\pi_E}) &= \max_{D \in (0,1)^{S \times A}} \mathbb{E}_\pi [\log(D(s,a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s,a))]
\end{align*}
\]  

Eqn. 14 is a fact from Corollary A.1.1
\[ \text{One Step [done, eqn. 4]} \]

\[ \text{RL} \circ \text{IRL}_\psi (\pi_E) = \arg \min_{\pi \in \Pi} -H(\pi) + \psi^*(\rho_\pi - \rho_{\pi_E}) \]  
\[ \psi^*_\text{GA}(\rho_\pi - \rho_{\pi_E}) = \max_{D \in (0,1)^S \times A} \mathbb{E}_\pi [\log(D(s,a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s,a))] \]  
\[ \mathbb{E}_\pi [\log(D(s,a))] + \mathbb{E}_{\pi_E} [\log(1 - D(s,a))] - \lambda H(\pi) \]
GAN Loss

- Train a discriminator $D$ to minimize this loss
- Train a generator to maximize this same loss

$$
E_\pi [\log(D(s, a))] + E_{\pi_E} [\log(1 - D(s, a))] - \lambda H(\pi)
$$

(16)
GAN Loss

- Train a discriminator $D$ to minimize this loss
- Train a generator to maximize this same loss

$$
E_\pi [\log(D(s, a))] + E_{\pi_E} [\log(1 - D(s, a))] - \lambda H(\pi) \tag{16}
$$

$D(s, a)$

= 0 if true distribution (ie drawn from expert)
= 1 if false distribution (ie done by learner’s policy)
The algo

**Algorithm 1** Generative adversarial imitation learning

1. **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters $\theta_0$, $w_0$
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

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

(17)

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

$$\hat{E}_{\tau_i} [\nabla_{\theta} \log \pi_{\theta}(a|s)Q(s, a)] - \lambda \nabla_{\theta} H(\pi_{\theta}),$$

where $Q(\bar{s}, \bar{a}) = \hat{E}_{\tau_i} [\log(D_{w_{i+1}}(s, a)) | s_0 = \bar{s}, a_0 = \bar{a}]$

(18)

6. end for
Results
## Table 1: Environments

<table>
<thead>
<tr>
<th>Task</th>
<th>Observation space</th>
<th>Action space</th>
<th>Random policy performance</th>
<th>Expert performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartpole-v0</td>
<td>4 (continuous)</td>
<td>2 (discrete)</td>
<td>18.64 ± 7.45</td>
<td>200.00 ± 0.00</td>
</tr>
<tr>
<td>Acrobot-v0</td>
<td>4 (continuous)</td>
<td>3 (discrete)</td>
<td>-200.00 ± 0.00</td>
<td>-75.25 ± 10.94</td>
</tr>
<tr>
<td>Mountain Car-v0</td>
<td>2 (continuous)</td>
<td>3 (discrete)</td>
<td>-200.00 ± 0.00</td>
<td>-98.75 ± 8.71</td>
</tr>
<tr>
<td>Reacher-v1</td>
<td>11 (continuous)</td>
<td>2 (continuous)</td>
<td>-43.21 ± 4.32</td>
<td>-4.09 ± 1.70</td>
</tr>
<tr>
<td>HalfCheetah-v1</td>
<td>17 (continuous)</td>
<td>6 (continuous)</td>
<td>-282.43 ± 79.53</td>
<td>4463.46 ± 105.83</td>
</tr>
<tr>
<td>Hopper-v1</td>
<td>11 (continuous)</td>
<td>3 (continuous)</td>
<td>14.47 ± 7.96</td>
<td>3571.38 ± 184.20</td>
</tr>
<tr>
<td>Walker-v1</td>
<td>17 (continuous)</td>
<td>6 (continuous)</td>
<td>0.57 ± 4.59</td>
<td>6717.08 ± 845.62</td>
</tr>
<tr>
<td>Ant-v1</td>
<td>111 (continuous)</td>
<td>8 (continuous)</td>
<td>-69.68 ± 111.10</td>
<td>4228.37 ± 424.16</td>
</tr>
<tr>
<td>Humanoid-v1</td>
<td>376 (continuous)</td>
<td>17 (continuous)</td>
<td>122.87 ± 35.11</td>
<td>9575.40 ± 1750.80</td>
</tr>
</tbody>
</table>
## Table 1: Environment Spaces

<table>
<thead>
<tr>
<th>Task</th>
<th>Observation space</th>
<th>Action space</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartpole-v0</td>
<td>4 (continuous)</td>
<td>2 (discrete)</td>
<td>0.90</td>
</tr>
<tr>
<td>Acrobat-v0</td>
<td>4 (continuous)</td>
<td>3 (discrete)</td>
<td>0.94</td>
</tr>
<tr>
<td>Mountain Car-v0</td>
<td>2 (continuous)</td>
<td>3 (discrete)</td>
<td>1.71</td>
</tr>
<tr>
<td>Reacher-v1</td>
<td>11 (continuous)</td>
<td>2 (continuous)</td>
<td>2.70</td>
</tr>
<tr>
<td>HalfCheetah-v1</td>
<td>17 (continuous)</td>
<td>6 (continuous)</td>
<td>3.83</td>
</tr>
<tr>
<td>Hopper-v1</td>
<td>11 (continuous)</td>
<td>3 (continuous)</td>
<td>4.20</td>
</tr>
<tr>
<td>Walker-v1</td>
<td>17 (continuous)</td>
<td>6 (continuous)</td>
<td>5.62</td>
</tr>
<tr>
<td>Ant-v1</td>
<td>111 (continuous)</td>
<td>8 (continuous)</td>
<td>4.16</td>
</tr>
<tr>
<td>Humanoid-v1</td>
<td>376 (continuous)</td>
<td>17 (continuous)</td>
<td>10.80</td>
</tr>
<tr>
<td>Task</td>
<td>Observations</td>
<td>Expert performance</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------</td>
<td>---------------------------------</td>
<td></td>
</tr>
<tr>
<td>Cartpole-v0</td>
<td>4 (cc)</td>
<td>200.00 ± 0.00</td>
<td></td>
</tr>
<tr>
<td>Acrobot-v0</td>
<td>4 (cc)</td>
<td>−75.25 ± 10.94</td>
<td></td>
</tr>
<tr>
<td>Mountain Car-v0</td>
<td>2 (cc)</td>
<td>−98.75 ± 8.71</td>
<td></td>
</tr>
<tr>
<td>Reacher-v1</td>
<td>11 (c)</td>
<td>−4.09 ± 1.70</td>
<td></td>
</tr>
<tr>
<td>HalfCheetah-v1</td>
<td>17 (c)</td>
<td>4463.46 ± 105.83</td>
<td></td>
</tr>
<tr>
<td>Hopper-v1</td>
<td>11 (c)</td>
<td>3571.38 ± 184.20</td>
<td></td>
</tr>
<tr>
<td>Walker-v1</td>
<td>17 (c)</td>
<td>6717.08 ± 845.62</td>
<td></td>
</tr>
<tr>
<td>Ant-v1</td>
<td>111 (c)</td>
<td>4228.37 ± 424.16</td>
<td></td>
</tr>
<tr>
<td>Humanoid-v1</td>
<td>376 (c)</td>
<td>9575.40 ± 1750.80</td>
<td></td>
</tr>
</tbody>
</table>

- **FetchPickAndPlace-v1**: Lift a block into the air.
- **FetchPush-v1**: Push a block to a goal position.
- **FetchReach-v1**: Move Fetch to a goal position.
- **FetchSlide-v1**: Slide a puck to a goal position.
- **HandManipulateBlock-v0**: Orient a block using a robot hand.
- **HandManipulateEgg-v0**: Orient an egg using a robot hand.
Results
<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset size</th>
<th>Behavioral cloning</th>
<th>FEM</th>
<th>GTAL</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartpole</td>
<td>1</td>
<td>72.02 ± 35.82</td>
<td>200.00 ± 0.00</td>
<td>200.00 ± 0.00</td>
<td>200.00 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>169.18 ± 59.81</td>
<td>200.00 ± 0.00</td>
<td>200.00 ± 0.00</td>
<td>200.00 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>188.60 ± 29.61</td>
<td>200.00 ± 0.00</td>
<td>199.94 ± 1.14</td>
<td>200.00 ± 0.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>177.19 ± 52.83</td>
<td>199.75 ± 3.50</td>
<td>200.00 ± 0.00</td>
<td>200.00 ± 0.00</td>
</tr>
<tr>
<td>Acrobat</td>
<td>1</td>
<td>-130.60 ± 55.08</td>
<td>-133.14 ± 60.80</td>
<td>-81.35 ± 22.40</td>
<td>-77.26 ± 18.03</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-93.20 ± 32.58</td>
<td>-94.21 ± 47.20</td>
<td>-94.80 ± 46.08</td>
<td>-83.12 ± 23.31</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>-96.92 ± 34.51</td>
<td>-95.08 ± 46.67</td>
<td>-95.75 ± 46.57</td>
<td>-82.56 ± 20.95</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>-95.09 ± 33.33</td>
<td>-77.22 ± 18.51</td>
<td>-94.32 ± 45.61</td>
<td>-78.91 ± 15.76</td>
</tr>
<tr>
<td>Mountain Car</td>
<td>1</td>
<td>-136.76 ± 34.44</td>
<td>-100.97 ± 12.54</td>
<td>-115.48 ± 36.35</td>
<td>-101.55 ± 10.32</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-133.25 ± 29.97</td>
<td>-99.29 ± 8.33</td>
<td>-143.58 ± 50.08</td>
<td>-101.35 ± 10.63</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>-127.34 ± 29.15</td>
<td>-100.65 ± 9.36</td>
<td>-128.96 ± 46.13</td>
<td>-99.90 ± 7.97</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>-123.14 ± 28.26</td>
<td>-100.48 ± 8.14</td>
<td>-120.05 ± 36.66</td>
<td>-100.83 ± 11.40</td>
</tr>
<tr>
<td>HalfCheetah</td>
<td>4</td>
<td>-493.62 ± 246.58</td>
<td>734.01 ± 84.59</td>
<td>1008.14 ± 280.42</td>
<td>4515.70 ± 549.49</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>637.57 ± 1708.10</td>
<td>-375.22 ± 291.13</td>
<td>226.06 ± 307.87</td>
<td>4280.65 ± 1119.93</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>2705.01 ± 2737.00</td>
<td>343.58 ± 159.66</td>
<td>1084.26 ± 317.02</td>
<td>4749.43 ± 149.04</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>3718.58 ± 1856.25</td>
<td>502.29 ± 375.78</td>
<td>869.55 ± 447.90</td>
<td>4840.07 ± 95.36</td>
</tr>
<tr>
<td>Hopper</td>
<td>4</td>
<td>50.57 ± 0.95</td>
<td>3571.98 ± 6.35</td>
<td>3065.21 ± 147.79</td>
<td>3614.22 ± 7.17</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>1025.84 ± 266.86</td>
<td>3572.30 ± 12.03</td>
<td>3502.71 ± 14.54</td>
<td>3615.00 ± 4.32</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>1949.09 ± 500.61</td>
<td>3230.68 ± 4.58</td>
<td>3201.05 ± 6.74</td>
<td>3600.70 ± 4.24</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>3383.96 ± 657.61</td>
<td>3331.05 ± 3.55</td>
<td>3458.82 ± 5.40</td>
<td>3560.85 ± 3.09</td>
</tr>
<tr>
<td>Walker</td>
<td>4</td>
<td>32.18 ± 1.25</td>
<td>3648.17 ± 327.41</td>
<td>4945.80 ± 65.97</td>
<td>4877.98 ± 2848.37</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>5946.81 ± 1733.73</td>
<td>4723.44 ± 117.18</td>
<td>6139.29 ± 91.48</td>
<td>6850.27 ± 39.19</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>1268.82 ± 1347.74</td>
<td>4184.34 ± 485.54</td>
<td>5288.68 ± 37.29</td>
<td>6964.68 ± 46.30</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1599.36 ± 1456.59</td>
<td>4386.15 ± 267.17</td>
<td>4867.80 ± 186.22</td>
<td>6832.01 ± 254.64</td>
</tr>
<tr>
<td>Ant</td>
<td>4</td>
<td>1011.75 ± 359.54</td>
<td>-2052.51 ± 49.41</td>
<td>-5743.81 ± 723.48</td>
<td>3186.80 ± 903.57</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>3065.59 ± 633.19</td>
<td>-4462.70 ± 53.84</td>
<td>-6252.19 ± 409.42</td>
<td>3306.67 ± 988.39</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>2597.22 ± 1366.57</td>
<td>-5148.62 ± 37.80</td>
<td>-3067.07 ± 177.20</td>
<td>3033.87 ± 1460.96</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>3235.73 ± 1186.38</td>
<td>-5122.12 ± 703.19</td>
<td>-3271.37 ± 226.46</td>
<td>4132.90 ± 878.67</td>
</tr>
<tr>
<td>Humanoid</td>
<td>80</td>
<td>1397.06 ± 1057.84</td>
<td>5093.12 ± 533.11</td>
<td>5096.43 ± 24.96</td>
<td>10200.73 ± 1324.47</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>3655.14 ± 3714.28</td>
<td>5120.52 ± 17.07</td>
<td>5412.47 ± 19.53</td>
<td>10119.80 ± 1254.73</td>
</tr>
<tr>
<td></td>
<td>240</td>
<td>5660.53 ± 3600.70</td>
<td>5192.34 ± 24.59</td>
<td>5145.94 ± 21.13</td>
<td>10361.94 ± 61.28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>Dataset size</th>
<th>Behavioral cloning</th>
<th>Ours (λ = 0)</th>
<th>Ours (λ = 10⁻³)</th>
<th>Ours (λ = 10⁻²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reacher</td>
<td>4</td>
<td>-10.97 ± 7.07</td>
<td>-67.23 ± 88.99</td>
<td>-32.37 ± 39.81</td>
<td>-46.72 ± 82.88</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>-6.23 ± 3.29</td>
<td>-6.06 ± 5.36</td>
<td>-6.61 ± 5.11</td>
<td>-9.26 ± 21.88</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>-4.76 ± 2.31</td>
<td>-8.25 ± 21.99</td>
<td>-5.66 ± 3.15</td>
<td>-5.04 ± 2.22</td>
</tr>
</tbody>
</table>
Discussion of results

- Shows evidence that gail out performs benchmark on increasingly larger environments
- Which environments does GAIL fail?
  - Images
  - Stochastic Dynamics
  - Robotics gym env
Very good paper but -- Limitations

- Slow learning
- Qualitative performance not captured in metric
- Does not comment on whether there are better regularizers. Seems intuitive to apply gan to imitation learning but obviously the math proofs to show the equivalence is huge.
- Difficult paper to understand so less experimental insights is unfortunate
Recap

1. Occupancy measure is unique to a policy
2. Optimizing occupancy measure is the dual to optimizing over policy
3. We use this to reframe the vanilla imitation learning: RL(IRL(\pi_E)) problem
4. Entropy allows for better exploration
5. Regularization allows us to recover different algorithm classes
6. Particular regularizer was chosen to allow scalability to large environments while also allowing exact matching of expert policy/occupancy measure
7. The conjugate of the regularizer + vanilla imitation learning resembles GAN loss