階層型強化学習における効率的な下位方策の分離と獲得

EP19105 森　勇貴　　指導教授：山下 隆義

1. はじめに

階層型強化学習は，サブタスクごとの方策である下位方策と，その方策を選択する上位方策により，複雑なタスクを解くことができる．階層型強化学習の代表手法として，下位方策を学習過程で自動獲得する Meta Learning Shared Hierarchies (MLSH) [1] がある．MLSH は下位方策の自動獲得を実現している反面，役割が類似した下位方策が獲得されてしまう．そこで，本研究では下位方策間を分離する損失関数 JS Divergence Loss の導入を提案する．これにより，下位方策間で役割が異なる方策の獲得を促す．

2. Meta Learing Shared Hierarchies

MLSH とは，下位方策を自動獲得する階層型強化学習手法である．上位方策は一定ステップ毎に下位方策を選択し，選択された下位方策が環境に対する行動を出力する．そして上位方策と選択された下位方策のみ，報酬をもとに学習する．また，ここでの上位方策は下位方策が学習するごとに変化するため，一定ステップ経過で重みを初期化し，変化後の下位方策から再度適切な下位方策の選択方法を模索する．これにより，環境とのインタラクションを通じて得る経験を基に獲得することが可能となる．このように獲得した下位方策を用いて，上位方策のみ学習を行うことで複雑なタスクを解くことが可能である．

3. 提案手法

本研究では役割の異なる下位方策獲得を目的とし， JS Divergence Loss L_{JSdiv} を用いた下位方策分離法を提案する．図 1 に提案手法の構造を示す．

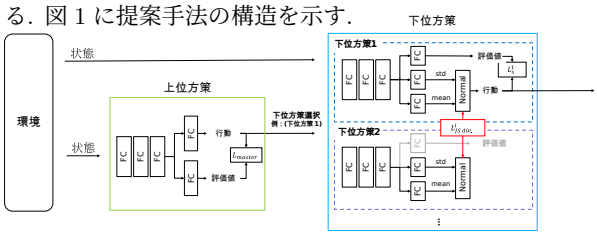

図 1: 提案手法の構造

下位方策は確率分布のため，確率分布間の相違度を示す JS Divergence を用いることで下位方策間の相違度を算出する．JS Divergence を用いた損失関数 $L_{\mathrm{JSdiv.}}$ を式 (1) に示す．

$$L^i_{\mathrm{JSdiv.}} = js_{\max} - \frac{1}{(n-1)} \sum_{j \neq i}^{n} f_{\mathrm{JSdiv.}}(JS[\pi^i_{sub}, \pi^j_{sub}]) \quad (1)$$

$$f_{\mathrm{JSdiv.}}(x) = \begin{cases} 0, & \text{if } x = 0, \\ js_{max}, & \text{if } x \neq 0, \end{cases} \quad (2)$$

ここで，$L^i_{\mathrm{JSdiv.}}$ は i 番目の下位方策に対する JS Divergence Loss 値，n は下位方策数，$\pi_{\mathrm{sub}i}, \pi_{\mathrm{sub}j}$ は i, j 番目の下位方策 $(j = \backslash i)$，JS[·] は JS Divergence である．JS Divergence は相違度を示すため下位方策間の分布が離れているほど高い値となる．そのため，下位方策間での JS Divergence が取りうる最大値 $js_{\max}$ を設定し，下位方策間の JS Divergence による相違度をクリップする．そして最大値 $js_{\max}$ から減算する．これにより下位方策間が離れているほど 0 に近く，下位方策間が類似しているほど高い値となる．また L_{JSdiv} を組み込んだ下位方策の損失関数を式 (3) に示す．

$$L_{\mathrm{sub}} = L^i_{\mathrm{s}} + \alpha * L^i_{\mathrm{JSdiv.}} \quad (3)$$

ここで，L^i_{s} は下位方策の損失関数，α は L_{JSdiv} に対する係数である．下位方策学習時に JS Divergence Loss を加算し学習することで，下位方策間を分離し役割を異なる下位方策獲得を促す．

4. 評価実験

AntBandits を用いた実験により，提案手法の有用性を確認する．

4.1. 学習環境

AntBandits は上/右どちらかに生成されるゴールへ向け Ant を制御するタスクである．観測情報は Ant の座標や関節角度等 (27 次元) であり，制御値は Ant8 関節のトルクである．報酬は，エージェントとゴール地点までの距離を負の報酬として与えた．エピソード終了条件は，環境が 2000 ステップ経過した場合である．AntRightUp は，生成されるゴールを右上としゴールへ向け Ant を制御するタスクである．報酬は，エージェントがゴール地点に到達している場合のみ +1 与える．観測情報とエピソード終了条件は AntBandits と同様である．

4.2. 実験概要

AntBandits タスクにて MLSH と提案手法をそれぞれ下位方策 240,000 ステップ，上位方策 400,000 ステップ学習する．獲得した下位方策の相違度の比較と可視化を行い，その傾向を比較する．その後，獲得した下位方策を用いて AntRightUp タスクを上位方策のみ 60,000 ステップ学習する．そして，報酬値の比較を行い，獲得した下位方策の有用性を検証する．本実験では下位方策を 4 つ，JS Divergence の学習係数 α を 0.5，$js_{\max}$ を 5.0 として学習する．学習アルゴリズムには Proximal Policy Optimization[2] を用いる．

4.3. 実験結果

表 1 に JS Divergence による下位方策の相違度，図 2 にその可視化結果，表 2 に AntRightUp タスクに対するゴール回数と最大報酬値を示す．表 1 より，提案手法は MLSH よりも相違度が 10 倍程度高い．また，図 2 から，MLSH は上に行く下位方策が 2 つ獲得されていることが分かる．一方で提案手法は，停止，上，右，右下と異なる下位方策である．また，表 2 から，提案手法は MLSH よりも 3 倍ゴールに到達した．

表 1: 下位方策の相違度比較

	下位方策 1	下位方策 2	下位方策 3	下位方策 4	平均値
MLSH	521.4	776.4	779.6	893.3	742.6
提案手法	7999.0	5261.3	7582.0	9574.3	7604.2

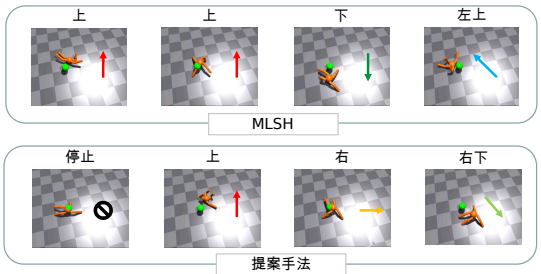

図 2: 下位方策の可視化結果

表 2: AntRightUp のゴール回数と最大報酬値

	ゴール到達回数	最大報酬
MLSH	6 回	7.0
提案手法	18 回	11.0

これらの結果から，類似した下位方策を学習する MLSH の問題点を解決することができた．

5. おわりに

本研究では，役割の異なる下位方策獲得を目的とし，MLSH に JS div.Loss を組み込む手法を提案した．実験結果として，類似した下位方策が学習されることを抑制できた．今後は，他環境での実験や方策数を変化させた際に JS div.Loss がどの程度有用であるか調査を行う．

参考文献

[1] K. Frans, *et al.*, “Meta Learning shared hierarchies-meta”, ICLR, 2018.

[2] J. Schulman, *et al.*, “Proximal Policy Optimization Algorithms”, *arXiv preprint arXiv:1707.06347*, 2017.