A Simple Imitation Learning Method via Contrastive Regularization

David S. Hippocampus*
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Abstract

Learning to imitate expert behavior from demonstrations is a challenging problem, especially in environments with high-dimensional, continuous observations and unknown dynamics. The simplest methods are behavioral cloning (BC), but they suffer from the problem of distribution shift: it can shift away from demonstrated states due to accumulated errors, since the agent greedily imitates demonstrated actions. Recent methods using reinforcement learning (RL), such as generative adversarial imitation learning (GAIL) and its variants, overcome this issue by training an RL agent to match the demonstrations over a long horizon. However, they all require a brittle adversarial training process with unstable rewards. And in order to augment RL process, some other papers build a specific generative model for the expert demonstrations, which increase the model and implementation complexity significantly. In this paper, we propose to train the policy as a classifier over states in expert dataset, and attenuate distribution shift by RL with fixed rewards. Here we calculate fixed rewards, based on an energy-based model (EBM) hidden in the policy. Moreover, we train this EBM by contrastive divergence method, further regularized by contrastive representation learning. Different from adversarial learning-based methods, we use fixed rewards obtained in a simple manner. There are no extra models needed here for distribution estimation or rewards modeling, reducing the model and implementation complexity significantly. The experiments on various Atari games show its performance improvement over many previous methods.

1 Introduction

It is an essential task to train artificial agents to perform complex tasks in many applications in robotics, video games and dialogue. If the goal on the task can be accurately described using a reward or cost function, reinforcement learning (RL) methods offer an approach to learning policies, and it has been proven to be successful in a wide range of practical applications [11, 17, 18, 19]. However, in other cases the desired behavior may only be roughly specified and it is unclear how to design a reward function to characterize it. For example, training a video game agent to adopt more human-like behavior using RL would require designing a reward function which characterizes behaviors as more or less human-like, which is difficult. Algorithms training RL agent to learn expert behaviors fall into the category of Imitation Learning (IL)[14].

Among IL methods, behavior cloning (BC) is an elegant approach whereby agents are trained to directly mimic the behaviors of an expert rather than optimizing a reward function [15, 23, 24, 28]. It

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

basically consists of training a policy to predict the expert's actions from states in the demonstration data using supervised learning, which has the simplest model and implementation complexity among IL methods. While appealingly simple, BC suffers from the problems of error accumulation and covariance shift. It is the fact that the distribution over states observed at execution time can differ from the distribution observed during training. Minor errors which initially produce small deviations can be accumulated and become amplified as the policy encounters states further and further from its training distribution. This problem was first formally tackled by [22] and a regret bound was derived with tightness proved. Further in [24] the regret can be reduced to linear bound if the policy is allowed to further interact with the environment and make queries to the expert policy. However the queries to experts are not available in general situations.

Recently methods based on adversarial learning have been proposed to tackle the covariate shift of BC [6, 7, 13, 15]. These methods train an RL agent not only to imitate demonstrated actions, but also to visit demonstrated states. Since the true rewards are unknown, a reward function is constructed from the demonstrations and visited trajectories via adversarial learning. However, the alternative training of policy and discriminator can make the learning process unstable, significantly increasing the sampling complexity [2]. Some work solve the imitation learning problem in the frameworks of Q-learning [21, 25]. However, since these methods set the reward based on the appearance of transitions in the expert demonstrations, resulting the problem of sparse reward when few demonstrations are available. Another stream of work [1, 3, 29] uses an extra model, such as random network distillation and disagreement, to estimate the support of the expert's distribution in state-action space, and minimizes an RL cost designed to guide the agent towards the states within the expert's support. But these estimation models increases the model and implementation complexity. And they may not give a good distance between states in replay buffer and those covered by expert demonstrations, especially in high-dimensional cases, which may mislead the agent to wrong states far from expert's support.

In this work, we propose a simple imitation learning method by incorporating the idea of contrastive learning. Recently the work [9] has reinterpreted a standard discriminative classifier as an energy-based model for the joint distribution of sample and labels, achieving both successful generative and discriminative learning in the same model. Inspired by that, in addition to the supervised learning of expert's actions, we use the generative model hidden within the policy network as the reward generator, guiding the agent back to the support of expert's demonstrations in state space, alleviating the covariate shift problem. This generative model is trained by contrastive divergence method [12], by maximizing the unnormalized density between states in the expert's support and those in replay buffer. We use an RL process, which maximizes the rewards given by this generative model, guiding the agent to the states covered by expert demonstrations. In order to further reduce the sample complexity of RL process, we extract high-level features by learning contrastive representations in comparison of states covered by expert's demonstrations and those just visited by the learning policy. Different from previous contrastive learning methods [4, 27], we modify the learning objective, compatible with the supervised learning of expert's actions.

Compared with previous work, our method has multiple merits. First, since supervised learning for cloning expert's actions and contrastive learning for generating rewards are conducted in one policy model, it keeps the simplicity of behavior cloning, without adding extra models for support or distribution estimation. Second, by avoiding the adversarial learning, the training process can be stable. Third, as far as we know, it is first to introduce the contrastive representation learning into imitation learning, which can reduce the sample complexity significantly. The empirical experiments on difficult imitation learning tasks, such as image-based Atari games, show the significant improvement of our method compared with previous work.

2 Methodology

Our method is motivated from the perspective of divergence minimization over imitation learning [8]. We decompose the occupancy measure divergence into two parts, i.e.,

$$D_f(\rho^{\pi_E}(s,a)\|\rho^{\pi_\theta}(s,a)) = D_f(\pi_E(s,a)\|\pi_\theta(a|s)) + D_f(\rho^{\pi_E}(s)\|\rho^{\pi_\theta}(s)) \tag{1}$$

Minimizing the first term in (1) is same as that the learning policy mimics expert's actions conditioned on the states, in the support of expert's demonstrations. But in order to minimize the second term of (1), the RL agent should be able to realize and return when it's outside of the support of states in expert's demonstrations.

Specifically, the first part is addressed by the standard behavior cloning [16, 28], i.e., a supervised learning on expert's actions. Regarding the second part, we use an RL process to maximize rewards in proportional to the similarity of current states and states of expert's behavior. Without using extra models, we utilize a generative model hidden in the policy model, trained by contrastive divergence method. In order to augment the sample efficiency, contrastive learning is incorporated into imitation learning here, specially designed to boost the behavior cloning and reward learning at the same time. Our method has minimal changes on the architecture and learning pipeline compared with classical ones. It is simple to implement and train.

Here we denote $\pi_\theta(\cdot|s)$ and $\pi_E(\cdot|s)$ as learning policy and expert policy respectively. Denote $\mathcal{D}_E = \{(s_i, a_i)\}$ as the dataset containing state-action pairs along expert's demonstrations. Let d_π denote the distribution over states induced by following π. For discrete action environment, we use N_A to denote the number of possible actions. In contrastive learning, we denote $\mathcal{P}$ as the set of positive samples and $\mathcal{N}$ as the set of negative samples.

We first define the behavior cloning (BC) loss. Following the classical work [22] the supervised behavior cloning loss J_{BC} is integrated over the state distribution induced by expert policy, i.e.,

$$J_{\text{BC}}(\theta) = \mathbb{E}_{s\sim d_{\pi_E}}[\|\pi_E(\cdot|s) - \pi_\theta(\cdot|s)\|] \tag{2}$$

For environments with discrete actions, the BC objective becomes cross-entropy loss same as that in classification problems.

2.1 Reward Learning

In order to guide the agent return to the states covered by expert's behavior, we have to establish a reward signal for notification when the agent is away from the support of expert's states. In this paper, we propose a simple method for measuring the distance of states and use it as reward signal. Here we utilize a generative model hidden inside the policy network. More specifically, in discrete action cases, the policy network gives, for $\forall a \in \mathcal{A}$,

$$\pi_\theta(a|s) = \frac{\exp(f_\theta(s)[a])}{\sum_{a'\in\mathcal{A}} \exp(f_\theta(s)[a'])} \tag{3}$$

where $f_\theta(s) : \mathcal{S} \to \mathbb{R}^{N_A}$ is the mapping from state space to the logits, i.e., $f_\theta(s)[a]$ is the logit corresponding to label a. What's more, we utilize these logits to define an energy based model of the joint distribution of state-action pairs, and an unnormalized density model of states, i.e.,:

$$p_\theta(s, a) = \frac{\exp(f_\theta(s)[a])}{Z(\theta)}, \qquad p_\theta(s) = \frac{\sum_{a'\in\mathcal{A}} \exp(f_\theta(s)[a'])}{Z(\theta)} \tag{4}$$

Then we can define an energy function at state s as below

$$E_\theta(s) = -\log \sum_{a'\in\mathcal{A}} \exp(f_\theta(s)[a']) \tag{5}$$

The negation of this energy function is used as reward during the RL process, i.e., $r(s_t, a_t) = -E_\theta(s_{t+1})$, where s_{t+1} is the next state by applying action a_t. And it's trained in a loss function inspired by contrastive learning.

$$J_{\text{R}}(\theta) = \frac{\sum_{s\in\mathcal{P}} \exp(-E_\theta(s))}{\sum_{s\in\mathcal{P}} \exp(-E_\theta(s)) + \sum_{s'\in\mathcal{N}} \exp(-E_\theta(s'))} \tag{6}$$

where samples in $\mathcal{P}$ are states sampled from expert dataset, while samples in $\mathcal{N}$ are states sampled from trajectories visited by learning policy.

2.2 Contrastive Regularization

In order to further alleviate covariate shift and reduce sample complexity, we introduce a contrastive objective as another auxiliary loss, with minimal changes to the RL algorithm and architectures. Here we first define an objective compatible with both behavior cloning and contrastive learning, especially in discrete-action environments. Without building another pair of encoders [27], we produce queries and keys of states directly at the second last layer of learning and target policy networks, where the

target policy network is updated in a soft way. And every observed state is augmented by random cropping, to increase the generalizability of learned representations. Different from previous work [5, 10, 20], there is no need to have a separate model for learning context representation.

Denoting the number of state-action pairs having action a in the expert minibatch $\mathcal{P}$ as $N_a^E, \forall a \in \mathcal{A}$. Denote the latent (second last) layer of policy network as $g_\theta(s)$, i.e., $f_\theta(s) = \boldsymbol{W}_\pi g_\theta(s)$, where $\boldsymbol{W}_\pi \in \mathbb{R}^{N_A \times D}$ and D is the dimension of latent representation.

For every observed state s, it is first augmented by random cropping process, denoted as $\text{rc}(s)$. Then query q and key z are generated by the latent layer of learning policy π_θ and target policy $\pi_{\theta^{\text{target}}}$, i.e., $q(s) = \text{rc}(g_\theta(s))$ and $z(s) = \text{rc}(g_{\theta^{\text{target}}}(s))$. Here we adopt bilinear product [20] as the similarity between query and key pairs of each state, i.e., $q(s)\boldsymbol{W}_c z(s)$, where $\boldsymbol{W}_c \in \mathbb{R}^{D \times D}$. Then the contrastive object used in discrete-action environments is as below.

$$J_{\text{C}}(\theta) = \sum_{a \in \mathcal{A}} \frac{\frac{1}{N_a^E} \sum_{(s',a') \in \mathcal{P}, a'=a} q(s') \boldsymbol{W}_c z(s')}{\sum_{(s',a') \in \mathcal{P}} q(s') \boldsymbol{W}_c z(s') + \sum_{(s'',a'') \in \mathcal{N}} q(s'') \boldsymbol{W}_c z(s'')} \tag{7}$$

Therefore, the total loss for updating the policy network is

$$J_{\text{BC}}(\theta) + \alpha J_{\text{R}}(\theta) + \beta J_{\text{C}}(\theta, \boldsymbol{W}_c) \tag{8}$$

where α and β are tuned empirically. And parameters of policy network is update as $\theta := (1-\gamma)\theta + \gamma\theta^{\text{target}}$. Both θ and $\boldsymbol{W}_c$ are updated by stochastic gradient descent separately in experiments. The overall algorithm is summarized as below.

Algorithm 1: Imitation Learning via Contrastive Regularization

Input : Expert demonstration data $\mathcal{D}_E = \{(s_i, a_i)\}_{i=1}^N$

Initialize the policy π_θ and replay buffer $\mathcal{B}$

Pre-training π_θ by behavior cloning on $\mathcal{D}_E$

for $e = 1, \ldots,$ **do**

  Sample trajectories by playing policy π_θ and store them into replay buffer $\mathcal{B}$.

  Sample minibatch $\mathcal{P}$ and $\mathcal{N}$ from $\mathcal{D}_E$ and $\mathcal{B}$ respectively.

  Conduct SGD with objective (8) to optimize θ and $\boldsymbol{W}_c$ separately.

  Conduct one step of policy gradient on π_θ, with negative energy function of next state (5) as rewards.

end

3 Experiments

We evaluated the proposed IL method on many Atari environments. The expert policy is trained by PPO [26] and generate a butch of expert trajectories stored in $\mathcal{D}_E$. In order to stabilize the training process, the reward is clipped into -1 or 1 based on the threshold, set by the $\beta-$quantile of rewards over all the states in expert's demonstrations [1].

The baselines for comparison are standard behavior cloning (BC) [22] and generative adversarial imitation learning (GAIL) [13]. We find that the propose method can outperform both BC and GAIL in all of the evaluated environments. It is already known that GAIL cannot perform well on image-based environments [21], and our method has significant improvement over that.

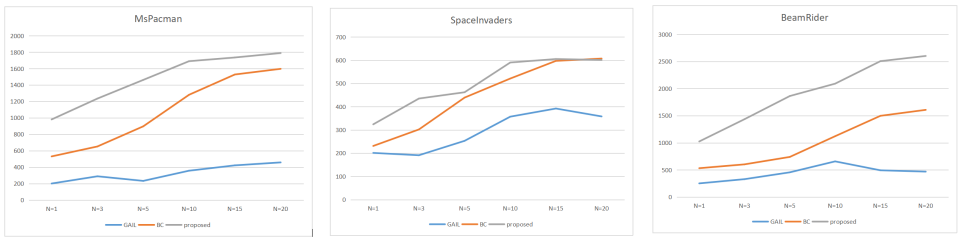

Figure 1: Experiments on Atari games. Average reward vs number of expert trajectories.

References

[1] Kiante Brantley, Wen Sun, and Mikael Henaff. Disagreement-regularized imitation learning. In *International Conference on Learning Representations*, 2020.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[3] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Debidatta Dwibedi, Jonathan Tompson, Corey Lynch, and Pierre Sermanet. Learning actionable representations from visual observations. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1577–1584. IEEE, 2018.

[6] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pages 49–58, 2016.

[7] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[8] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. *arXiv preprint arXiv:1911.02256*, 2019.

[9] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

[10] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[11] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[12] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[13] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[14] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[15] Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

[16] Sergey Levine and Vladlen Koltun. Guided policy search. In *International Conference on Machine Learning*, pages 1–9, 2013.

[17] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[18] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[21] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: imitation learning via regularized behavioral cloning. *arXiv preprint arXiv:1905.11108*, 2019.

[22] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[23] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[24] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[25] Fumihiro Sasaki, Tetsuya Yohira, and Atsuo Kawaguchi. Sample efficient imitation learning for continuous control. In *International Conference on Learning Representations*, 2019.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[27] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[28] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018.

[29] Ruohan Wang, Carlo Ciliberto, Pierluigi Vito Amadori, and Yiannis Demiris. Random expert distillation: Imitation learning via expert policy support estimation. In *International Conference on Machine Learning*, pages 6536–6544, 2019.