The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

エントロピ正則された強化学習を用いた模倣学習

Imitation learning based on entropy-regularized reinforcement learning

内部 英治 *1
Eiji Uchibe

*1国際電気通信基礎技術研究所
ATR Computational Neuroscience Labs.

This paper proposes Entropy-Regularized Imitation Learning (ERIL) that is given by a combination of forward and inverse reinforcement learning. ERIL utilizes the soft Bellman optimality equation in which the reward function is augmented by the entropy of the learning policy and the Kullback-Leibler (KL) divergence between the learning and the baseline policies. We show that inverse RL is interpreted as estimating the log-ratio between two policies and the log-ratio is efficiently solved by binary logistic regression. Forward RL is given by a variant of Dynamic Policy Programming and our algorithm is interpreted as minimization of the KL divergence between the learning policy and the estimated expert policy. Experimental results on the MuJoCo-simulated environments show that ERIL is more sample efficient than the previous methods such as GAIL and AIRL because the forward RL step of ERIL is off-policy.

1. はじめに

強化学習は最適方策（制御則）を試行錯誤によって獲得するための手法で，知能ロボットに自律的な学習能力を与えるために必要である．またヒトや動物の意思決定過程の計算モデルとして計算論的神経科学の分野で注目されている．一般に強化学習では各状態で実行した行動に対する即時評価である報酬を必要とする．しかし所望の最適方策を達成する報酬を設計するのは簡単ではない．たとえばタスクを達成した場合にのみ正の報酬値を与え，それ以外は 0 であるような非常にスパースな報酬関数を用いた場合，原理的には最適方策を学習できるが必要とするデータ数や学習時間は非常に膨大なものとなる．学習効率を改善するために複雑な報酬関数を設計することも考えられるが，多くの場合は意図しない最適方策を学習することになる．

この問題に対処するための方法として逆強化学習がある．逆強化学習ではタスクを達成しているエキスパートの存在を仮定し，エキスパートが用いている報酬関数をエキスパートからのデータから推定する．近年，強化学習と逆強化学習の組み合わせが敵対的生成ネットワーク (Generative Adversarial Networks; GAN) として定式化できることが示された [3, 4, 6]．逆強化学習は GAN における識別器に対応し，エキスパートからのデータとロボット自身が生成したデータを区別する．強化学習は GAN における生成器に対応し，逆強化学習によって推定された報酬の期待積算を最大にするように新しい方策を学習する．Generative Adversarial Imitation Learning (GAIL) [6] や Adversarial Inverse Reinforcement Learning (AIRL) [4] は逆強化学習と強化学習を繰り返すことで単純な模倣学習である Behavior Cloning (BC) よりもエキスパート方策を精度良く復元できることを示した．GAIL や AIRL はエキスパートからのサンプル数が少ない場合でも有効であるが，その一方で生成器の改善に必要な環境とのインタラクションに関するデータ効率は低いことが知られている．一つの理由として GAIL や AIRL は方策オン型の強化学習法である Trust Region Policy Optimization (TRPO) [9] を用いていることがあげられる．重点サンプリング等の技術を用いない限り過去に学習方策が生成したデータを再利用できないため方策の改善には現在の学習方策が多くのデータを生成する必要がある．

そこで強化学習時のデータ効率を改善するために，本研究ではエントロピ正則された模倣学習 (Entropy-Regularized Imitation Learning; ERIL) を提案する．報酬関数は学習方策のエントロピと学習方策とベースライン方策の Kullback-Leibler (KL) ダイバージェンスによって正則化される．この定式化のもとで得られるソフトベルマン最適方程式を用いて，逆強化学習を密度比推定問題として定式化する．密度比推定はエキスパート方策からのデータと学習者自身が生成したデータを区別するロジスティック回帰によって効率的に解くことができる．識別器は報酬，状態価値関数，学習者の方策によって記述され，従来法で用いられた識別器 [4, 12] を一般化したものである．逆強化学習の結果エキスパート方策が推定されるため，KL ダイバージェンスによって新たな学習方策を更新する，これは方策オフ型の強化学習である Soft Actor-Critic [5] で用いられるものと同一で，強化学習時におけるデータ効率を期待できる．

OpenAI gym [2] で提供されているロボット制御課題を用いて 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．

連絡先: 内部英治，国際電気通信基礎技術研究所 脳情報研究所 ブレインロボットインタフェース研究室，〒 619-0288 京都府相楽郡精華町光台二丁目 2 番地 2，uchibe@atr.jp

2. エントロピ正則化された強化学習

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 [1, 5, 7]．具体的には報酬関数が

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

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，$\mathrm{KL}(\pi(\cdot \mid \boldsymbol{x}) \parallel b(\cdot \mid \boldsymbol{x}))$ 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．このとき以下のベルマン最適性方程式

$$V(\boldsymbol{x}) = \max_{\pi} \mathbb{E}_{\boldsymbol{u}\sim\pi(\cdot|\boldsymbol{x})} \left[r(\boldsymbol{x},\boldsymbol{u}) - \kappa \ln \pi(\boldsymbol{u} \mid \boldsymbol{x}) - \eta \ln \frac{\pi(\boldsymbol{u} \mid \boldsymbol{x})}{b(\boldsymbol{u} \mid \boldsymbol{x})} + \gamma \mathbb{E}_{\boldsymbol{x}'\sim p_T(\cdot|\boldsymbol{x},\boldsymbol{u})} \left[V(\boldsymbol{x}') \right] \right] \quad (2)$$

が成り立つ．ここで $V(\boldsymbol{x})$ は状態価値関数と呼ばれる．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．エントロピ正則化の利点は式 (2) の右辺の方策に関する最大化が Lagrange の未定乗数法によって解析的に求められることである．結果として状態価値関数 $V(\boldsymbol{x})$ と状態行動価値関数 $(\boldsymbol{x},\boldsymbol{u})$ に対して以下の関係式

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

$$Q(\boldsymbol{x},\boldsymbol{u}) = r(\boldsymbol{x},\boldsymbol{u}) + \eta \ln b(\boldsymbol{u} \mid \boldsymbol{x}) + \gamma \mathbb{E}_{\boldsymbol{x}'\sim p_T(\cdot|\boldsymbol{x},\boldsymbol{u})} \left[V(\boldsymbol{x}') \right]. \quad (4)$$

が成り立つ．ここで

$$\beta \triangleq \frac{1}{\kappa + \eta}$$

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．最適方策は

$$\pi(\boldsymbol{u} \mid \boldsymbol{x}) = \exp\left[\beta\left(Q(\boldsymbol{x},\boldsymbol{u}) - V(\boldsymbol{x})\right)\right], \quad (5)$$

と与えられる．$\eta = 0$ のとき Soft Q-learning や Soft Actor-Critic [5] が，$\kappa = 0$ のとき Dynamic Policy Programming (DPP) [1] が導出される．

3. エントロピ正則化された模倣学習

エントロピ正則化された強化学習は報酬関数 $r(\boldsymbol{x},\boldsymbol{u})$ が与えられれば最適方策を求めることができる．本節では報酬関数のかわりに最適方策から生成された状態行動データから最適方策を求める模倣学習を提案する．

3.1 密度比推定による逆強化学習

式 (1) において最適方策を $\pi(\boldsymbol{u} \mid \boldsymbol{x}) \equiv \pi^E(\boldsymbol{u} \mid \boldsymbol{x})$，ベースライン方策を $b(\boldsymbol{u} \mid \boldsymbol{x}) \equiv \pi^G(\boldsymbol{u} \mid \boldsymbol{x})$ とする．このとき式 (4), (5) より以下の関係式

$$\frac{1}{\beta} \ln \frac{\pi^E(\boldsymbol{u} \mid \boldsymbol{x})}{\pi^G(\boldsymbol{u} \mid \boldsymbol{x})} = r(\boldsymbol{x},\boldsymbol{u}) - \kappa \ln \pi^G(\boldsymbol{u} \mid \boldsymbol{x}) + \gamma \mathbb{E}_{\boldsymbol{x}'\sim p_T(\cdot|\boldsymbol{x},\boldsymbol{u})} \left[V(\boldsymbol{x}') \right] - V(\boldsymbol{x}) \quad (6)$$

を得る．さらに $\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ を

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

と提議し，$\pi^G(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ も同様に定義する．このとき式 (6) の左辺は

$$\ln \frac{\pi^E(\boldsymbol{u} \mid \boldsymbol{x})}{\pi^G_k(\boldsymbol{u} \mid \boldsymbol{x})} = \ln \frac{\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')}{\pi^G_k(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')} - \ln \frac{\pi^E(\boldsymbol{x})}{\pi^G_k(\boldsymbol{x})} \quad (7)$$

と書き換えることができる．式 (7) の右辺は二つの密度比の差であり，それぞれ π^E, π^G からのサンプルがあれば推定できる [10]．たとえば $\pi^E(\boldsymbol{u} \mid \boldsymbol{x})$ から収集された状態行動遷移を

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

とし，同様に π^G からのサンプル $\mathcal{D}^G$ を収集する．密度比推定問題を通して報酬や状態価値関数を推定することが逆強化学習に対応する．

密度比を推定する様々な手法が提案されている [10] が，ここではロジスティック回帰を用いる．そのために二値変数 $L \in \{0, 1\}$ を導入し，$L = 1$ のとき π^E からサンプル，$L = 0$ のとき π^G からサンプルする．すなわち

$$\pi^G(\boldsymbol{x}) = \Pr(\boldsymbol{x} \mid L = 0), \quad \pi^E(\boldsymbol{x}) = \Pr(\boldsymbol{x} \mid L = 1)$$

とする．このとき

$$\ln \frac{\pi^E(\boldsymbol{x})}{\pi^G(\boldsymbol{x})} = \ln \frac{D(\boldsymbol{x})}{1 - D(\boldsymbol{x})} - \ln \frac{\Pr(L = 1)}{\Pr(L = 0)}$$

と書き換えることができる．ここで $D(\boldsymbol{x})$ は状態 $\boldsymbol{x}$ が π^E からのサンプルであるかどうかを識別する識別器 $D(\boldsymbol{x}) \triangleq \Pr(L = 1 \mid \boldsymbol{x})$ である．同様に $\pi^E(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')/\pi^G_k(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ も識別器 $D(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}') \triangleq \Pr(L = 1 \mid \boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ によって記述できる．以上より式 (6) を

$$\frac{1}{\beta} \ln \frac{D(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')}{1 - D(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')} = \frac{1}{\beta} \ln \frac{D(\boldsymbol{x})}{1 - D(\boldsymbol{x})} + r(\boldsymbol{x},\boldsymbol{u}) - \kappa \ln \pi^G(\boldsymbol{u} \mid \boldsymbol{x}) + \gamma V(\boldsymbol{x}') - V(\boldsymbol{x}) \quad (8)$$

と書き換える．対数密度比 $\ln D(\boldsymbol{x})/(1 - D(\boldsymbol{x}))$ を $g(\boldsymbol{x})$ とおく．このとき識別器 $D(\boldsymbol{x})$ は

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

となり，通常の二値分類問題として $g(\boldsymbol{x})$ を推定できる．同様にして $D(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')$ を

$$D(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}') = \frac{\exp(\beta f(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}'))}{\exp(\beta f(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')) + \exp(\beta\kappa \ln \pi^G(\boldsymbol{u} \mid \boldsymbol{x}))} \quad (9)$$

と構築する．ここで

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

と定義している．式 (9) 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 [12] の識別器と一致する．

以上をまとめると逆強化学習は次の二つのステップで構成される．まず識別器 $D(\boldsymbol{x})$ を学習することで $g(\boldsymbol{x})$ を推定する．次に識別器 $D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')$ を学習して $r(\boldsymbol{x}, \boldsymbol{u})$ と $V(\boldsymbol{x})$ を推定する．この際 $g(\boldsymbol{x})$ と $\pi^G(\boldsymbol{u} \mid \boldsymbol{x})$ は固定する．$D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')$ は対数尤度

$$J(D) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim\mathcal{D}^E}\left[\ln D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')\right] + \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim\mathcal{D}^G}\left[\ln\left(1 - D(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')\right)\right]$$

を最大にすることで得られる．ここで $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}') \sim \mathcal{D}$ は $\mathcal{D}$ からのサンプルの期待値を意味する．

3.2 ベースライン方策の更新

密度比推定による逆強化学習の結果，エキスパート方策 π^E の推定値

$$\frac{1}{\beta}\ln\hat{\pi}^E(\boldsymbol{u} \mid \boldsymbol{x}) \approx r(\boldsymbol{x}, \boldsymbol{u}) + \eta \ln \pi^G(\boldsymbol{u} \mid \boldsymbol{x}) + \gamma V(\boldsymbol{x}') - V(\boldsymbol{x})$$

が得られる．これを用いて新しいベースライン方策 π^G_{new} を KL ダイバージェンス

$$J(\pi) = \mathbb{E}\left[\mathrm{KL}(\pi(\cdot \mid \boldsymbol{x}) \parallel \hat{\pi}^E(\cdot \mid \boldsymbol{x}))\right] \tag{10}$$

を最小にするように求める．

ベースライン方策の更新は Soft Actor-Critic [5] で用いられるものと同一である．Soft Actor-Critic では最適方策は以下の KL ダイバージェンス

$$J(\pi) = \mathbb{E}\Big[\mathrm{KL}\Big(\pi(\cdot \mid \boldsymbol{x}) \parallel \exp\big\{\beta\left(Q(\boldsymbol{x}, \cdot) - V(\boldsymbol{x})\right)\big\}\Big)\Big],$$

を最小にすることで計算される．Soft Actor-Critic では報酬 $r(\boldsymbol{x}, \boldsymbol{u})$ が与えられたとき式 (3), (4) を用いて $V(\boldsymbol{x})$ と $Q(\boldsymbol{x}, \boldsymbol{u})$ を計算したのちに最適方策を推定するのに対し，提案手法ではエキスパートからのデータ $\mathcal{D}^E$ が与えられたときに密度比推定によって最適方策を推定することになる．よって式 (10) による方策改善が強化学習ステップに相当する．

ただし式 (10) による更新では $\mathcal{D}^E$ と現在の学習方策 π^G によって生成されたデータ $\mathcal{D}^G$ だけから推定されたエキスパート方策を用いているため，過去の学習方策が用いたデータを利用していない．そこで逆強化学習によって推定された報酬と対数密度を固定し，状態価値関数と状態行動価値関数を Soft Actor-Critic にしたがって更新する．式 (5) より，状態価値関数を更新するための目的関数は

$$J(V) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[\frac{1}{2}\left(V(\boldsymbol{x}) - \tilde{V}(\boldsymbol{x})\right)^2\right],$$

と与えられる．ここで

$$\tilde{V}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{u}\sim\pi^G}\left[Q(\boldsymbol{x}, \cdot) - \beta^{-1}\ln\pi^G(\cdot \mid \boldsymbol{x})\right]$$

である．状態行動価値関数を学習する目的関数は

$$J(Q) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{u},\boldsymbol{x}')\sim\mathcal{D}}\left[\frac{1}{2}\left\{Q(\boldsymbol{x}, \boldsymbol{u}) - \tilde{Q}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}')\right\}^2\right],$$

となる．ここで

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

とする．$\mathcal{D}$ はエキスパートデータも含めたすべてのデータをまとめたものである．

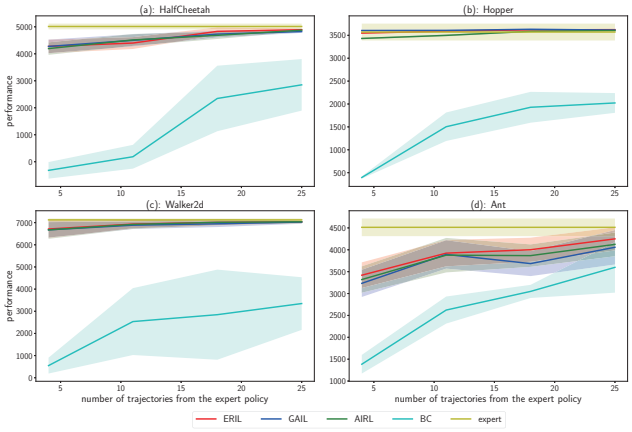

図 1: エキスパートからのデータ数についての性能比較．横軸はエキスパート方策 π^E から与えられた軌跡の個数，縦軸は本来の報酬のもとで評価した場合の性能を示す．

4. 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 [2] で提供されている Half-Cheetah, Hopper, Walker, Ant という 4 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 [11] 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 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 [*1]．また方策 $\pi^G(\boldsymbol{u} \mid \boldsymbol{x})$ 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．初期ベースライン方策は BC によって初期化する．1 エポックあたり学習方策 π^G 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．正則化のパラメータは $\kappa = 1, \eta = 10$ とし，割引率は $\gamma = 0.99$ とした．

まず $\mathcal{D}^E$ 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が 50 である．逆強化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のもとで評価する．図 1 に ERIL と GAIL, AIRL, BC を比較した結果を示す．ERIL, GAIL, AIRL ともに BC よりも高い制御性能を示す方策を獲得している．一方で 3 種類の模倣学習の間にはそれほど違いは見られなかった．Fu et al. [4] は GAIL と AIRL で実質的な差はなかったことを報告しており，本報告の結果と一致する．また前述したように対数密度 $g(\boldsymbol{x})$ とメタパラメータ κ, η の値によって ERIL と AIRL の識別器は一致する．実際にこの課題では学習方策を BC によって初期化しているため，ほとんどの状態 $\boldsymbol{x}$ において $g(\boldsymbol{x}) \approx 0$ となっていた．

次にエキスパート方策 π^E からの軌跡の数は 25 と固定し，

*1 Fu et al. [4] が指摘したように逆強化学習では報酬を状態と行動の関数として近似した場合は表現能力の高さから過剰適合しやすい．そのため報酬関数は状態の身に依存する関数として表現した．

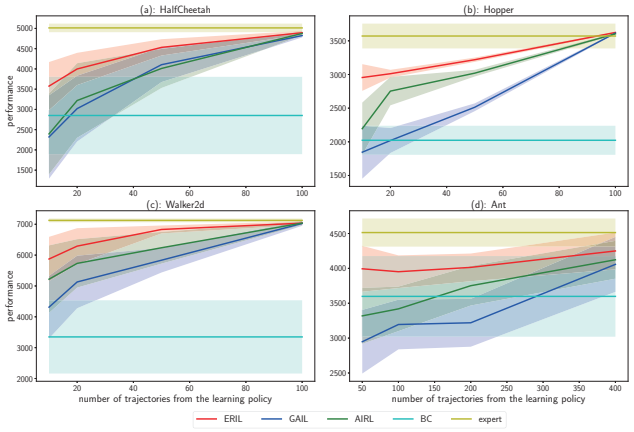

図 2: 学習方策からのデータ数についての性能比較．横軸は学習方策 π^G から各エポックで与えられた軌跡の個数，縦軸は本来の報酬のもとで評価した場合の性能を示す．

学習方策が生成する軌跡の数を変更した場合の性能を比較した．図 2 に結果を示す．軌跡の数が少ない場合，ERIL は GAIL, AIRL, BC よりも高い性能を達成する学習方策を獲得した．この理由としては ERIL は方策オフ型の強化学習を用いているのに対し，GAIL と AIRL は方策オン型である TRPO を用いているためであると考えられる．

5. 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模倣学習 ERIL を提案した．GAN-GCL, GAIL, AIRL などの従来法と異なり，提案手法はソフトベルマン最適方程式に基づき導出されている．そのため強化学習に相当するベースラインの更新ステップは方策オフ型である．シミュレーション結果より ERIL は従来法よりもサンプル効率が良いことが示された．

ERIL には注意深く調整すべきメタパラメータがあり，強化学習の場合には式 (1) で用いている正則化のパラメータ κ, η は学習過程の安定性や関数近似誤差に影響を与えることが知られている [7]．逆強化学習時においても識別器の性能に影響を与えるが，これらのパラメータをどのように学習中に調整するかは今後の課題である．

強化学習ステップにおける ERIL と AIRL の性能差は主に Soft Actor-Critic と TRPO の違いによって説明できると考えられる．Neu et al. は DPP，TRPO など様々な強化学習アルゴリズムの性質を調査した [8] が，Soft Actor-Critic や逆強化学習を組み合わせた場合の性質はいまだ明らかではない．この研究を拡張することで ERIL と AIRL の差を考察することも今後の課題である．

謝辞 本研究の成果は，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) の委託業務および JST，未来社会創造事業，JPMJMI18B8 の支援を受けたものである．

参考文献

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

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

[3] C. Finn, P. Christiano, P. Abbeel, and S. Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. In *NIPS 2016 Workshop on Adversarial Training*, 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*, pages 1856–1865, 2018.

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

[7] 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.

[8] G. Neu, V. Gómez, and A. Jonsson. A unified view of entropy-regularized markov decision processes. *arXiv preprint*, 2017.

[9] 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*, pages 1889–1897, 2015.

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

[11] 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*, pages 5026–5033, 2012.

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