Off-Policy Actor-Criticアルゴリズムによる強化学習

○木村 元,　小林 重信
東京工業大学 大学院総合理工学研究科

Reinforcement Learning by Off-Policy Actor-Critic Algorithms

○ Hajime Kimura,　Shigenobu Kobayashi
Interdisciplinary Graduate School of Science and Eng., Tokyo Institute of Technology

Abstract: In real-robot applications, a robot is often required to obtain control rules for multiple tasks respectively over continuous state-action space. Off-policy learning techniques enable the learner to solve the reinforcement learning problems efficiently for multiple tasks on the same robot. Q-learning is one of the simplest off-policy learning method, however, it is computationally undesirable to apply to continuous action space. Policy gradient methods can easily be applied to continuous state-action and continuing (not episodic) tasks, and also it can improve policies under some class of POMDP environments. However, many gradient methods are limited to on-policy learning. Importance sampling techniques are promising approach to the policy gradient methods making use of all experiences for the off-policy learning, but it is limited to only episodic tasks. In this paper, we develop a gradient-based actor-critic algorithm combining with importance sampling for continuing (not episodic) tasks. We demonstrate it through simulations of a single-legged robot problem.

1　はじめに

強化学習問題においては，しばしば同一環境中で異なる複数のタスクを学習する場面がある．例えば，同一迷路においてタスク毎にゴール位置が異なったり，4足ロボットにおいて前進タスクだけでなく旋回や横歩きなどを学習することが考えられる．ある試行のもとでの経験を全てのタスク学習へ利用できれば，試行回数を削減できる．経験を共有するテクニックの一つに，off-policy 学習がある．Q-learning はその代表例だが，連続な行動空間の扱いが困難であるという問題がある．Actor-Critic や政策勾配法は，パラメータ化された確率的政策を学習するものであり，連続値の行動空間の扱いが容易だが，多くの場合 on-policy 学習に限定されている．Peshkin や Shelton らは，重点サンプリング (importance sampling) を政策勾配法に適用し，off-policy 学習を可能にした [8][5]．ただしタスクは episodic な場合に限られる．本研究では，重点サンプリングによる政策勾配法を，episodic なタスク以外に終状態のないタスク (continuing task) にも適用できるよう拡張する．

2　問題の定式化

状態，行動，実数の各集合をそれぞれ $\mathcal{S}$，$\mathcal{A}$，$\mathcal{R}$ とする．各時刻 t において学習主体のエージェントは状態 $s_t \in \mathcal{S}$ を観測し，行動 $a_t \in \mathcal{A}$ を実行すると環境の状態遷移結果として報酬 $r_t \in \mathcal{R}$ を受け取る．一般に強化学習でよく扱われるマルコフ決定過程 (MDP) では報酬や遷移先状態は確率的で，その分布は s_t および a_t のみに依存する．MDP では次の状態 s_{t+1} は遷移確率 $T(s_{t+1}|s_t, a)$ に従って選ばれ，報酬 r_t は期待値関数 $r(s_t, a)$ に従って与えられる．

エージェントは事前に $T(s_{t+1}|s_t, a)$ および $r(s_t, a)$ については知識を持たない．強化学習の学習目標は，エージェントの挙動を最大化する政策を形成することである．政策 π は状態が与えられたもとでの行動の確率分布として定義される．有限あるいは無限期間における自然な評価規範は，以下の割引報酬合計である：

$$V_t = \sum_{k=t}^{T} \gamma^{k-t} r_{t+k} \ , \tag{1}$$

割引率 γ ただし $0 \leq \gamma < 1$ は将来の報酬の重みを割り引くパラメータで，V_t は時刻 t の価値関数を表す．MDP では，政策 π のもとでの価値関数は状態の関数として定義される：

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, \pi \right], \tag{2}$$

ただし $E\{\cdot\}$ は期待値操作を表す．MDP のもとでの学習目標は，式 2 で定義される価値関数を各状態 s で最大化する政策を求めることである．

標準的な MDP の定式化では，報酬はスカラーで定義されるが，本論文ではベクトルとして与え，報酬ベクトルの各要素の値をそれぞれ別のタスクの報酬として与える．エージェントは複数の政策を持ち，各タスクへは対応した政策が割り当てられる．

3　重点サンプリング

重点サンプリングは，ある確率分布のもとでの何らかの推定値を別の分布からサンプルされたデータを用いて推定するための統計学的なテクニックである．本論文ではターゲット政策 π (target policy) の評価値を推定するのに行動政策 π' (behavior policy) のもとでの経験を用いる．長さ $T+1$ の時系列を $H = \{< s_0, a_0, r_0, \ldots s_{T-1}, a_T, r_T, s_{T+1} >\}$ とおき，初期状態 $s_0 = s$ の履歴を $h(s) \in H$ と表す．その割引報酬合計 $R(h)$ を以下で定義する：$R(h) = \sum_{t=0}^{T} \gamma^t r_t$．行動政策 π' のもとで行動選択を行って履歴をサンプルすることを考える．標準的な重点サンプリング法 [8][5] を用

いると，1つの時系列からターゲット政策のもとでの状態評価値を推定するための計算は以下で与えられる：

$$\begin{aligned}\tilde{V}^\pi(s) &= R(h)\frac{\Pr(h|\pi)}{\Pr(h|\pi')} = R(h)\frac{\pi_0\pi_1\cdots\pi_T}{\pi'_0\pi'_1\cdots\pi'_T} \\ &= \left(r_0+\gamma r_1+\cdots\gamma^T r_T\right)\frac{\pi_0\pi_1\cdots\pi_T}{\pi'_0\pi'_1\cdots\pi'_T},\end{aligned}$$

ただし π_t は政策 π のもとで時刻 t での状態で行動 a_t を実行する確率（またはその密度）を示し，r_t は時系列 h 中の時刻 t における報酬を表す．直観的に，報酬 r_t における尤度比 $\frac{\pi_0\pi_1\cdots\pi_T}{\pi'_0\pi'_1\cdots\pi'_T}$ は，時刻 t 以降の未来には依存すべきではない．この考えに基づき，per-decision 重点サンプリング [6] による計算式は以下で与えられる：

$$\begin{aligned}\tilde{V}^\pi(s) &= \left(r_0\frac{\pi_0}{\pi'_0}\right)+\left(\gamma r_1\frac{\pi_0\pi_1}{\pi'_0\pi'_1}\right)+\cdots \\ &\quad+\left(\gamma r_{T-1}\frac{\pi_0\pi_1\cdots\pi_T}{\pi'_0\pi'_1\cdots\pi'_T}\right) \\ &= \sum_{t=0}^{T}\gamma^t r_t\prod_{k=0}^{t}\frac{\pi_k}{\pi'_k} \qquad (3)\end{aligned}$$

これらの値はバイアスが無い，つまり $\mathrm{E}\{\tilde{V}^\pi(s)\}=V^\pi(s)$ ただし $T\to\infty$ であることが示されている．MDP では式 2 で定義される状態評価値は以下の Bellman 方程式を満たす：

$$\begin{aligned}V^\pi(s) &= \sum_{s\in\mathcal{S}}\sum_{a_t\in\mathcal{A}}\pi_t\left(r(s_t,a)+\gamma T(s|s_t,a)V^\pi(s)\right) \\ &= \sum_{s\in\mathcal{S}}\sum_{a_t\in\mathcal{A}}\frac{\pi_t}{\pi'_t}\pi'_t\left(r(s_t,a)+\gamma T(s|s_t,a)V^\pi(s)\right) \\ &= \mathrm{E}_{\pi'}\left\{\frac{\pi_t}{\pi'_t}\left(r_t+\gamma V^\pi(s_{t+1})\right)\right\}\ , \qquad (4)\end{aligned}$$

ただし $\mathrm{E}_{\pi'}$ は政策 π' に従って行動選択したときの期待値操作を表す．この結果より，重点サンプリングに対応した TD 法による学習則を以下で与えることができる：

$$\begin{aligned}\mathrm{TD\_err}_t &= \frac{\pi_t}{\pi'_t}\left(r_t+\gamma\hat{V}^\pi(s_{t+1})\right)-\hat{V}^\pi(s_t)\ , \qquad (5)\\ \hat{V}(s_t) &\leftarrow \hat{V}(s_t)+\alpha\ \mathrm{TD\_err}_t,\end{aligned}$$

ただし $\hat{V}(s)$ は critic で推定されている状態評価値である．

4 政策勾配

エージェントの学習目標である目的関数 $\rho(\pi)$ について，以下のように定式化する：$\rho(\pi)=V^\pi(s_0)$，$Q^\pi(s,a)=E\{V_t|s_t=s,a_t=a,\pi\}$．ここで $d^\pi(s)=\sum_{t=0}^{\infty}\gamma^t\Pr(s_t=s|s_0,\pi)$ とすると，政策勾配定理 [11] より以下の式が与えられる：

$$\frac{\partial\rho}{\partial\theta}=\sum_{s\in\mathcal{S}}d^\pi(s)\sum_{a\in\mathcal{A}}\frac{\partial\pi(s,a)}{\partial\theta}(Q^\pi(s,a)-\hat{V}^\pi(s))\ . \qquad (6)$$

環境の MDP にはエルゴード性を有するものとし，初期状態は政策 π のもとでの定常状態分布 $o^\pi(s)$ で与えられるとき，$\rho(\pi)=\sum_{s\in\mathcal{S}}o^\pi(s)V^\pi(s)$ かつ $d^\pi(s)=\frac{1}{1-\gamma}o^\pi(s)$，よって式 6 は以下のようになる：

$$\frac{\partial\rho}{\partial\theta}=\sum_{s\in\mathcal{S}}\frac{o^\pi(s)}{1-\gamma}\sum_{a\in\mathcal{A}}\frac{\partial\pi(s,a)}{\partial\theta}(Q^\pi(s,a)-\hat{V}^\pi(s))\ . \qquad (7)$$

エージェントの行動政策がターゲット政策 π と同一か，あるいは行動選択確率分布関数が状態間で独立の場合，強化学習エージェントは以下の式に従って政策を改善できる：

$$\Delta\theta=\sum_{a\in\mathcal{A}}\frac{\partial\pi(s_t,a)}{\partial\theta}(Q^\pi(s_t,a)-\hat{V}^\pi(s_t)) \qquad (8)$$

各状態について計算する．これは以下の式に変形できる：

$$\begin{aligned}&\sum_{a\in\mathcal{A}}\frac{\partial\pi(s_t,a)}{\partial\theta}(Q^\pi(s_t,a)-\hat{V}^\pi(s_t)) \\ &= \sum_{a\in\mathcal{A}}\frac{\partial\pi(s_t,a)}{\partial\theta}Q^\pi(s_t,a)-0 \\ &= \sum_a\sum_s\frac{\partial\pi(s_t,a)}{\partial\theta}\left(r(s_t,a)+\gamma T(s|s_t,a)V^\pi(s)\right) \\ &= \sum_a\sum_s\frac{1}{\pi(s_t,a)}\frac{\partial\pi(s_t,a)}{\partial\theta}\frac{\pi(s_t,a)}{\pi'(s_t,a)}\pi'(s_t,a) \\ &\quad\times\left(r(s_t,a)+\gamma T(s|s_t,a)V^\pi(s)\right) \\ &= E_{\pi'}\left\{\frac{\partial\ln\pi_t}{\partial\theta}\left(\frac{\pi_t}{\pi'_t}\left(r_t+\gamma V^\pi(s_{t+1})\right)+\mathrm{const.}\right)\right\} \\ &= E_{\pi'}\left\{\frac{\partial\ln\pi_t}{\partial\theta}\left(\frac{\pi_t}{\pi'_t}\left(r_t+\gamma V^\pi(s_{t+1})\right)-\hat{V}^\pi(s_t)\right)\right\}\end{aligned} \qquad (9)$$

ただし $\hat{V}^\pi(s_t)$ は行動 a_t とは条件付独立であると仮定する．一般に，式 9 中の真の評価値 $V^\pi(s_{t+1})$ についてはエージェントにとって未知である．真の評価値 $V^\pi(s_{t+1})$ を推定値 $\hat{V}^\pi(s_{t+1})$ へ置き換えることにより，式 5 に示された TD エラーを利用した重点サンプリング版の actor-critic アルゴリズムを得る．また，真の評価値 $V^\pi(s_{t+1})$ を式 3 で示された推定値 $\tilde{V}^\pi(s_{t+1})$ に置き換えることにより，別のバージョンの重点サンプリング版 actor-critic アルゴリズムにもなる．

5 重点サンプリング版 Actor-Critic 法

5.1 アルゴリズムの提案

前章で示した政策勾配に基づいた actor-critic アルゴリズムを提案する．Fig.1 はその概要である．行動政策 π' がターゲット政策 π と同一のとき，このアルゴリズムは従来手法 [3] と同一になる．ターゲット政策は複数保持できるが，各政策毎に政策パラメータベクトル θ とその適正度の履歴 (eligibility trace)$\bar{e}(t)$ のためのメモリーを要する．政策パラメータベクトル θ は $\Delta\theta_t$ の合計を用いて適切に更新される：例えば各時刻において $\theta\leftarrow\theta+\alpha_p\frac{\Delta\theta_t}{|\Delta\theta_t|}$ のように更新することが考えられる．ただし α_p は小さな正の定数である．

1. 状態 s_t を観測し，行動政策の確率（または確率密度）$\pi'_t = \pi'(a_t, s_t)$ に従って行動 a_t を選択して実行する．その結果，報酬 r_t と遷移先の状態 s_{t+1} を観測する．

2. ターゲット政策 π が状態 s_t のもとで行動 a_t を選択する確率（または確率密度）を計算する．

3. ターゲット政策のパラメータ θ についての適正度を計算：

$$\begin{aligned} e_t &= \frac{\partial}{\partial\theta}\ln\pi_t \ , \\ \bar{e}_t &= e_t + \gamma\,\lambda_p\,\frac{\pi_t}{\pi'_t}\bar{e}_{t-1} \ , \end{aligned}$$

ただし γ は報酬の割引率，λ_p は適正度の履歴の割引率で $(0 \le \lambda_p \le 1)$ である．

4. 以下のように TD_error を計算して critic における状態評価値を更新する：

$$\begin{aligned} \mathrm{TD\_err}_t &= \frac{\pi_t}{\pi'_t}\left(r_t + \gamma\hat{V}^\pi(s_{t+1})\right) - \hat{V}^\pi(s_t) \ , \\ \hat{V}^\pi(s_t) &\leftarrow \hat{V}^\pi(s_t) + \alpha\mathrm{TD\_err}_t \ , \end{aligned}$$

ただし α は学習率である．

5. 政策勾配の更新値を以下のように計算：

$$\Delta\theta_t = \bar{e}_t\ \mathrm{TD\_err}_t \ ,$$

6. 時刻を $t \leftarrow t+1$ に進めて step 1 より繰返す．

Figure 1: 重点サンプリングを利用した actor-critic アルゴリズムによる政策勾配の推定法

5.2 提案手法の解析

学習パラメータ $\lambda_p = 0$ と仮定する．このとき値 $\Delta\theta_t$ は

$$\begin{aligned} \Delta\theta_t &= e_t\left(\frac{\pi_t}{\pi'_t}\left(r_t + \gamma\hat{V}^\pi(s_{t+1})\right) - \hat{V}^\pi(s_t)\right) \ , \quad (10) \\ &= \frac{\partial\ln\pi_t}{\partial\theta}\left(\frac{\pi_t}{\pi'_t}\left(r_t + \gamma\hat{V}^\pi(s_{t+1})\right) - \hat{V}^\pi(s_t)\right) \ . \end{aligned}$$

よって $\Delta\theta_t$ の期待値は，真の状態評価地値 $V^\pi(s_{t+1})$ が推定値 $\hat{V}^\pi(s_{t+1})$ で置き換えられている点を除いては式 9 の政策勾配に類似している。この場合，critic で推定される状態価値関数を用いて政策勾配が推定される．

$\lambda_p = 1$ のとき，$\Delta\theta_t$ は以下のように展開できる：
$t = 0$ のとき，

$$\Delta\theta_0 = \frac{\pi_0}{\pi'_0}e_0 r_0 + \gamma\frac{\pi_0}{\pi'_0}e_0\hat{V}^\pi(s_1) - \hat{V}^\pi(s_0)e_0$$

$t = 1$ のとき, $\Delta\theta_1 =$

$$\gamma\frac{\pi_0\pi_1}{\pi'_0\pi'_1}e_0 r_1 + \gamma^2\frac{\pi_0\pi_1}{\pi'_0\pi'_1}e_0\hat{V}^\pi(s_2) - \gamma\hat{V}^\pi(s_1)\frac{\pi_0}{\pi'_0}e_0$$
$$+\frac{\pi_1}{\pi'_1}e_1 r_1 + \gamma\frac{\pi_1}{\pi'_1}e_1\hat{V}^\pi(s_2) - \hat{V}^\pi(s_1)e_1$$

$t = 2$ のとき，

$$\begin{aligned} \Delta\theta_2 = {} & \gamma^2\frac{\pi_0\pi_1\pi_2}{\pi'_0\pi'_1\pi'_2}e_0 r_2 + \gamma^3\frac{\pi_0\pi_1\pi_2}{\pi'_0\pi'_1\pi'_2}e_0\hat{V}^\pi(s_3) \\ & -\gamma^2\hat{V}^\pi(s_2)\frac{\pi_0\pi_1}{\pi'_0\pi'_1}e_0 \\ & +\gamma\frac{\pi_1\pi_2}{\pi'_1\pi'_2}e_1 r_2 + \gamma^2\frac{\pi_1\pi_2}{\pi'_1\pi'_2}e_1\hat{V}^\pi(s_3) - \gamma\hat{V}^\pi(s_2)\frac{\pi_1}{\pi'_1}e_1 \\ & +\frac{\pi_2}{\pi'_2}e_2 r_2 + \gamma\frac{\pi_2}{\pi'_2}e_2\hat{V}^\pi(s_3) - \hat{V}^\pi(s_2)e_2 \end{aligned}$$

$\Delta\theta_t$ を $t = 0$ から T まで合計すると，隣り合ういくつかの項はキャンセルされて以下を得る：

$$\begin{aligned} \sum_{t=0}^{T}\Delta\theta_t = {} & e_0\Big(\frac{\pi_0}{\pi'_0}r_0 + \gamma\frac{\pi_0\pi_1}{\pi'_0\pi'_1}r_1 + \cdots \\ & +\gamma^T\frac{\pi_0\cdots\pi_T}{\pi'_0\cdots\pi'_T}r_T - \hat{V}^\pi(s_0)\Big) \\ & +e_1\Big(\frac{\pi_1}{\pi'_1}r_1 + \gamma\frac{\pi_1\pi_2}{\pi'_1\pi'_2}r_2 + \cdots \\ & +\gamma^{T-1}\frac{\pi_1\cdots\pi_T}{\pi'_1\cdots\pi'_T}r_T - \hat{V}^\pi(s_1)\Big) \\ & \cdots \\ & +e_T\left(\frac{\pi_T}{\pi'_T}r_T - \hat{V}^\pi(s_T)\right) \\ = {} & \sum_{t=0}^{T}e_t\left(\tilde{V}^\pi(s_t) - \hat{V}^\pi(s_t)\right) \qquad (11) \\ = {} & \sum_{t=0}^{T}\frac{\partial\ln\pi_t}{\partial\theta}\left(\frac{\pi_t}{\pi'_t}\left(r_t + \tilde{V}^\pi(s_{t+1})\right) - \hat{V}^\pi(s_t)\right) \end{aligned}$$

よって $\lambda_p = 1$ のとき，提案手法の $\Delta\theta_t$ の合計の期待値は真の状態評価地値 $V^\pi(s_{t+1})$ が式 3 で定義された推定値 $\tilde{V}^\pi(s_t)$ で置き換えられている点を除いては式 9 の政策勾配に類似している。この場合，政策勾配はサンプル (actual return) から推定されるが，critic で推定される値は使われない．Critic で推定される状態評価値 $\hat{V}^\pi(s)$ は政策勾配を推定する際に分散を低減させる効果がある．

5.3 提案手法の特徴

- 尤度比 $\frac{\pi_0\pi_1\cdots\pi_T}{\pi'_0\pi'_1\cdots\pi'_T}$ は各時間ステップにおいて逐次的に計算され，エージェントは状態-行動-報酬の時系列を記憶しておく必要がない．
- 行動選択が従う行動政策 (behavior policy) は一つだが、ターゲット政策は複数同時に学習できる．報酬をベクトル形式で与え，ベクトルの各要素を各タスクへ対応させ，複数のターゲット政策をそれぞれ各タスクへ割り当てて重点サンプリングによって学習することにより，１つの経験を全タスクの学習へ生かすことができる．特に学習初期において異なるタスクを学習するための複数のターゲット政策が皆同じようなランダム政策からスタートする場合、全ての政策で経験を共有できるので学習時間が政策の個数分の１倍に短縮される

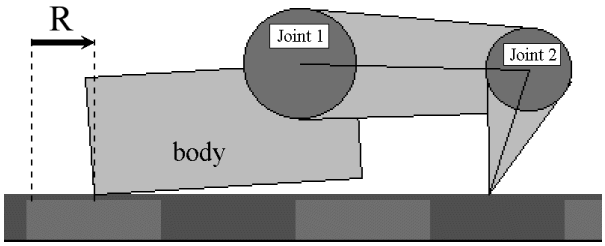

Figure 2: 自由度 2 のアームロボット．状態-行動空間は連続.

- しかし学習が進行して，同一状態における行動選択確率が各タスク間で大きく異なってくると，尤度比 π'/π はゼロに近づき，経験の共有ができなくなる．
- 状態訪問頻度の問題：本手法の導出の基本となった式 8 によって学習を行うためには、行動政策がターゲット政策 π に等しいか，あるいは政策パラメータが各状態毎に独立であるという条件が必要である．これは、continuing task 特有で、episodic task には見られない特徴である．重点サンプリングにおいて行動政策がターゲット政策 π に等しいという条件はナンセンスである．よって政策パラメータが各状態毎に独立でなければならないが，これは関数近似による状態汎化や状態観測の不完全性への対処の能力において，on-policy 学習に比べると問題があることを示している．

6 関連研究

Off-policy TD 学習についての研究 [6][7]，POMDP の環境において重点サンプリングによる off-policy 学習で政策勾配を推定して政策改善を行う手法 [8][5] が提案されている．Intra-option learning [10] は Q-learning に基づく off-policy 学習によって複数のサブタスクをそれぞれ同時に学習できる．On-policy actor-critic アルゴリズムの収束について解析も示されている [4]．GPOMDP アルゴリズム [1] は，終端状態のない POMDP の環境において平均報酬を評価関数とした政策パラメータについての勾配を on-policy で推定する．GPOMDP では平均報酬を近似するため，環境の MDP の有している mixing time に関連する割引率を注意深く選びながら割引報酬についての勾配を利用している．ATPG アルゴリズム [2] では政策を改善するために，TD エラーに相当するような，行動実行時の相対評価値を推定するが，終端状態のあるタスクの学習に限られる．

7 実験

Fig.2 に示すロボットに本手法を適用する．学習目標は，前進する（タスク０）または後退する（タスク１）ために，アームを足のように作用させる動作の獲得である．関節は位置制御のサーボモーターによって角度を制御される．各時間ステップにおいて、エージェントは２つの関節モータの角度および足先のタッチセンサーの状態という３つの状態量を要素としを３次元ベクトルで観測する：(ϕ_1, ϕ_2, ϕ_3) ただし各要素は $[0, 1]$ の範囲で正規化される．観測に応じて行動政策に基づき行動を選択する．関節角度 ϕ_1, ϕ_2 は２次元連続空間で定義され，タッチセンサ ϕ_3 は 0 または 1 の２値である．行動は，関節角度の目標値を指示し，２次元ベクトル $(a_{(1)}, a_{(2)})$ の各要素がそれぞれ関節角度を表す．行動ベクトルの各要素は $[-1, 1]$ の範囲に限定される．行動が選ばれると，モータは指示された目標位置へ動きはじめる．関節角度が指示された位置まで動くか，あるいはタッチセンサの値が変化すると，状態遷移の結果として報酬が与えられ，次の時刻へ進む．関節のモータが目標位置まで動く途中でセンサの値が変化すると，そこで意思決定イベントが発生して動きが打ち切られるため，次のステップでの関節角度は行動として出力された目標角度には一致しない．よって状態遷移には不確実性が存在する．タスク０では，報酬はボディが前進した距離と方向で与えられる．ロボットが後退した場合，タスク０の政策は負の報酬を受け取る．タスク１では，全ての報酬は単にタスク０の報酬の符号を反転させて与える．このシミュレータプログラムのコードは Java で記述され，下記 URL より入手できる：`http://www.fe.dis.titech.ac.jp/indexj.html`.

状態汎化の影響を調べるため，状態表現のための特徴ベクトルを２種類用意した．一つは線形コーディングで，$X = (\phi_1, \phi_2, \phi_3, 1-\phi_1, 1-\phi_2, 1-\phi_3)$ で与えられる６次元の連続ベクトルである．どんな状態においてもノルムは一定に保たれる．もう一つは均一タイルコーディングで，状態空間を３次元のタイルで $8 \times 8 \times 2$ に分割する．Critic における状態評価関数の推定値 $\hat{V}^\pi(s)$ は線形関数近似 [9] を用いた．

政策関数はコーシー分布に基づく連続分布関数を修正して用いた．それは行動 $(a_{(1)}, a_{(2)})$ の各要素が $[-1, 1]$ の範囲に限定されていることによる．エージェントは各モータ i 毎に式 12 のコーシー分布に従って，$[-1, 1]$ の範囲に収まっているサンプルを得るまでランダムにサンプリングを行い，得たサンプルを行動 $a_{(i)}$ として出力する．

$$P(a) = \frac{1}{\pi\sigma_{(i)}} \frac{1}{1 + \frac{(a-\mu_{(i)})^2}{\sigma_{(i)}^2}}. \tag{12}$$

ここでパラメータ $\mu_{(i)}$ および $\sigma_{(i)}$ は以下で与えられる：

$$\mu_{(i)} = \frac{2}{1 + \exp\left(-\sum_j x_j\, \theta_{j,(i)}\right)} - 1, \tag{13}$$

$$\sigma_{(i)} = \frac{1}{1 + \exp\left(-\theta_{\sigma,(i)}\right)}, \tag{14}$$

ただし $\theta_{j,(i)}$ および $\theta_{\sigma,(i)}$ は政策パラメータである．$\theta_{j,(i)}$ 中の j は，特徴ベクトル X の要素 x_j に対応付けられる．このとき政策関数 $\pi(s, a)$ は以下で表される：

$$\pi(s, a_{(i)}) = \frac{1}{S_i\sigma_{(i)}} \frac{1}{1 + \frac{(a-\mu_{(i)})^2}{\sigma_{(i)}^2}}, \tag{15}$$

ただし $\pi(s, a) = \prod_i \pi(s, a_{(i)})$ で，S は以下で与えられる：

$$S_i = \int_{-1}^{1} \frac{1}{\sigma_{(i)}} \frac{1}{1 + \frac{(a-\mu_{(i)})^2}{\sigma_{(i)}^2}} da$$

$$= \tan^{-1}\left(\frac{1-\mu_{(i)}}{\sigma_{(i)}}\right) - \tan^{-1}\left(\frac{-1-\mu_{(i)}}{\sigma_{(i)}}\right). \quad (16)$$

このとき $\mu_{(i)}$ と $\sigma_{(i)}$ の適正度は以下で計算される：

$$\frac{\partial \ln \pi(s,a_{(i)})}{\partial \mu_{(i)}} = \frac{1}{\sigma_{(i)}}\left(\frac{1}{S_i}\left(\frac{1}{1+\left(\frac{1-\mu_{(i)}}{\sigma_{(i)}}\right)^2} - \frac{1}{1+\left(\frac{-1-\mu_{(i)}}{\sigma_{(i)}}\right)^2}\right) + 2(a_{(i)}-\mu_{(i)})S_i\pi(s,a_{(i)})\right), \quad (17)$$

$$\frac{\partial \ln \pi(s,a_{(i)})}{\partial \sigma_{(i)}} = \frac{1}{\sigma_{(i)}^2 S_i}\left(\frac{1-\mu_{(i)}}{1+\left(\frac{1-\mu_{(i)}}{\sigma_{(i)}}\right)^2} - \frac{-1-\mu_{(i)}}{1+\left(\frac{-1-\mu_{(i)}}{\sigma_{(i)}}\right)^2}\right) - \left(1-\frac{(a_{(i)}-\mu_{(i)})^2}{\sigma_{(i)}^2}\right) S_i\pi(s,a_{(i)}) \quad (18)$$

式 13, 13, 17, 18 より，政策パラメータの適正度は：

$$\frac{\partial \ln \pi(s,a_i)}{\partial \theta_{j,(i)}} = \frac{\partial \mu_{(i)}}{\partial \theta_{j,(i)}}\frac{\partial \ln \pi(s,a_{(i)})}{\partial \mu_{(i)}} = \frac{x_j}{2}(1+\mu_{(i)})(1-\mu_{(i)})\frac{\partial \ln \pi(s,a_{(i)})}{\partial \mu_{(i)}} \quad (19)$$

$$\frac{\partial \ln \pi(s,a_i)}{\partial \theta_{\sigma,(i)}} = \frac{\partial \sigma_{(i)}}{\partial \theta_{\sigma,(i)}}\frac{\partial \ln \pi(s,a_{(i)})}{\partial \sigma_{(i)}} = \sigma_{(i)}(1-\sigma_{(i)})\frac{\partial \ln \pi(s,a_{(i)})}{\partial \sigma_{(i)}} \quad (20)$$

本実験では，Fig.1 中の $e(t)$ は式 19,20 で計算される．全ての実験において割引率 $\gamma = 0.9$，actor の学習率 $\alpha_p = 0.02$ を用いた．

学習パラメータは 3 種類の設定で実験を行った．1 つは actor の適正度の履歴を使わず式 10 に従った学習で，$\alpha = 0.1$ および $\lambda_p = 0$ に設定した（Fig.3）．2 番目設定は，actor の適正度の履歴を使う式 11 による学習で，$\alpha = 0.1$ および $\lambda_p = 1$ に設定した（Fig.4）．3 番目の設定は，actor の適正度の履歴を使う式 11 による学習だが，critic を全く機能させない設定である．$\alpha = 0$ および $\lambda_p = 1$ に設定した（Fig.5）．

Critic を用いる場合，どちらの設定においても，重点サンプリングを用いたアルゴリズムでは，線形コーディングによる許容可能な政策は獲得できなかった．驚いたことに $\lambda_p = 1$ の on-policy actor-critic だけが線形コーディングで学習できた．

タイルコーディングを用いた場合，重点サンプリングを用いたほうが通常の on-policy の actor-critic よりも学習初期において高速に学習が進行する．しかしながら，さらに学習が進行すると政策の質が悪化する．Critic を使用しない方法 ($\alpha = 0$，$\lambda_p = 1$) では全ての設定において学習できたが，設定間で目立った差は見られず，学習は極めて遅い．

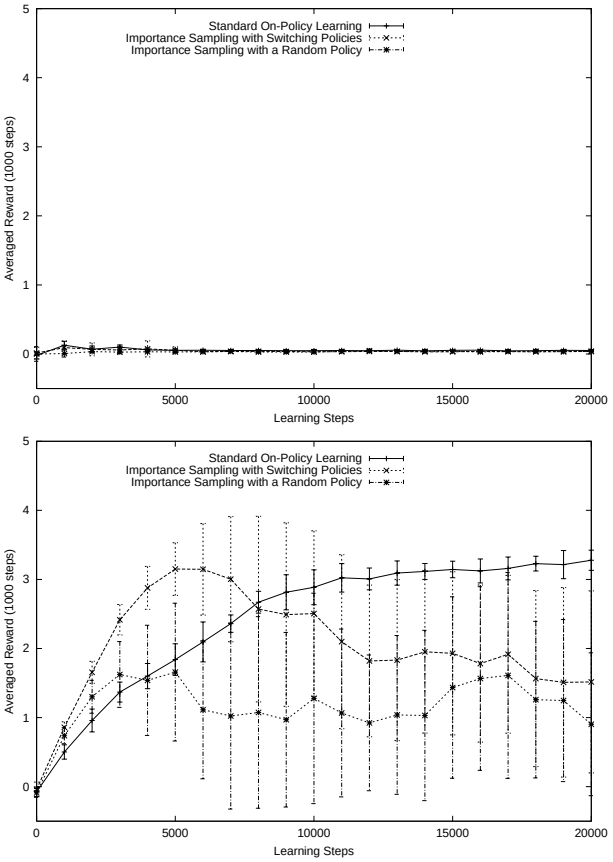

Figure 3: タスク 0 における 10 試行平均の学習の様子．パラメータ設定は $\alpha = 0.1$ および $\lambda_p = 0$．上側のグラフは線形コーディング，下側はタイルコーディングを用いた場合．

8 考察とまとめ

本論文では，終端状態のないタスクにおいて，重点サンプリングを用いる actor-critic 法を提案した．本手法は連続な行動空間を有する問題における off-policy 政策改善法として有望である．特に学習初期に全ての政策が類似している場合，重点サンプリングによって経験を共有する効果があることを示した．提案手法は状態汎化と同時に用いると問題が生じることを示した．この問題点の解決策の 1 つとして，on-policy の場合と off-policy の場合における政策勾配の違いを利用して，混同している状態を検出・判別することが考えられる．また別の解決策として，on-policy の場合と off-policy の場合の状態訪問頻度分布の比率を各状態毎に推定して，更新値を補正することも考えられる．実験では，1 本腕 (足) ロボットにおいて連続値行動の範囲に制限がある場合，コーシー分布を利用して実装する方法を示した．提案手法の実機への適用は今後の課題である．

References

[1] Baxter, J. & Bartlett, P.L.: Reinforcement learning in POMDP's via direct gradient ascent, *Proceedings of the 17th International Conference on Machine Learning*, pp.41–48 (2000).

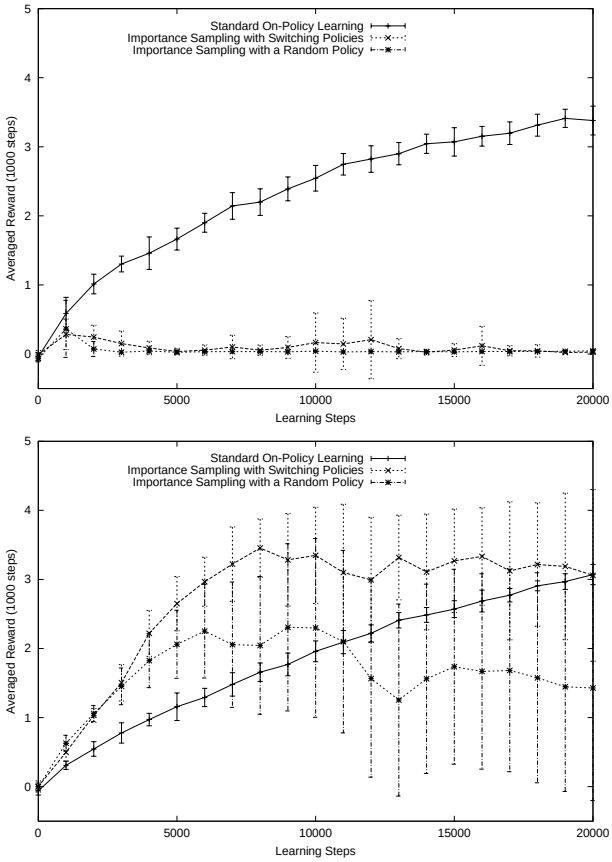

Figure 4: タスク0における10試行平均の学習の様子．パラメータ設定は $\alpha = 0.1$ および $\lambda_p = 1$．上側のグラフは線形コーディング，下側はタイルコーディングを用いた場合．

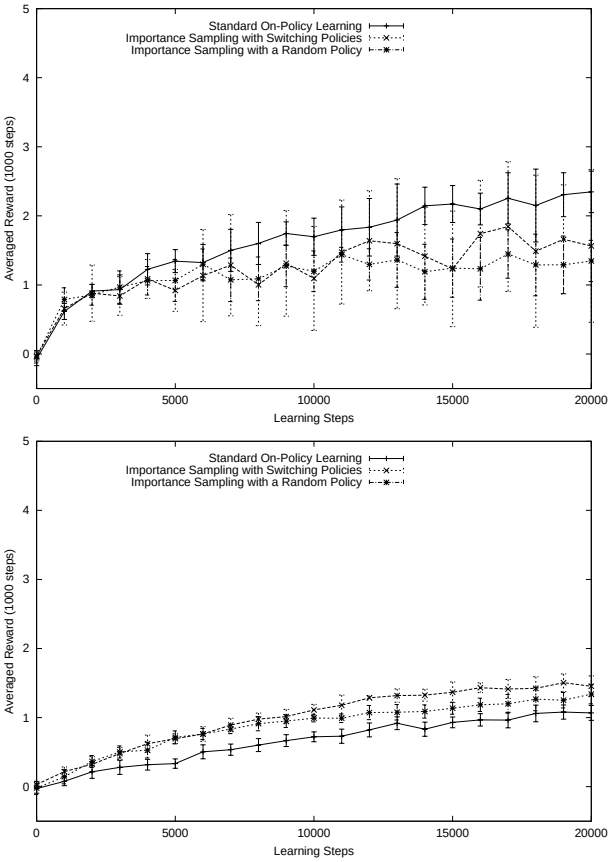

Figure 5: タスク0における10試行平均の学習の様子．パラメータ設定は $\alpha = 0$ および $\lambda_p = 1$．上側のグラフは線形コーディング，下側はタイルコーディングを用いた場合．

[2] Grudic, G. & Unger, L.: Localizing policy gradient estimates to action transitions, *Proceedings of the 17th International Conference on Machine Learning*, pp. 343–350 (2000).

[3] 木村 元，小林 重信: Actor に適正度の履歴を用いた Actor-Critic アルゴリズム– 不完全な Value-Function のもとでの強化学習，人工知能学会誌, Vol.15, No.2, pp.267–275 (2000).

[4] Konda, V.R. & Tsitsiklis, J.N.: Actor-critic algorithms, *Advances in Neural Information Processing Systems 12*, pp. 1008–1014 (2000).

[5] Peshkin, L. & Shelton, C.R.: Learning from scarce experience, *19th International Conference on Machine Learning*, pp.498–505 (2002).

[6] Precup, D., Sutton, R.S. & Singh, S.: Eligibility traces for off-policy policy evaluation, *17th International Conference on Machine Learning*, pp.759–766 (2000).

[7] Precup, D., Sutton, R. S. & Dasgupta, S.: Off-policy temporal-difference learning with function approximation, *18th International Conference on Machine Learning*, pp.417–424 (2001).

[8] Shelton, C.R.: Policy improvement for POMDPs using normalized importance sampling, *17th Conference on Uncertainty in Artificial Intelligence*, pp.496–503 (2001).

[9] Sutton, R.S. & Barto, A.: Reinforcement learning: An introduction, *A Bradford Book*, The MIT Press (1998).

[10] Sutton, R. S., Precup, D. & Singh, S.: Intra-option learning about temporary abstract actions, *15th International Conference on Machine Learning*, pp.556–564 (1998).

[11] Sutton, R.S., McAllester, D., Singh, S. & Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation, *Advances in Neural Information Processing Systems 12*, pp. 1057–1063 (2000).

[12] Uchibe, E. & Doya, K.: Reinforcement Learning using Importance Sampling for Multiple Learning Modules, Technical report of the Institute of Electronics, Information and Communication Engineerings (in Japanese).