The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

軌道のスコアに基づく逆強化学習を用いた非線形な報酬関数の推定

Estimation of a non-linear reward function using score-based inverse reinforcement learning

渡邉　夏美 *1　　増山　岳人 *2　　梅田　和昇 *1
Natsumi Watanabe　Gakuto Masuyama　Kazunori Umeda

*1中央大学　　*2名城大学
Chuo University　Meijo University

This paper presents a novel inverse reinforcement learning method, which estimates a reward function from arbitrary trajectories and their scores. Whereas standard inverse reinforcement learning methods query (near-) optimal demonstrations to an expert and estimate a linear reward function, our method 1) queries scores of arbitrary trajectories, and 2) estimates a non-linear reward function. Our method inherits a benefit of score-based inverse reinforcement learning enabling the expert to free him/herself from a burden to generate demonstration trajectories. The method also can estimate a non-linear reward function by introducing kernel function to score-based inverse reinforcement learning. We tested our method by a control task of a simulated robotic manipulator. The results demonstrate that our method estimates a non-linear reward function, from which reinforcement learner generate trajectories with a high score.

1. 緒言

強化学習とは，未知環境における試行錯誤的な行動学習の枠組みであり，ロボットによる自律的な制御則の学習などに利用される [1]．強化学習における学習者であるエージェントは，選択した行動に対する評価値である報酬を観測することで，行動の選択基準となる方策を学習する．設計者は報酬によってタスクを表現するため，エージェントが獲得する方策の性能は報酬によって変化する．しかし，多様で複雑な環境における報酬の設定は人手では困難であり，報酬の設定が適切でない場合エージェントによるタスクの達成は不可能である．

逆強化学習 [2, 3, 4, 5] では，目的のタスクを表現する真の報酬関数をエキスパートの演示から推定する．エキスパートは目的のタスクにおける熟練者であり，真の報酬関数に対する最適方策に従って振る舞うと仮定される．推定した報酬関数を用いて強化学習を行うことで，人手により報酬を設定することなく方策を学習することができる．しかし，適切な報酬関数の推定には，演示が目的のタスクにおいて最適であることがしばしば要求され，エキスパートにとって大きな負担となり得る．

また，一般的な逆強化学習では，特徴量の線形結合で表現される報酬関数を推定する．線形な報酬関数は特徴量の設定に強く依存するため，設計者はタスクの表現に有効な特徴量を設定する必要がある．また，特徴量に関する報酬の線形性により，複雑なタスクに対して表現能力が不足する場合がある．そこで，報酬関数の表現能力の向上を目的とし，推定する報酬関数を非線形化した逆強化学習が提案されている [6]．

本稿では，エキスパートによる最適な演示を不要とし，高い表現能力を有する報酬関数を推定可能な逆強化学習を提案する．提案手法では，軌道のスコアに基づく逆強化学習 [7] を拡張し，非線形化した報酬関数を推定する．軌道のスコアに基づく逆強化学習では，エキスパートに対し任意の軌道をクエリとして提示し，エキスパートが与える軌道のスコアから真の報酬関数を推定する．そのため，一般的な逆強化学習と異なりエキスパートによる演示が不要である．提案手法では，軌道のスコアに基づく逆強化学習と同様に，任意の軌道のスコアから回帰により報酬関数を推定する．また，カーネル関数を用いて報酬関数を非線形化することで，タスクに対する表現能力を向上する．

連絡先: 渡邉夏美，中央大学大学院理工学研究科精密工学専攻，〒 112-8551 東京都文京区春日 1-13-27，n.watanabe@sensor.mech.chuo-u.ac.jp

2. 軌道のスコアに基づく逆強化学習

Burchfiel らによる軌道のスコアに基づく逆強化学習（Distance Minimization Inverse Reinforcement Learning; DM-IRL）[7] では，任意の軌道とそれらに対してエキスパートが付与するスコアから，目的のタスクにおける真の報酬関数を推定する．スコアは，軌道が目的のタスクにおいてどの程度望ましいかを表す評価値であり，エキスパートによって与えられる．ここで，エキスパートによるスコアは真の報酬関数に従って決定されると仮定する．DM-IRL では，エキスパートに課される負担は軌道に対するスコア付与のみであり，最適な演示は不要である．

エージェントがとり得る状態の集合を S，選択し得る行動の集合を A とする．方策 $\pi : S \times A \mapsto [0, 1]$ をエージェントが状態 $s \in S$ において行動 $a \in A$ を選択する確率，$\boldsymbol{\phi} : S \times A \mapsto \mathbb{R}^k$ を状態 s において行動 a を選択したときの特徴量とする．また，DM-IRL における真の報酬関数 $R : S \times A \mapsto \mathbb{R}$ は，式 (1) に示すモデルで表され，特徴量に関して線形である．

$$R(s, a) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(s, a) \tag{1}$$

したがって，ここでの報酬関数の推定は重み $\boldsymbol{w} \in \mathbb{R}^k$ の推定と等価である．

軌道は状態行動対の列であり，i 本目の軌道 $\tau_i = \{(s^i_0, a^i_0), \ldots, (s^i_{|\tau_i|-1}, a^i_{|\tau_i|-1})\}$ のスコア v_i は，軌道を通した報酬の累積和として式 (2) で定義される．

$$v_i = \sum_{t=0}^{|\tau_i|-1} \gamma^t R(s^i_t, a^i_t) \tag{2}$$

ただし，$\gamma \in [0, 1)$ は割引率である．式（1）を代入すると，v_i

は式（3）に書き換えられる．

$$v_i = \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(s_t^i, a_t^i) = \boldsymbol{w}^{\mathrm{T}} \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{\phi}(s_t^i, a_t^i) \quad (3)$$

r 本の軌道とそれらに対しエキスパートが付与したスコア $(\tau_1, v_1^*), \ldots, (\tau_r, v_r^*)$ が与えられたとき，式 (4) の線形回帰により重み $\boldsymbol{w}$ が推定される．

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|M\boldsymbol{w} - \boldsymbol{v}^*\| \quad (4)$$

ただし，$\boldsymbol{v}^* \in \mathbb{R}^r$ は軌道のスコア $v_1^*, \ldots, v_r^*$ を要素とするベクトル，$M \in \mathbb{R}^{r \times k}$ は式（5）の行列である．

$$M = \begin{pmatrix} \sum_{t=0}^{|\tau_1|-1} \gamma^t \boldsymbol{\phi}(s_t^1, a_t^1)^{\mathrm{T}} \\ \vdots \\ \sum_{t=0}^{|\tau_r|-1} \gamma^t \boldsymbol{\phi}(s_t^r, a_t^r)^{\mathrm{T}} \end{pmatrix} \quad (5)$$

本稿で提案する逆強化学習では，DM-IRL と同様に，任意の軌道とそれらに対しエキスパートが付与するスコアから，回帰により報酬関数を推定する．ただし，DM-IRL では特徴量に関し線形な報酬関数を推定するのに対し，提案手法では非線形化した報酬関数を推定する．式 (1) の報酬関数をカーネル関数を用いて非線形化することで，タスクに対する表現能力の向上を図る．

3. 非線形な報酬関数の推定

状態 s において行動 a を選択したときの観測値を要素とするベクトルを $\boldsymbol{x} : S \times A \mapsto \mathbb{R}^l$ とする．本稿では，カーネル関数 $k(\boldsymbol{x}, \boldsymbol{x}^{(i)})$ を用いて，報酬関数を式 (6) で定義する．

$$R(\boldsymbol{x}) = \sum_{i=1}^{n} a_i k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x}) \quad (6)$$

ここで，$\boldsymbol{x}^{(i)}$ は学習に用いる軌道 $\tau_1, \ldots, \tau_r$ に現れる n 個の観測値ベクトルである．また，$\boldsymbol{k}(\boldsymbol{x}) \in \mathbb{R}^n$ は $k(\boldsymbol{x}, \boldsymbol{x}^{(1)}), \ldots, k(\boldsymbol{x}, \boldsymbol{x}^{(n)})$ を要素とするベクトルである．カーネル関数を用いることで，特徴量を明示的に扱うことなく，特徴量に関し非線形な報酬関数を定義する．

ここでは，状態行動対の列 $\{(s_0^i, a_0^i), \ldots, (s_{|\tau_i|-1}^i, a_{|\tau_i|-1}^i)\}$ に対応する観測値の時系列を軌道 $\tau_i = \{\boldsymbol{x}_0^i, \ldots, \boldsymbol{x}_{|\tau_i|-1}^i\}$ とする．式 (2) と同様に，軌道のスコアを軌道を通した報酬の累積和として定義すると，軌道 τ_i のスコア v_i は式 (7) で表される．

$$v_i = \sum_{t=0}^{|\tau_i|-1} \gamma^t R(\boldsymbol{x}_t^i) = \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{a}^{\mathrm{T}} \boldsymbol{k}(\boldsymbol{x}_t^i) = \boldsymbol{a}^{\mathrm{T}} \sum_{t=0}^{|\tau_i|-1} \gamma^t \boldsymbol{k}(\boldsymbol{x}_t^i) \quad (7)$$

r 本の軌道とそれらに対しエキスパートが付与したスコア $(\tau_1, v_1^*), \ldots, (\tau_r, v_r^*)$ が与えられたとき，式 (8) の回帰により，式 (6) のパラメータ $\boldsymbol{a} \in \mathbb{R}^n$ を推定する．

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a}}{\operatorname{argmin}} \|K\boldsymbol{a} - \boldsymbol{v}^*\| \quad (8)$$

ただし，$K \in \mathbb{R}^{r \times n}$ は式 (9) の行列である．

$$K = \begin{pmatrix} \sum_{t=0}^{|\tau_1|-1} \gamma^t \boldsymbol{k}(\boldsymbol{x}_t^1)^{\mathrm{T}} \\ \vdots \\ \sum_{t=0}^{|\tau_r|-1} \gamma^t \boldsymbol{k}(\boldsymbol{x}_t^r)^{\mathrm{T}} \end{pmatrix} \quad (9)$$

式（6）において，軌道 $\tau_1, \ldots, \tau_r$ で現れる観測値ベクトル $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ を全て用いた場合，n は非常に大きな値となる．学習に用いる軌道の長さが全て等しく $|\tau_i| - 1 = T$ であるとき，K の算出による計算量のオーダは $O(rnT)$ であり，連続値のタスクを扱う場合は現実的でない．そこで，軌道内に現れた観測値ベクトルをクラスタリングし，各クラスタの平均 $\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_m$ を代表値として用いる．

4. シミュレーション

提案手法の有用性を検証するため，物理シミュレータ MuJoCo[8] を利用した強化学習用インタフェース OpenAI Gym[9] による環境 Reacher を用いてシミュレーションを行った．Fig.1 のように，Reacher は 2 リンクマニピュレータの手先をターゲット位置に移動させる制御問題である．

提案手法，DM-IRL による推定報酬関数を用いて強化学習を行い，獲得した軌道をエキスパートによるスコアで評価した．

4.1 設定

マニピュレータのリンク i の関節角とその角速度，関節に加えるトルクをそれぞれ θ_i，$\dot{\theta}_i$，T_i とする．ただし，$i = 1, 2$ である．また，ターゲット位置の x 座標，y 座標をそれぞれ x_{target}，y_{target} とする．ターゲット位置は固定とし $x_{target} = 1.0$，$y_{target} = 1.0$，マニピュレータの初期状態は $\theta_i \sim \mathcal{U}(-\pi, \pi)$，$\dot{\theta}_i = 0$ とした．

状態 s はマニピュレータの関節角 θ_i，角速度 $\dot{\theta}_i$ とターゲット位置 x_{target}，y_{target}，行動 a は関節に加えるトルク T_i により定義される．シミュレーションで用いた観測値ベクトル $\boldsymbol{x}$（提案手法）と特徴量ベクトル $\boldsymbol{\phi}$（DM-IRL）は，式 (10) の通りである．

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ x_{target} \\ y_{target} \\ T_1 \\ T_2 \end{pmatrix}, \quad \boldsymbol{\phi}(s, a) = \begin{pmatrix} \sin\theta_1 \\ \sin\theta_2 \\ \cos\theta_1 \\ \cos\theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ x_{target} \\ y_{target} \\ d_x \\ d_y \\ T_1 \\ T_2 \end{pmatrix} \quad (10)$$

ただし，d_x，d_y はマニピュレータの手先とターゲット位置との x 方向，y 方向の距離であり，$\boldsymbol{x}$ とマニピュレータのリンク長から算出される．特徴量 $\boldsymbol{\phi}$ は gym における標準の特徴量に準拠して設定した．

Fig. 1 Simulator of a 2-link manipulator

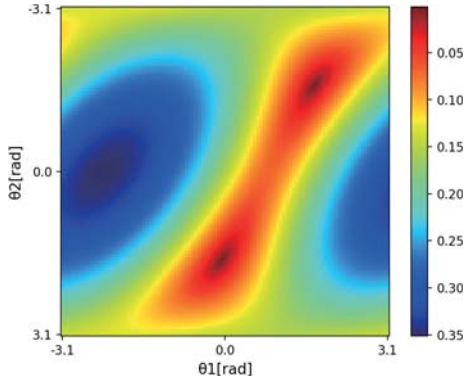


Fig. 2 A heatmap of distance to the target from the end-effector w.r.t. link angles

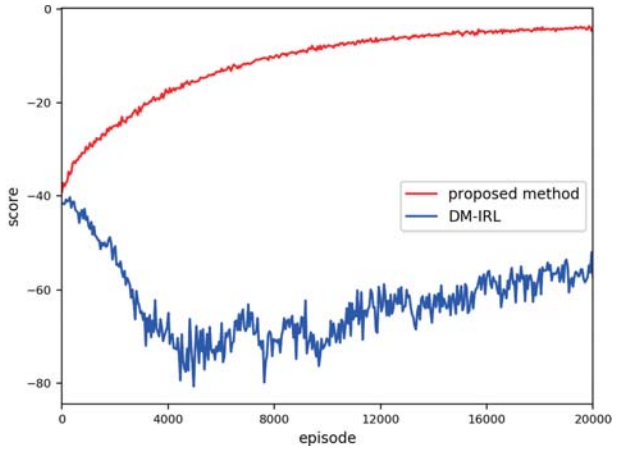


Fig. 3 Average score of each 50 episodes

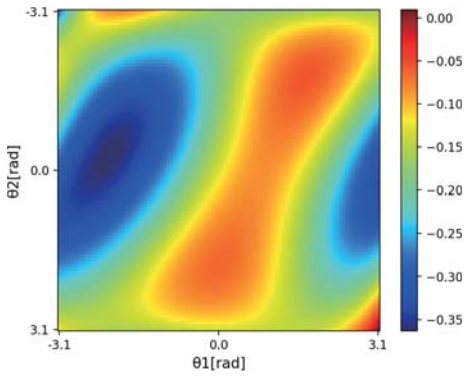
関節角に関する手先とターゲット位置との距離のヒートマップを Fig.2 に示す．Fig.1 におけるマニピュレータの関節角は，Fig.2 における右上の頂点に対応する関節角とほぼ等しい．

学習に用いる軌道 $\tau_1, \ldots, \tau_r$ は，ステップ数を 50 とし，各ステップにおいて各関節へのトルク入力 T_i を一様分布 $\mathcal{U}(-2.0, 2.0)$ からサンプルすることで生成した．軌道に対するエキスパートによるスコア $v_1^*, \ldots, v_r^*$ は式 (11) で定義した．

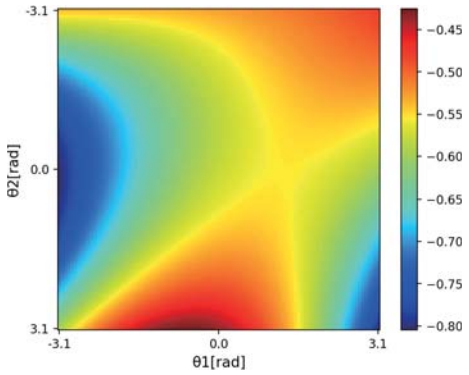
$$v_i^* = \sum_{t=0}^{|\tau_i|-1} \gamma^t \left(-\sqrt{{d_x^{(t)}}^2 + {d_y^{(t)}}^2} - ({T_1^{(t)}}^2 + {T_2^{(t)}}^2) \right) \quad (11)$$

ただし，$T^{(t)}$，$d^{(t)}$ はそれぞれステップ t におけるトルク，手先とターゲット位置との距離である．

式 (4)，(8) の回帰には最小二乗法を用い，提案手法におけるクラスタリングには K-Means を用いた．

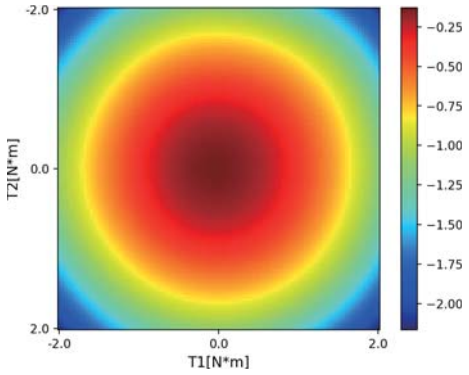


(a) Proposed method

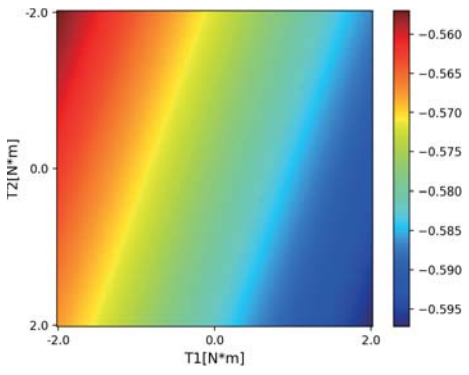


(b) DM-IRL

Fig. 4 Estimated reward function (link angles)



(a) Proposed method



(b) DM-IRL

Fig. 5 Estimated reward function (load torque)

学習に用いる軌道の本数を 2000 本とし，提案手法，DM-IRL により報酬関数を推定した．ただし，提案手法におけるクラスタリングはクラスタ数を 500 とした．続いて，推定した報酬関数を用いて強化学習を行い，式 (11) を用いて 50 エピソード毎の平均スコアを算出した．ここで，強化学習には Proximal Policy Optimization[10] を用い，エピソードのステップ数は 50 とした．

4.2 結果

推定した報酬関数を用いて行った強化学習における 50 エピソード毎の平均スコアの推移を Fig.3 に示す．提案手法では，エピソードの経過に伴いスコアが大きくなる様子が見てとれる．一方，DM-IRL による推定報酬関数を用いた場合，多数のエピソードを経てもスコアの向上は見られなかった．

また，$\dot{\theta}_i = 0$，$T_i = 0$ のときの関節角に関する推定報酬関数のヒートマップを Fig.4 に，$\theta_i = 0$，$\dot{\theta}_i = 0$ のときのトルクに関する推定報酬関数のヒートマップを Fig.5 に示す．それぞれ，(a) は提案手法，(b) は DM-IRL による推定報酬関数である．提案手法による推定報酬関数は，手先がターゲット位置付近に至る関節角において大きい値を示した．また，トルクに関しては T_1，T_2 が共にほぼ 0 となるときに最も大きい値を示した．これに対し DM-IRL では，トルクに関してはほぼ線形な報酬を示した．関節角に関しては，概形に Fig.2 との一定の類似性が見られるものの，提案手法と比べその類似性は小さい．

4.3 考察

報酬関数の推定にはランダムな方策から出力される軌道を用いたため，それらのスコアは必ずしも高くない．しかし，シミュレーションの結果から，提案手法により推定される報酬関数からは高いスコアで評価される軌道が生成可能であることが確認できたと考える．したがって，任意の軌道に対するスコアから，エキスパートが従う真の報酬関数と機能的に同等な報酬関数が推定可能であることが確認できた．

DM-IRL による推定報酬関数は特徴量の設定に依存するため，線形な報酬関数であっても有効な特徴量を用いればタスクを表現することができる．しかし，有効な特徴量の設計はタスクの複雑さに応じて困難になるため，カーネル関数により特徴量空間へ写像する提案手法は，複雑なタスクの達成において有用であるといえる．

5. 結言

任意の軌道のスコアから非線形な報酬関数を推定する逆強化学習を提案した．軌道に現れる観測値を入力とするカーネル関数を用いて報酬関数を非線形化することで，タスクに対する表現能力の向上を図った．マニピュレータの制御問題のシミュレーションにより，推定された報酬関数から学習された方策が高いスコアで評価される軌道を出力できることを確認した．

今後の展望として，クラスタリングに要する計算量の削減や学習に用いる軌道の生成方法の検討が挙げられる．

参考文献

[1] J. Kober, J. A. Bagnell and J. Peters, “Reinforcement Learning in Robotics: A Survey,” The International Journal of Robotics Research, Vol. 32, No. 11, pp. 1238–1274, 2013.

[2] A. Y. Ng and S. Russell, “Algorithms for Inverse Reinforcement Learning,” In Proceedings of the 17th International Confernce on Machine Learning, pp. 663–670, 2000.

[3] P. Abbeel and A. Y. Ng, “Apprenticeship Learning via Inverse Reinforcement Learning,” In Proceedings of the 21st International Confernce on Machine Learning, pp. 1–8, 2004.

[4] B. D. Ziebart, A. Maas, J. A. Bagnell and A. K. Dey, “Maximum Entropy Inverse Reinforcement Learning,” In Proceedings of the 23rd National Conference on Artificial Intelligence(AAAI’08), pp. 1433–1438, AAAI Press, 2008.

[5] A. Boularias, J. Kober and J. Peters, “Relative Entropy Inverse Reinforcement Learning,” In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, pp. 20–27, 2011.

[6] S. Levine, Z. Popović and V. Koltun, “Nonlinear Inverse Reinforcement Learning with Gaussian Processes,” In Proceedings of the 24th Advances in Neural Information Processing Systems, pp. 19–27, 2011.

[7] B. Burchfiel, C. Tomasi and R. Parr, “Distance Minimization for Reward Learning from Scored Trajectories,” In Proceedings of the 30th National Conference on Artificial Intelligence(AAAI’16), pp. 1–7, AAAI Press, 2016.

[8] E. Todorov, T. Erez and Y. Tassa, “MuJoCo: A physics engine for model-based control,” In Proceedings of Intelligent Robots and Systems, 2012 IEEE/RSJ International Conference on, pp. 5026–5033, IEEE, 2012.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang and W. Zaremba, “OpenAI Gym,” arXiv preprint arXiv:1606.01540, 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.