SQIL

: Imitation Learning via Reinforcement Learning with Sparse Rewards

Siddharth Reddy, Anca D. Dragan, Sergey Levine
University of California, Berkeley (2019)

IIE8557-01 동적계획법과 강화학습

경영과학연구실 김지원

Behavior Cloning and Inverse Reinforcement Learning

- Behavior cloning: When the agent drifts and encounters out-of-distribution states, the agent does not know how to return to the demonstrated states.

- Inverse Reinforcement Learning: training an RL agent not only to imitate demonstrated actions, but also to visit demonstrated states
- Generative Adversarial Imitation Learning: generative adversarial network를 이용해서 expert demonstrations와 agent의 trajectories를 통해 cost function을 학습

Soft Q learning

- Soft Q-learning assumes that expert behavior follows the maximum entropy model
- The expert is assumed to follow a policy π that maximizes reward

$$\pi(a|s) \triangleq \frac{\exp\left(Q(s,a)\right)}{\sum_{a' \in \mathcal{A}} \exp\left(Q(s,a')\right)},$$

$$Q(s,a) \triangleq R(s,a) + \gamma \mathbb{E}_{s'}\left[\log\left(\sum_{a' \in \mathcal{A}} \exp\left(Q(s',a')\right)\right)\right]$$

The study aims to achieve adversarial imitation methods

by a much simpler approach

that does not require adversarial training

Key Idea of Soft Q Imitaion Learning

Adversarial training 없이 (reward functio을 학습하지 않고) long horizon imitation이 가능함

1) an incentive (+1) to imitate the demonstrated actions in demonstrated states
2) an incentive (+1) to take actions that lead it back to demonstrated states when it encounters new, out-ofdistribution states

Soft Q Imitation Learning

(1) It initially fills the agent's experience replay buffer with demonstrations (Reward = +1)
(2) As the agent interacts with the world and accumulates new experiences, it adds them to the replay buffer (Reward=0)
(3) It balances the number of demonstration experiences and new experiences (50% each)

Soft Q Imitation Learning

Algorithm 1 Soft Q Imitation Learning (SQIL)

Require $\lambda_{\text{samp}} \in \mathbb{R}_{\geq 0}$
Initialize $\mathcal{D}_{\text{samp}} \leftarrow \emptyset$
while $Q_{\boldsymbol{\theta}}$ not converged **do**
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} (\delta^2(\mathcal{D}_{\text{demo}}, 1) + \lambda_{\text{samp}} \delta^2(\mathcal{D}_{\text{samp}}, 0))$ {See Equation 1}
    Sample transition (s, a, s') with imitation policy $\pi(a|s) \propto \exp(Q_{\boldsymbol{\theta}}(s, a))$
    $\mathcal{D}_{\text{samp}} \leftarrow \mathcal{D}_{\text{samp}} \cup \{(s, a, s')\}$ with an assigned reward of zero
end while

Soft Q Learning : choosing an action with weighted probabilities

Soft Bellman Error: entropy 최대화

$$\delta^2(\mathcal{D}, r) \triangleq \frac{1}{|\mathcal{D}|} \sum_{(s,a,s') \in \mathcal{D}} \left(Q_{\boldsymbol{\theta}}(s, a) - \left(r + \gamma \log \left(\sum_{a' \in \mathcal{A}} \exp\left(Q_{\boldsymbol{\theta}}(s', a')\right) \right) \right) \right)^2$$

Interpreting SQIL as Regularized Behavioral Cloning

SQIL is equivalent to a variant of behavioral cloning (BC) that uses regularization to overcome state distribution shift

BC **:** supervised learning maximizing the conditional likelihood of the demonstrated actions given the demonstrated states

Regularized BC: regularizing Qθ with a sparsity

$$\ell_{\mathrm{BC}}(\boldsymbol{\theta}) \triangleq \sum_{(s,a)\in\mathcal{D}_{\mathrm{demo}}} -\log \pi_{\boldsymbol{\theta}}(a|s).$$

$$\ell_{\mathrm{BC}}(\boldsymbol{\theta}) \triangleq \sum_{(s,a)\in\mathcal{D}_{\mathrm{demo}}} -\left(Q_{\boldsymbol{\theta}}(s,a) - \log\left(\sum_{a'\in\mathcal{A}} \exp\left(Q_{\boldsymbol{\theta}}(s,a')\right)\right)\right)$$

$$\ell_{\mathrm{RBC}}(\boldsymbol{\theta}) \triangleq \ell_{\mathrm{BC}}(\boldsymbol{\theta}) + \lambda\delta^2(\mathcal{D}_{\mathrm{demo}} \cup \mathcal{D}_{\mathrm{samp}}, 0),$$

Qθ outputs high values for actions that lead to states from which the demonstrated states are reachable

Interpreting SQIL as Regularized Behavioral Cloning

The gradient of the RBC loss is proportional to the gradient of the SQIL loss

Algorithm 1 Soft Q Imitation Learning (SQIL)

1: Require $\lambda_{\text{samp}} \in \mathbb{R}_{\geq 0}$
2: Initialize $\mathcal{D}_{\text{samp}} \leftarrow \emptyset$
3: **while** $Q_{\boldsymbol{\theta}}$ not converged **do**
4: $\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} (\delta^2(\mathcal{D}_{\text{demo}}, 1) + \lambda_{\text{samp}} \delta^2(\mathcal{D}_{\text{samp}}, 0))$ {See Equation 1}
5: $\quad$ Sample transition (s, a, s') with imitation policy $\pi(a|s) \propto \exp(Q_{\boldsymbol{\theta}}(s, a))$
6: $\quad \mathcal{D}_{\text{samp}} \leftarrow \mathcal{D}_{\text{samp}} \cup \{(s, a, s')\}$
7: **end while**

$$\ell_{\text{RBC}}(\boldsymbol{\theta}) \triangleq \ell_{\text{BC}}(\boldsymbol{\theta}) + \lambda \delta^2(\mathcal{D}_{\text{demo}} \cup \mathcal{D}_{\text{samp}}, 0),$$

SQIL solves a similar optimization problem to RBC

Image-Based Car Racing

	Domain Shift ($\mathcal{S}_0^{\text{train}}$)	No Shift ($\mathcal{S}_0^{\text{demo}}$)
Random	-21 ± 56	-68 ± 4
BC	-45 ± 18	$\mathbf{698 \pm 10}$
GAIL-DQL	-97 ± 3	-66 ± 8
SQIL (Ours)	$\mathbf{375 \pm 19}$	$\mathbf{704 \pm 6}$
Expert	480 ± 11	704 ± 79

Figure 1: Average reward on 100 episodes after training. Standard error on three random seeds.

Image-Based Experiments on Atari

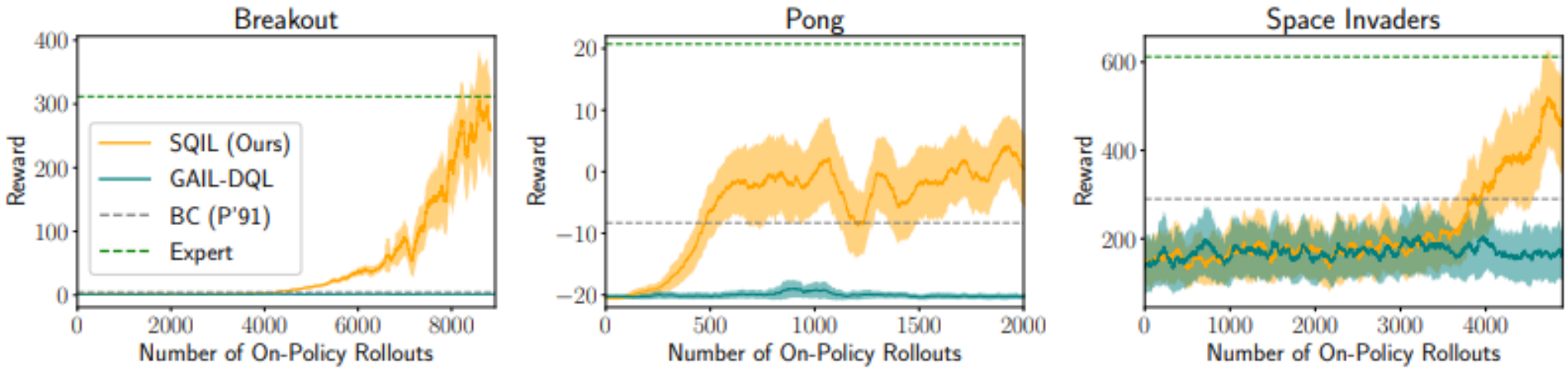

Figure 2: Image-based Atari. Smoothed with a rolling window of 100 episodes. Standard error on three random seeds. X-axis represents amount of interaction with the environment (not expert demonstrations).

Conclusions

- SQIL is derived from sparsity-regularized BC, while the prior methods are derived from an alternative formulation of the IRL objective
- SQIL is shown to outperform BC and GAIL in tasks with image observations and significant shift in the state distribution between the demonstrations and the training environment.