方策と環境モデルを生成モデルとして学習する敵対的生成模倣学習

○内部英治（国際電気通信基礎技術研究所）

1. はじめに

ビデオゲームや囲碁など計算機上で対象のダイナミクスが完全に記述できる問題で，ニューラルネットを用いた深層強化学習は人手で設計した方策（行動ルール）を超える性能を学習可能であることが示されている．一方でロボットなどの実世界との相互作用を含むタスクに強化学習を適用するにはまだまだ解決すべき問題も多く，ロボティクス固有の問題も指摘されている [7]．その一つに報酬設計問題がある．囲碁の場合，対戦の結果勝利すれば $+1$，敗北すれば -1，それ以外は 0 といった非常にスパースな報酬関数を使えば原理的には学習できる．スパースな報酬は容易に設計できるが，代償として実世界では不可能な試行回数を必要とするため，ロボット学習に用いるのは適切ではない．

そこで注目されるのが逆強化学習である．逆強化学習では報酬関数を設計する代わりに，タスクを達成しているエキスパートからの行動データが利用できると仮定し，エキスパートの報酬関数をデータから推定する．ロボット制御の問題では正解行動をデモンストレーションとして準備できる場合も多く，人や動物の意思決定の解析も可能となるため，非常に多くの研究がなされている [1]．

著者らはエキスパートからの行動データと学習者自身が生成した行動データから報酬を推定する逆強化学習と，推定された報酬を基に方策を改善する順強化学習を組み合わせたモデルフリー敵対的生成模倣学習 Model-Free Entropy-Regularized Imitation Learning (MF-ERIL) を提案している [14]．エントロピ正則に基づき導出された順・逆強化学習は報酬だけでなく状態価値関数も共有することで，学習効率を改善している．またハイパーパラメータも共有することで，従来の生成的模倣学習 [6, 4, 8] では無視されていた逆強化学習のハイパーパラメータを自然な形で導入できた．結果として，順・逆強化学習の学習速度を調整することが可能となった．

しかし MF-ERIL はモデルフリー強化学習と同様に，サンプル効率の悪さの問題が指摘されている．すなわち学習者の方策から行動データを生成するためには環境とのインタラクションが必要で，インタラクションの回数を削減することが大規模な問題に適用する際には必要になる．サンプル効率を改善する一つの方法は，環境のダイナミクスを明示的に推定し，環境モデルを実モデルの代わりに用いるモデルベース法がある．一般に，モデルベース法はモデルフリー法と比較しサンプル効率を大幅に改善することが知られている．しかし，モデルと実環境のギャップが学習の後期で問題となり，漸近性能が悪くなることも知られている．

そこで本研究ではサンプル効率を改善しつつも出るバイアスの問題を軽減するモデルベース敵対的生成模倣学習 Model-Based ERIL (MB-ERIL) を提案する．最大の特徴はモデル学習を通常の教師あり学習とするのではなく，得られる報酬系列に応じて重みづけする点である．さらに順・逆強化学習の損失関数を実データとモデルのギャップを補正する重点サンプリングを導入する．これは従来のモデルベース模倣学習では無視されてきた点である．従来の強化学習は，過去の方策から得られたデータを再利用するために重点サンプリングを用いることが多かったが，本研究では実環境とモデル環境の違いを補正するために用いられる．

OpenAI gym [3] で提供されているロボット制御課題を用いて MB-ERIL と従来手法を比較する．シミュレーション結果より，エキスパートからのデータ数に関しては従来法と同程度の効率であるが，環境とのインタラクションに関するデータ効率は従来法よりも改善されたことを示す．

2. モデルベース敵対的生成模倣学習

2.1 エントロピ正則された強化学習

離散時間の無限期間マルコフ決定過程 (Markov Decision Process; MDP) を考える．MDP は $(\mathcal{X}, \mathcal{U}, p_T, r, \gamma, p_0)$ の組によって定義される．ここで $\mathcal{X}$,$\mathcal{U}$ はそれぞれ状態空間，行動空間である．$p_T(\boldsymbol{x}' \mid \boldsymbol{x}, \boldsymbol{u})$ は状態 $\boldsymbol{x} \in \mathcal{X}$ で行動 $\boldsymbol{u} \in \mathcal{U}$ を実行した時に状態 $\boldsymbol{x}' \in \mathcal{X}$ に遷移する確率で，モデルフリー強化学習の枠組みでは未知である．$\tilde{r}(\boldsymbol{x}, \boldsymbol{u})$ は状態 $\boldsymbol{x}$，行動 $\boldsymbol{u}$ に対して与えられる即時報酬である．$\gamma \in (0, 1)$ は割引率，$p_0(\boldsymbol{x})$ は初期状態分布である．状態 $\boldsymbol{x}$ で行動 $\boldsymbol{u}$ を選択する確率を $\pi(\boldsymbol{u} \mid \boldsymbol{x})$ とする．強化学習の目的は期待割引積算報酬を最大にする方策を求めることである．

近年，報酬関数をエントロピによって正則化されたクラスの MDP が注目を集めている [2, 5, 9]．具体的には報酬関数が

$$\tilde{r}(\boldsymbol{x}, \boldsymbol{u}) \triangleq r(\boldsymbol{x}, \boldsymbol{u}) + \kappa^{-1}\mathcal{H}(\pi(\cdot \mid \boldsymbol{x})) - \eta^{-1}\mathrm{KL}(\pi(\cdot \mid \boldsymbol{x}) \,\|\, b(\cdot \mid \boldsymbol{x})) \quad (1)$$

のように正則化される．ここで $r(\boldsymbol{x}, \boldsymbol{u})$ は通常の意味での即時報酬関数，$\mathcal{H}(\pi(\cdot \mid \boldsymbol{x}))$ は方策 π のエントロピ，$\mathrm{KL}(\pi(\cdot \mid \boldsymbol{x}) \,\|\, b(\cdot \mid \boldsymbol{x}))$ は π とベースライン方策 $b(\boldsymbol{u} \mid \boldsymbol{x})$ の間の Kullback-Leibler (KL) ダイバージェンス，κ, η は実験者の定めるメタパラメータである．エントロピの役割は最適方策が決定論的になることを防ぎ，探索を促進する．KL ダイバージェンスの役割はベースライン方策 $b(\boldsymbol{u} \mid \boldsymbol{x})$ からあまり逸脱しないように方策改善ステップを保守的にする．

このとき以下の関係式が成り立つ．

$$Q(\boldsymbol{x},\boldsymbol{u}) = r(\boldsymbol{x},\boldsymbol{u}) + \eta^{-1}\ln b(\boldsymbol{u}\mid\boldsymbol{x}) + \gamma\mathbb{E}_{\boldsymbol{x}'\sim p_T(\cdot\mid\boldsymbol{x},\boldsymbol{u})}\left[V(\boldsymbol{x}')\right]. \quad (2)$$

$$V(\boldsymbol{x}) = \beta^{-1}\ln\int\exp\left(\beta Q(\boldsymbol{x},\boldsymbol{u})\right)\mathrm{d}\boldsymbol{u}, \quad (3)$$

$$\pi(\boldsymbol{u}\mid\boldsymbol{x}) = \exp\left[\beta\left(Q(\boldsymbol{x},\boldsymbol{u}) - V(\boldsymbol{x})\right)\right], \quad (4)$$

$V(\boldsymbol{x})$ は状態価値関数，$Q(\boldsymbol{x},\boldsymbol{u})$ は状態行動価値関数と呼ばれる．また，$\beta \triangleq 1/(\kappa+\eta)$ と定義している．行動が離散の場合，式 (3) の右辺は log-sum-exp 関数として知られる max 演算子を滑らかにしたものである．最適方策 (4) は確率的になっていることに注意されたい．$\eta = 0$ のとき Soft Q-learning や Soft Actor-Critic [5] が，$\kappa = 0$ のとき Dynamic Policy Programming (DPP) [2] が導出される．

2.2 Model-Free ERIL

MF-ERIL は，エキスパート方策 $\pi^E(\boldsymbol{u}\mid\boldsymbol{x})$ から収集された状態行動遷移対

$$\mathcal{D}^E = \{(\boldsymbol{x}_i,\boldsymbol{u}_i,\boldsymbol{x}'_i)\}_{i=1}^{N^E},$$
$$\boldsymbol{u}_i \sim \pi^E(\cdot\mid\boldsymbol{x}_i),\quad \boldsymbol{x}'_i \sim p_T(\cdot\mid\boldsymbol{x}_i,\boldsymbol{u}_i)$$

と，学習者の方策 π^L から生成された状態行動遷移対 $\mathcal{D}^L$ からエキスパート方策とそれを復元する報酬を推定する．ただしモデルフリーであるため，初期状態確率や状態遷移確率は未知である．

MF-ERIL の目的関数は Reverse KL ダイバージェンス

$$J^{\mathrm{MF-ERIL}}(\boldsymbol{\theta}) = \mathbb{E}_{\pi^L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')}\left[\frac{\pi^L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')}{\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')}\right]$$

と与えられる．ここで同時分布 $\pi^L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ は Markov 性の仮定の下

$$\pi^L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}') \triangleq p_T(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})\pi^L(\boldsymbol{u}\mid\boldsymbol{x})\pi^L(\boldsymbol{x})$$

と与えられる．$\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ も同様である．MF-ERIL の逆強化学習は，この Reverse KL ダイバージェンスの推定に対応し，二つの識別器（ロジスティック回帰）を用いた密度比推定 [11] によってなされる．一つ目の識別器は対数密度比 $\pi^L(\boldsymbol{x})/\pi^E(\boldsymbol{x})$ の推定に対応し，状態 $\boldsymbol{x}$ が学習者のデータであるとき 1，エキスパートデータのデータであるとき 0 となるよう構成し，

$$D^{(1)}(\boldsymbol{x}) = \frac{1}{1+\exp(-g(\boldsymbol{x}))}$$

と与えられる．これは通常の二値分類問題として $g(\boldsymbol{x})$ を推定できる．もう一つの識別器は対数密度比 $\pi^L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')/\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ の推定に対応するが，状態遷移確率 p_T がキャンセルアウトされること，$\pi^L(\boldsymbol{x})/\pi^E(\boldsymbol{x})$ のかわりに $g(\boldsymbol{x})$ を用いること，さらにエントロピ正則化強化学習の関係式を用いることで，識別器が構造化できることを利用する．結果として

$$D^{(2)}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}') = \frac{\exp(\beta\kappa^{-1}\ln\pi^L(\boldsymbol{u}\mid\boldsymbol{x}))}{\exp(\beta f(\boldsymbol{x},\boldsymbol{x}')) + \exp(\beta\kappa^{-1}\ln\pi^L(\boldsymbol{u}\mid\boldsymbol{x}))} \quad (5)$$

と構築する．ここで

$$f(\boldsymbol{x},\boldsymbol{x}') \triangleq r(\boldsymbol{x}) - \beta^{-1}g(\boldsymbol{x}) + \gamma V(\boldsymbol{x}') - V(\boldsymbol{x})$$

と定義している．式 (5) の識別器は従来の逆強化学習で用いられている識別器の拡張である．たとえば $g(\boldsymbol{x}) = 0$ かつ $\eta = 0$ とすれば AIRL [4] の識別器が，$\kappa = 0$ とすれば LogReg-IRL [13] の識別器と一致する．$D^{(2)}$ も通常の二値分類問題として $g(\boldsymbol{x})$ を推定できるが，推定結果として報酬関数だけでなく，状態価値関数も同時に推定できることが特徴である．

順強化学習は Soft Actor-Critic と同様の関係式を用いることで実現できる．

2.3 Model-Based ERIL

本研究では環境 $p_T(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})$ を明示的にモデル化し，モデルから生成されるデータを使った順・逆強化学習を実施する．環境のモデルを $q(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})$ とする．状態行動対 $(\boldsymbol{x},\boldsymbol{u})$ において，次の状態 $\boldsymbol{x}'$ がモデル q からせいせいされたものか，実環境 p_T から生成されたものかを判定する識別器

$$D^{(3)}(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u}) = \begin{cases} 1 & \boldsymbol{x}'\sim q(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u}) \\ 0 & \boldsymbol{x}'\sim p_T(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u}) \end{cases}$$

を導入する．この識別器の学習も二値分類問題として，以下の目的関数

$$J_D^{(3)} = \mathbb{E}_{\boldsymbol{x}'\sim q(\cdot\mid\boldsymbol{x},\boldsymbol{u}),(\boldsymbol{x},\boldsymbol{u})\sim p^{E/L}}[\ln D^{(3)}(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})] + \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim p^{E/L}}[\ln(1-D^{(3)}(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u}))] \quad (6)$$

を使って学習できる．ここで第 1 項の期待値を計算する分布は $(\boldsymbol{x},\boldsymbol{u})$ は実際の環境からサンプルされているが，$\boldsymbol{x}'$ のみがモデルから生成されているのに対し，第 2 項は $(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ すべてが実際の環境からサンプルされていることに注意されたい．$D^{(3)}$ が得られると密度比 $q(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})/p_T(\boldsymbol{x}'\mid\boldsymbol{x},\boldsymbol{u})$ の推定量が計算される．これを $ISW(D^{(3)})$ とする．

逆強化学習の損失関数は重点サンプリングを用いて

$$J_D^{(2)} = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim q^L}[ISW(D^{(3)})\ln D^{(2)}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')] + \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim p^E}[\ln(1-D^{(2)}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}'))] \quad (7)$$

と修正される．右辺第 1 項の期待値計算に用いる確率分布を $(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim p^L$ として実環境から得られる状態遷移を用いるように設定したものが，MF-ERIL の逆強化学習の目的関数となる．同様に順強化学習において中心的な役割を果たす状態行動価値関数の損失関数は

$$J_Q = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim q^L}\left[ISW(D^{(3)})(Q(\boldsymbol{x},\boldsymbol{u}) - \bar{Q}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}'))^2\right] \quad (8)$$

と修正される．ここで

$$\bar{Q}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}') = r(\boldsymbol{x}) + \eta^{-1}\ln\pi^L(\boldsymbol{u}\mid\boldsymbol{x}) + \gamma\bar{V}(\boldsymbol{x}')$$

であり，$\bar{V}(\boldsymbol{x}')$ は深層強化学習で用いられるターゲットネットワークである．

最後にモデルの学習法を説明する．最尤推定などで学習することも可能であるが，順強化学習の目的関数を重点サンプリングで補正したもの

$$\max_{\pi^L} \mathbb{E}_{\boldsymbol{z}\sim q^L}\left[ISW(D^{(3)})\tilde{r}(\boldsymbol{z})\right]$$

の分散を最小にするように構成する．ここで $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')$ であり，$\tilde{r}$ は逆強化学習で得られた報酬と状態価値関数から計算される TD 誤差とする．上式は不偏推定量であるが，分散を最小にするように q^L を修正できる．最適なモデルは

$$q^{L*} \propto |\tilde{r}(\boldsymbol{z})| p^L(\boldsymbol{z}), \tag{9}$$

と与えられる．しかし p^L に依存しているためこれは計算不可能である．そこで両者の KL ダイバージェンスを最小にするように q^L を推定する．KL ダイバージェンスの勾配は

$$\nabla \mathrm{KL}(q^{L*} \parallel q) \propto -\mathbb{E}_{p^L}\left[|\tilde{r}(\boldsymbol{z}|\nabla \ln q(\boldsymbol{x}' \mid \boldsymbol{x}, \boldsymbol{u})\right] \tag{10}$$

と与えられる．

2.4 Model-Based Model-Free ERIL

MB-ERIL はモデル q を用いて学習する．分散を最小にするように構成しても実環境とのギャップは存在し，従来研究で指摘されるように学習後期での漸近性能はモデルフリーに劣ることが想定される．そこで学習のエピソード数に対して，MB-ERIL から MF-ERIL に切り替えて方策を学習する MBMF-ERIL を提案する．これは $J_D^{(2)}$ や J_Q の損失関数を切り替えることで実現できる．

3. シミュレーション

提案手法の有効性を検証するために，OpenAI gym [3] で提供されている Ant, Humanoid という 2 種類のロボット制御課題に適用する．これらは物理エンジン MuJoCo [12] を用いている．これらのタスクの目的はできるだけ速く移動することである．まず本来設定されている報酬関数をもとに最適方策 π^E を TRPO [10] によって学習し，そこから得られるデータを $\mathcal{D}^E$ として用いる．関数近似として用いるニューラルネットワークの構造は従来研究 [4, 6] を参考に構築した．対数密度 $g(\boldsymbol{x})$，報酬 $r(\boldsymbol{x})$，状態価値関数 $V(\boldsymbol{x})$，行動価値関数 $Q(\boldsymbol{x}, \boldsymbol{u})$ は 2 層のニューラルネットワークを用い，活性度関数は ReLU，ユニット数はそれぞれ 400, 300 とした．また方策 $\pi^L(\boldsymbol{u} \mid \boldsymbol{x})$ はガウス分布によって構成し，その平均値を同じ構成のニューラルネットワークで表現した．1 エポックあたり学習方策 π^L によって生成される軌跡は 100 とし，各軌跡は 50 個の状態行動遷移対を含むとする．

まずエキスパート方策 π^E からの軌跡の数を Ant 環境では 30，Humanoid 環境では 350 と設定したときの順強化学習の性能を比較した．図 1, 2 に MB-ERIL, MBMF-ERIL と MF-ERIL，DAC，BC を比較した結果を示す．ただし模倣学習は不良設定問題であるため，各手法で推定された報酬は直接比較できない．そこで最終的に獲得された学習方策をシミュレータで提供される本来の報酬を使ったエピソード当たりの総報酬を正規化したもので評価する．Ant 環境では MBMF-ERIL, MB-ERIL，MF-ERIL，DAC の順で正規化総報酬の最大値に到達している．BC は従来研究の報告通り，エキスパート方策の性能には到達しなかった．Ant 環境ではモデル学習により，モデルフリーの MF-ERIL，DAC の学習効率を 10 倍程度改善することができた．

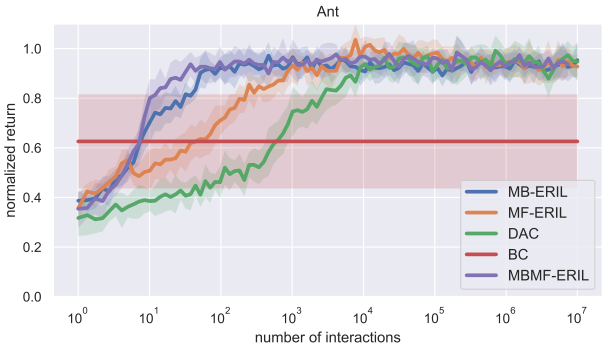

図 1　Ant 環境における順逆強化学習の性能評価

図 2　Humanoid 環境における順逆強化学習の性能評価

Humanoid 環境でも同様の結果が得られたが，最終的な制御性能を確認すると MB-ERIL は MF-ERIL よりも劣っていることがわかった．Humanoid 環境は Ant 環境よりも複雑でモデル化誤差の影響が大きいことも一つの原因と考えられ，ハイブリッドにした MBMF-ERIL は漸近性能も改善していることがわかる．実際に最終的に獲得された方策について，負の対数尤度をエキスパート方策から生成したテストデータを用いて評価した．結果を図 3 に示す．MBMF-ERIL と MF-ERIL の方策には統計的な優位さはなかったが，MB-ERIL とは統計的な優位さが確認された．このことは MB-ERIL で獲得された方策は MF-ERIL，MBMF-ERIL で獲得された方策とは異なることを意味している．

次に $\mathcal{D}^E$ からのサンプル数を変えることで逆強化学習のデータ効率を評価する．Ho and Ermon [6] に従い，一つの軌跡が 50 個の状態行動遷移対 $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')$ を含む，つまり 1 エピソードあたりのステップ数を 50 とする．図 4, 5 に MB-ERIL，MBMF-ERIL と MF-ERIL，DAC，BC を比較した結果を示す．すべての敵対的生成模倣学習は BC よりも少ないエキスパートデータ数で高い制御性能を示す方策を獲得している．一方で MB-ERIL, MF-ERIL の間にはそれほど違いは見られず，モデルの学習による逆強化学習のサンプル効率の改善は得られなかった．MBMF-ERIL のようにハイ

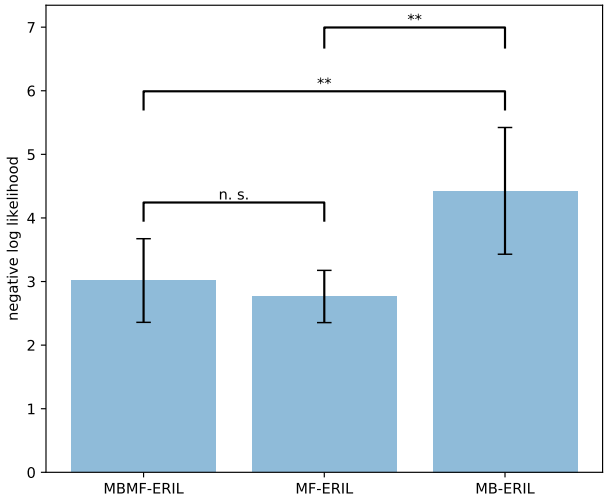

図 3　Humanoid 環境で得られた最終方策の比較

図 4　Ant 環境における逆強化学習の性能評価

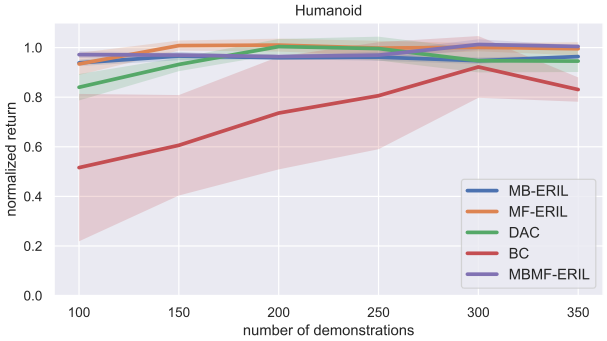

図 5　Humanoid 環境における逆強化学習の性能評価

ブリッドにしたことによる効果もなかった．

4.　おわりに

本稿ではエントロピ正則化強化学習に基づいたモデルベース敵対的生成模倣学習 MB-ERIL を提案した．MB-ERIL，DAC, BC よりも学習効率を改善しつつ，通常のモデルベース法で問題となる漸近性能も改善することが確認された．また MF-ERIL と MB-ERIL を統合することで，漸近性能をさらに改善できることが示された．

本研究では重点サンプリングで用いる密度比の推定に識別器 $D^{(3)}$ を導入している．この密度比は逆強化学習の計算に用いている識別器 $D^{(2)}$ のように構造化されていないため，学習には多くのサンプルを必要とする．今後は $D^{(3)}$ を構造化するとともに，実ロボットを用いた検証を計画している．

謝　辞　本研究の成果は，防衛装備庁が実施する安全保障技術研究推進制度 JPJ004596 および JST、未来社会創造事業、JPMJMI21B1 の支援を受けたものである．また，本研究の一部は国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP20006）の結果得られたものです．

参　考　文　献

[1] S. Arora and P. Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 2021.

[2] M. G. Azar, V. Gómez, and H. J. Kappen. Dynamic policy programming. *Journal of Machine Learning Research*, 13:3207–3245, 2012.

[3] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. OpenAI Gym. *arXiv preprint*, 2016.

[4] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *Proc. of the 6th International Conference on Learning Representations*, 2018.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. of the 35th International Conference on Machine Learning*, pp. 1856–1865, 2018.

[6] J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*. 2016.

[7] J. Ibarz, J. T. abd C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine. How to train your robot with deep reinforcement learning: Lessons we have learned. *The International Journal of Robotics Research*, 2021.

[8] I. Kostrikov, K. K. Agrawal, D. Dwibedi, S. Levine, and J. Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *Proc. of the 7th International Conference on Learning Representations*, 2019.

[9] T. Kozuno, E. Uchibe, and K. Doya. Theoretical analysis of efficiency and robustness of softmax and gap-increasing operators in reinforcement learning. In *Proc. of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

[10] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proc. of the 32nd International Conference on Machine Learning*, pp. 1889–1897, 2015.

[11] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

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

[13] E. Uchibe. Model-free deep inverse reinforcement learning by logistic regression. *Neural Processing Letters*, 47(3):891–905, 2018.

[14] E. Uchibe and K. Doya. Forward and inverse reinforcement learning sharing network weights and hyperparameters. *arXiv: 2008.07284*, 2021.