Vol.2019-CVIM-217 No.11
2019/5/30

模倣学習を用いた動画からの動作獲得

大里 虹平[1,a)]　川本 一彦[2]

概要：本研究では，カメラを用いた模倣学習について検討する．模倣学習では，人間の行動系列から最適な方策を獲得し，ロボット制御へ適用する．このとき，獲得した方策をある特定のロボット制御に適した力やトルク等で表現することになるが，その方策を別のロボットへを移植するときには手間がかかる．そこで，特定のロボットに依存しない方策の中間表現の獲得のために空間情報のみを用いる模倣学習を導入する．OpenAI Gym と Mujoco を用いた仮想環境で，いくつかの模倣学習法を評価し，提案手法について検証している．

1. はじめに

近年，ロボット技術の発展に伴い，従来人間が行ってきた作業の多くがロボットによる作業に置き換えられている．しかし，その場面は多岐にわたるうえ，人間の複雑な動作を古典的なプログラミング手法で設計するには非常に時間がかかる．模倣学習では，予め示された手本の振る舞いを模倣するように学習することによって，この問題を解決する．

通常，エキスパートの動作を観測するときやロボットの制御時にセンサーを直接つける必要がある．また，模倣学習は多くの場合，特定のロボットをシミュレートした環境で行う．このとき，獲得した方策の行動は力やトルク等で表現されるため，獲得した方策を別のロボットに移植するときに手間がかかる．

本研究では非接触のセンサーとして動画を用いる．そして，空間情報のみを用いて状態・行動を定義し，模倣学習手法を適用する．このとき獲得した方策の表現には特定のロボットの力やトルクを用いない．これによって，

- 多くのセンサーを直接つける必要がない
- 学習データを容易に準備できる
- 獲得した方策が特定のロボットに依存しない

といったメリットを享受することができる．

[1] 千葉大学 大学院融合理工学府
Graduate School of Science and Engineering, Chiba University, 1–33, Yayoicho, Inage-ku Chiba-shi, Chiba, Chiba 263–8522, Japan
[2] 千葉大学 大学院工学研究院
Graduate School of Engineering, Chiba University, 1–33, Yayoicho, Inage-ku Chiba-shi, Chiba, Chiba 263–8522, Japan
[a)] rainbow.o@chiba-u.jp

2. 模倣学習

模倣学習の目的はエキスパートの行動系列から最適な方策を抽出することである．代表的な 2 つ手法として，教師あり学習手法である behavioral cloning[1] と逆強化学習 [2], [3] を用いてコスト関数を求めた後に強化学習を適用する手法がある．

2.1 behavioral cloning

behavioral cloning[1] はエキスパート行動系列の状態-行動ペアを利用した教師あり学習手法である．この手法は単純であるが，大量のデータを必要とすることが知られている [4], [5].

2.2 逆強化学習を利用した模倣学習

強化学習では，予め設計された報酬関数をもとに収益を最大化するように方策関数を学習する．しかし，問題によっては報酬関数の設計が困難な場合がある．たとえば，迷路において，ゴールに到達した状態にのみ報酬を付与しても，そこに至る行動系列のなかでどの行動が報酬の獲得に寄与したかが分からないため，学習は困難になる．

この報酬設計問題の解決法として逆強化学習 [2], [3] は有効である．逆強化学習では，エキスパートの行動系列から，エキスパート方策のコスト（負の報酬）が他の方策のコストより必ず低くなるようなコスト関数を推定する．推定した報酬関数に対して強化学習を適用することによって，手作業で報酬関数を設計することなく最適方策を得ることができる．しかし，多くの逆強化学習手法は内部で強化学習を用いるうえ，推定した報酬関数はエージェントがどのように動くべきかを直接示すものではない．そのため，模倣

学習に逆強化学習を用いる手法では，非常に多くの計算が必要となる問題がある．

2.3 GAIL

Generative Adversarial Imitation Learning（GAIL）[6] は逆強化学習を用いた手法が持つ問題を解決し，より効率的に模倣を行えるようにした．

逆強化学習を用いた手法は最適な方策を求めるために 2 段階の最適化問題を解く必要があった．[6] では正則化関数の導入と変数変換によってこれを 1 段階の最適化問題に定式化できることを示した．GAIL では，この 1 段階最適化問題に対して，Generative Adversarial Networks[7] の目的関数を正則化関数として導入することで模倣学習を行う．すなわち，以下の最適化問題として定式化する．

$$\min_{\pi\in\Pi}\max_{D\in(0,1)^{\mathcal{S}\times\mathcal{A}}} \mathbb{E}_{\pi}[\log(D(s,a))] + \mathbb{E}_{\pi_E}[\log(1-D(s,a))] - \lambda H(\pi) \tag{1}$$

ここで，Π は状態空間 $\mathcal{S}$ と行動空間 $\mathcal{A}$ によって定義される方策関数族，s は状態，a は行動，π は方策，π_E はエキスパート方策である．$D(s,a)$ は状態-行動のペアがエキスパート方策由来でない確率を表し，$\log(D(s,a))$ はコスト関数の代わりとなる．

GAIL は識別器と生成器の 2 つのネットワークから構成される．識別器は入力の状態-行動ペアがエキスパート方策由来であるか，生成器が生成したものであるかを識別する．一方，生成器は識別器の識別が困難になるような方策を生成する．GAIL では生成された振る舞いに対して，識別器が直接フィードバックを与えるため，効率的に学習することができる．

GAIL では生成器が表現する方策からサンプリングしながら，識別器と生成器を交互に更新することで学習する．パラメータ w の識別器ネットワークは以下の勾配に対して Adam[8] を用いた勾配降下法を適用して更新する．

$$\hat{\mathbb{E}}_{\tau_i\sim\pi_{\theta_i}}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E\sim\pi_E}[\nabla_w \log(1-D_w(s,a))] \tag{2}$$

パラメータ θ の生成器ネットワークは以下の勾配に対して Trust Region Policy Optimization[9] を用いた自然方策勾配法 [10] を適用して更新する．

$$\hat{\mathbb{E}}_{\tau_i\sim\pi_{\theta_i}}[\nabla_\theta \log\pi_\theta(a|s)Q(s,a)] - \lambda\nabla_\theta H(\pi_\theta), \quad \text{where} \quad Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a))|s_0=\bar{s},a_0=\bar{a}] \tag{3}$$

3. 提案手法

現実世界でロボットを動かしながらの学習は，電力等のコスト面や学習時間の面から現実的ではない．そこで，ロボットを物理シミュレーションソフト上で動かしながら学習する方法が取られる．しかし，この手法では特定のロボットを仮定した上で，行動を力やトルクで表現する．そのため，（形状の同じ）別のロボットへの方策の移植が困難になる．

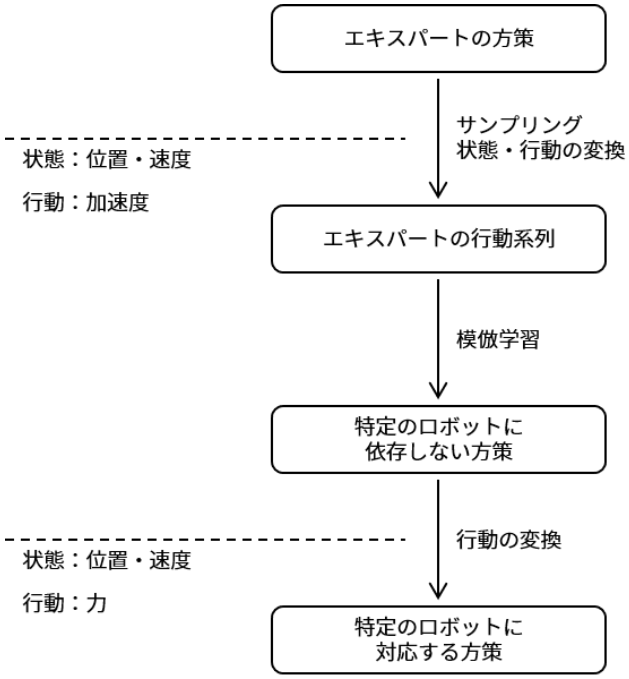

図 1 提案手法の流れ

本研究の目的は，状態と行動を特定のロボットに依存しないように定義することで，学習の一部を共通化することである．

まず，特定のロボットに依存しない表現として空間情報を用いて状態・行動を定義をする．このとき，空間情報の取得には動画を用いる．次に，定義した状態・行動のもとで模倣学習手法を適用する．この時に獲得した方策は特定のロボットに依存しない．最後に，獲得した方策の行動を加速度や角加速度から力や特定のロボットに対応したトルクに変換する．全体の流れを**図 1** に示す．

これから，動画から状態・行動を変換する手法を説明する．3 次元空間上を動くタスクでは，3 次元復元が必要であるため，複数視点のカメラ動画が必要になる．また，2 次元空間上を動くタスクでは，単視点動画を平面の法線方向からの視点からの動画に変換する．

状態は，関節の位置（角度）・速度（角速度）によって定義する．時刻 t におけるエージェントの状態 $\boldsymbol{s}_t$ を 1 フレーム前との差分を用いての以下のように計算する．

$$\boldsymbol{s}_t = \begin{bmatrix} \boldsymbol{x}_t \\ \dot{\boldsymbol{x}}_t \end{bmatrix}, \quad \dot{\boldsymbol{x}}_t = (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) \times n \tag{4}$$

ここで，$\boldsymbol{x}$ は位置，$\boldsymbol{x}$ は速度，n は動画の 1 秒間あたりのフレーム数である．

行動は，関節の加速度（角加速度）によって定義する．

加速度を用いる理由は外部から力が働かなければロボットの力に比例するためである．時刻 t における行動 $\boldsymbol{a}_t$ を 1 フレーム先との差分を用いての以下のように計算する．

$$\boldsymbol{a}_t = \ddot{\boldsymbol{x}}_t = (\dot{\boldsymbol{x}}_{t+1} - \dot{\boldsymbol{x}}_t) \times n \tag{5}$$

ここで，$\ddot{\boldsymbol{x}}$ は加速度である．

模倣学習手法によって獲得した方策 $\pi(\boldsymbol{a}|\boldsymbol{s})$ の行動 $\boldsymbol{a}$ は，加速度や角加速度によって表現されているが，実際に特定のロボットに適用するためにはこの行動を力やトルクによって表現される行動 $\hat{\boldsymbol{a}}$ に変換する必要がある．パラメータ $\boldsymbol{\xi}$ のモデル $\hat{\boldsymbol{a}} = f(\boldsymbol{a};\boldsymbol{\xi})$ によって行動を変換し，最終的な方策 $\hat{\pi}(\hat{\boldsymbol{a}}|\boldsymbol{s})$ を得る．

変換モデルのパラメータ $\boldsymbol{\xi}$ は，実際にロボットからサンプリングした加速度を用いた行動 $\boldsymbol{a}_i^{sample}$ と力を用いた行動 $\hat{\boldsymbol{a}}_i^{sample}$ のペアから以下の式を最小化することで求める．

$$\mathbb{E}[\|\hat{\boldsymbol{a}}^{sample} - f(\boldsymbol{a}^{sample};\boldsymbol{\xi})\|_2^2] \tag{6}$$

ただし，$\mathbb{E}[\cdot]$ は期待値，$\|\cdot\|_2$ は L_2 ノルムを表す．

実際のロボットの制御は撮影しながら行い，式 (4) によって得られた状態 $\boldsymbol{s}$ に対して獲得した方策 $\hat{\pi}$ を適用する．

4. 実験

実験は OpenAI Gym[11] の Reacher-v2 環境で行い，提案手法によって獲得した方策，エキスパート方策，ランダム方策でのスコアを比較した．

4.1 実験手順

この実験は，実世界環境の代わりに物理シミュレーションソフト MuJoCo[12] 上の環境を用いる．Reacher-v2 環境は MuJoCo 上で定義され，重力と垂直な平面状を動く連結アームの先端をランダムな位置に現れるターゲット位置へ近づけることを目標とするタスクである（**図 2**）．OpenAI Gym で定義されている報酬はアーム先端とターゲット間の距離とトルクの大きさから決まる．1 エピソードは 50 タイムステップで，エピソード収益は 1 エピソードの各タイムステップの報酬の和である．

まず，この報酬関数のもとで最適な方策を持つエキスパートを強化学習手法 Proximal Policy Optimization（PPO）[13] を用いて生成した．次に，このエキスパート方策にしたがって動作するエージェントの行動系列をいくつか動画としてサンプリングし，状態・行動を空間情報を用いたものに変換した．その後，状態-行動ペアに対して模倣学習手法を適用し方策を獲得した．この方策を式 (7) の線形回帰を用いてトルクに変換し，評価した．

$$\hat{a}_i = \alpha_i a_i + \beta_i, \quad (i = 0, 1) \tag{7}$$

ここで，$\hat{a}_i$ と a_i はそれぞれ関節 i のトルクと加速度であり，α_i と β_i はそれぞれ係数と切片である．

表 1 エキスパート方策とランダム方策での平均エピソード収益（$n = 50$）

	収益
エキスパート方策	-4.37
ランダム方策	-42.51

本手法では模倣学習手法を指定しないため，実験では behavioral cloning と GAIL の 2 種類を用いた．多くのエキスパートサンプルを必要とする behavioral cloning とそうでない GAIL での結果を比較するために，3，10，32，100 の 4 種類のエキスパートサンプル数を用いた．方策の評価には 50 エピソードの平均エピソード収益を用い，PPO によって生成したエキスパート方策，毎フレームランダムな行動をする方策と比較した．エピソード収益はエキスパート方策で最も高くなる．

4.2 学習条件

方策関数はニューラルネットワークを用いて表現する．ネットワークは 2 層の隠れ層をもち，各隠れ層は 100 個のユニットから構成される．活性化関数には tanh を用いた．

4.2.1 behavioral cloning

状態-行動ペアの 70%を訓練データ，30%を検証データとして利用した．ミニバッチサイズ 128 で Adam を用いて検証誤差が下がらなくなるまで重みを更新した．

4.2.2 GAIL

識別器のネットワークは 2 層の隠れ層をもち，各層は 100 個のユニットから構成される．活性化関数には tanh を用いた．イテレーション数は 2000 で，各イテレーションで生成器を 5 回，識別器を 1 回更新した．1 回の生成器更新で 50000 個の状態-行動ペアをサンプリングした．

4.3 実験結果

表 1 はエキスパート方策とランダム方策にそれぞれ従うエージェントの平均エピソード収益である．**表 2** は提案手法で獲得した方策にそれぞれ従うエージェントの平均エピソード収益である．達成度はエキスパート方策を 100%，ランダム方策を 0%となるようにスケールを調節した．

いずれの手法，サンプル数でも達成度は 80%以上に達した．したがって，獲得した方策はエキスパート方策に近いものになった．エキスパートサンプル数が少ない場合には，behavioral cloning より GAIL の方が高い達成度となった．

5. おわりに

本研究では，状態・行動を骨格の空間情報のみを用いて定義し，模倣学習手法を適用することによって，動画から特定のロボットに依存しない方策を獲得する手法を提案した．実験ではいくつかの模倣学習手法とエキスパートサン

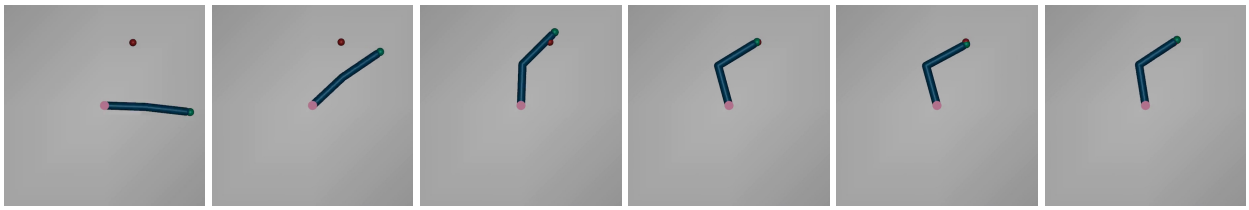

図 2 Reacher-v2 環境．連結アームの先端（緑）をランダムな位置に現れるターゲット（赤）に近づけるタスク．

表 2 提案手法での平均エピソード収益（$n = 50$）

模倣学習手法	サンプル数	収益	達成度（%）
behavioral cloning	3	-9.81	85.7
	10	-8.85	88.2
	32	-6.00	95.7
	100	-5.24	97.7
GAIL	3	-7.84	90.9
	10	-6.05	95.6
	32	-5.53	97.0
	100	-5.61	96.7

プル数で評価した．

今後の課題として，形が異なるロボットへの方策の移植やロボットとオブジェクトのインタラクションがあるタスクへの応用，模倣学習手法によって得られた方策を高効率・高精度に特定のロボットに対応した方策へ変換する手法の検討がある．これらの課題は転移学習との組み合わせによって解決されると考えている．また，3 次元空間上を動くタスクにおいても，深層学習を用いた 3 次元姿勢推定 [14], [15] と組み合わせることで単視点動画を利用できると考えている．

さらに，空間情報のみを用いた本手法は，CG 映像やゲームといった分野への応用も期待される．

謝辞 本研究は JSPS 科研費 JP16K00231，JP19K12039 の助成を受けたものです．

参考文献

[1] Pomerleau, D.: Efficient Training of Artificial Neural Networks for Autonomous Navigation, *Neural Computation*, Vol. 3, No. 1, pp. 88–97 (1991).
[2] Russell, S. J.: Learning Agents for Uncertain Environments, *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, Vol. 98, pp. 101–103 (1998).
[3] Ng, A. Y. and Russell, . S. J.: Algorithms for Inverse Reinforcement Learning, *ICML*, pp. 663–670 (2000).
[4] Ross, S. and Bagnell, . J. A.: Efficient Reductions for Imitation Learning, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 661–668 (2010).
[5] Ross, S., Gordon, . G. J. and Bagnell, . J. A.: A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635 (2011).
[6] Ho, J. and Ermon, S.: Generative Adversarial Imitation Learning, *NeurIPS*, pp. 4565–4573 (2016).
[7] Goodfellow, I. J., Pouget-Abadie, . J., Mirza, . M., Xu, . B., Warde-Farley, . D., Ozair, . S., Courville, . A. C. and Bengio, . Y.: Generative Adversarial Nets, *NeurIPS*, pp. 2672–2680 (2014).
[8] Kingma, D. P. and Ba, . J.: Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980* (2014).
[9] Schulman, J., Levine, . S., Moritz, . P., Jordan, . M. I. and Abbeel, . P.: Trust Region Policy Optimization, *ICML*, pp. 1889–1897 (2015).
[10] Ho, J., Gupta, . J. K. and Ermon, . S.: Model-Free Imitation Learning with Policy Optimization, *ICML*, pp. 2760–2769 (2016).
[11] Brockman, G., Cheung, . V., Pettersson, . L., Schneider, . J., Schulman, . J., Tang, . J. and Zaremba, . W.: OpenAI Gym, *arXiv preprint arXiv:1606.01540* (2016).
[12] Todorov, E., Erez, . T. and Tassa, . Y.: MuJoCo: A physics engine for model-based control, *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033 (2012).
[13] Schulman, J., Wolski, . F., Dhariwal, . P., Radford, . A. and Klimov, . O.: Proximal Policy Optimization Algorithms, *arXiv preprint arXiv:1707.06347* (2017).
[14] Vondrak, M., Sigal, . L., Hodgins, . J. K. and Jenkins, . O. C.: Video-based 3D motion capture through biped control, *ACM Transactions On Graphics (TOG)*, Vol. 31, No. 4, p. 27 (2012).
[15] Kanazawa, A., Black, . M. J., Jacobs, . D. W. and Malik, J.: End-to-End Recovery of Human Shape and Pose, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7122–7131 (2018).