Learning robust rewards with adversarial inverse reinforcement learning (J. Fu, K. Luo, S. Levine)

Markus Vieth

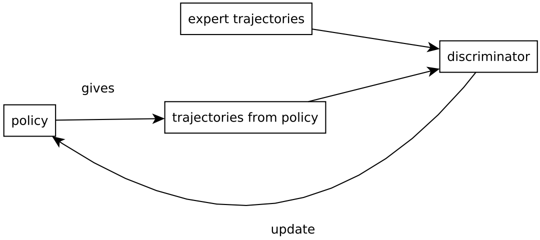
June 20, 2018

Inverse Reinforcement Learning (IRL)

- Infer an expert's reward function from demonstrations
- Alternative: imitation learning, which does not recover reward
- Advantage: re-optimize reward in novel environment, analyze agent's intentions

Adversarial learning

- Two agents working against each other
- GANs try to mimic an input data set
- Here: discriminator tries to classify expert data from policy samples, policy tries to confuse the discriminator
- Advantage: policy (generator) does not see the data set directly, better generalization



Disentangled rewards

- Learned reward function can only depent on current state s
- Or else reward not robust to changes in dynamics
- Reward transformation: $\hat{r}(s, a, s') = r(s, a, s') + \gamma\phi(s') - \phi(s)$
- $D_{\theta,\phi} = \frac{exp\{f_{\theta,\phi}(s,a,s')\}}{exp\{f_{\theta,\phi}(s,a,s')\}+\pi(a|s)}$
- $f_{\theta,\phi}(s, a, s') = g_\theta(s, a) + \gamma h_\phi(s') - h_\phi(s)$
- With reward approximator g_θ and shaping term h_ϕ

Code

$$D_{\theta,\phi} = \frac{exp\{f_{\theta,\phi}(s,a,s')\}}{exp\{f_{\theta,\phi}(s,a,s')\}+\pi(a|s)} = \exp\left(f_{\theta,\phi} - \log(\exp(f_{\theta,\phi}) + \pi(a|s))\right)$$

$$f_{\theta,\phi}(s,a,s') = g_\theta(s,a) + \gamma h_\phi(s') - h_\phi(s)$$

From airl_state.py:

```
self.reward = reward_arch(rew_input, dout=1, **reward_arch_args)
# value function shaping
fitted_value_fn_n = value_fn_arch(self.nobs_t, dout=1)
self.value_fn = fitted_value_fn = value_fn_arch(self.obs_t, dout=1)
# Define log p_tau(a|s) = r + gamma * V(s') - V(s)
self.qfn = self.reward + self.gamma*fitted_value_fn_n
log_p_tau = self.reward  + self.gamma*fitted_value_fn_n - fitted_value_fn
log_q_tau = self.lprobs
log_pq = tf.reduce_logsumexp([log_p_tau, log_q_tau], axis=0)
self.discrim_output = tf.exp(log_p_tau-log_pq)
self.loss = -tf.reduce_mean(self.labels*(log_p_tau-log_pq)
                 + (1-self.labels)*(log_q_tau-log_pq))
```

Algorithm: Adversarial inverse reinforcement learning

Obtain expert trajectories τ_i^E
Initialize policy π and discriminator $D_{\theta,\phi}$.
for step t in $\{1, ..., N\}$ **do**
  Collect trajectories $\tau_i = (s_0, a_0, ..., s_T, a_T)$ by executing π.
  Train $D_{\theta,\phi}$ via binary logistic regression to classify expert data τ_i^E from samples τ_i.
  Update reward $r_{\theta,\phi}(s, a, s') \leftarrow \log D_{\theta,\phi}(s, a, s') - \log(1 - D_{\theta,\phi}(s, a, s'))$
  Update π with respect to $r_{\theta,\phi}$ using any policy optimization method.
end for

Code

From irl_batch_polopt.py:

```
#itr is the iteration parameter
paths = self.obtain_samples(itr)
paths = self.compute_irl(paths, itr=itr)
samples_data = self.process_samples(itr, paths)
self.optimize_policy(itr, samples_data)
```

Experiments: Ant

- observation space of 111, action space of 8
- reward = forward_reward - ctrl_cost - contact_cost + flipped_rew
- linear function approximator for the reward term g and a 2-layer ReLU network for the shaping term h
- policy: two-layer (32 units) ReLU gaussian policy

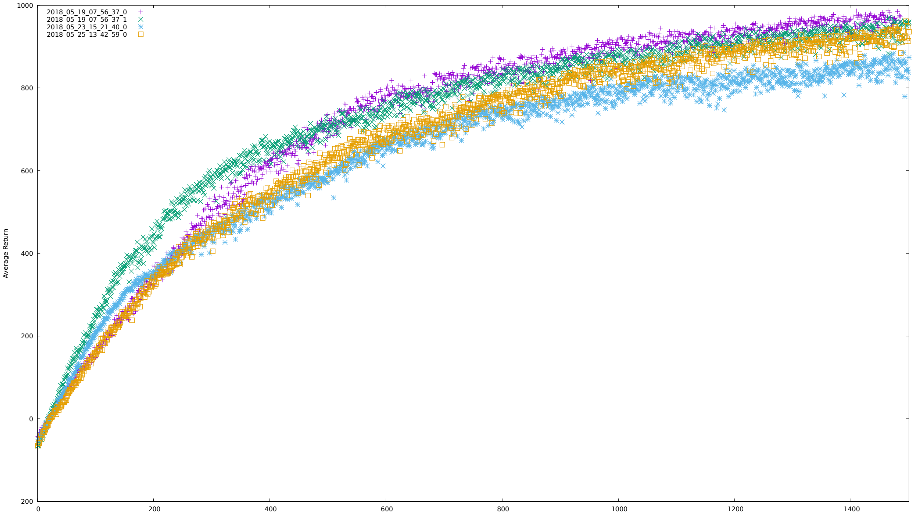
Table 1: Results on transfer learning tasks. Mean scores (higher is better) are reported over 5 runs. We also include results for TRPO optimizing the ground truth reward, and the performance of a policy learned via GAIL on the training environment.

	State-Only?	Pointmass-Maze	Ant-Disabled
GAN-GCL	No	-40.2	-44.8
GAN-GCL	Yes	-41.8	-43.4
AIRL **(ours)**	No	-31.2	-41.4
AIRL **(ours)**	Yes	**-8.82**	**130.3**
GAIL, policy transfer	N/A	-29.9	-58.8
TRPO, ground truth	N/A	-8.45	315.5

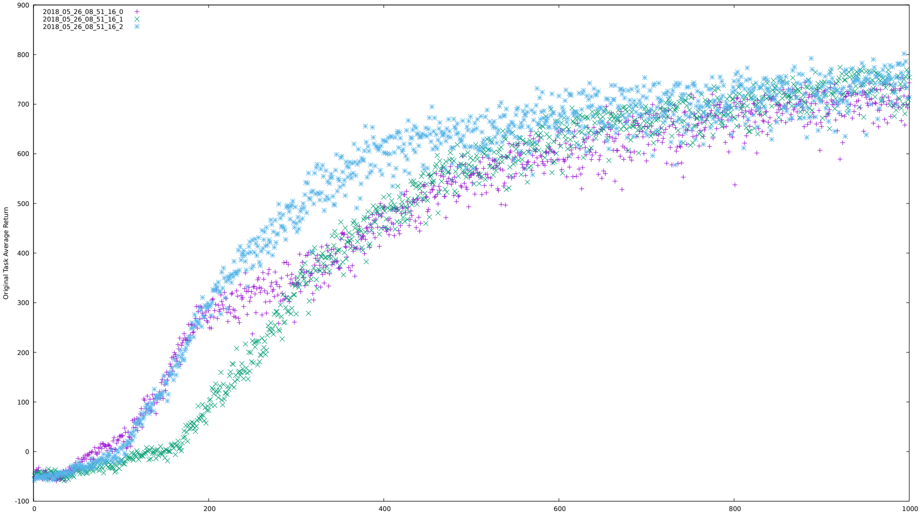
Table 2: Results on imitation learning benchmark tasks. Mean scores (higher is better) are reported across 5 runs.

	Pendulum	Ant	Swimmer	Half-Cheetah
GAN-GCL	-261.5	460.6	-10.6	-670.7
GAIL	-226.0	**1358.7**	**140.2**	1642.8
AIRL **(ours)**	**-204.7**	1238.6	139.1	**1839.8**
Expert (TRPO)	-179.6	1537.9	141.1	1811.2
Random	-654.5	-108.1	-11.5	-988.4

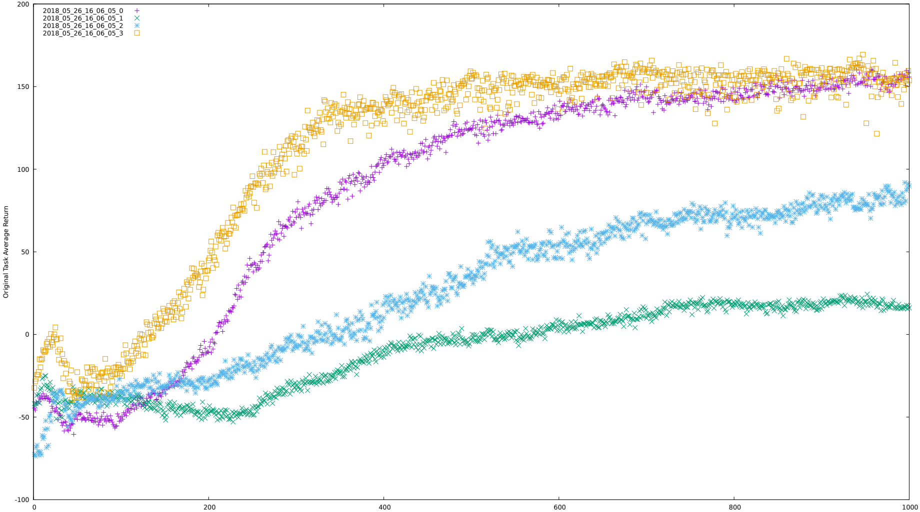
ant_data_collect



ant_state_irl



ant_transfer



Source and links

- https://sites.google.com/view/adversarial-irl
- https://github.com/mvieth/inverse_rl
- https://arxiv.org/abs/1710.11248