部分的環境変化に効率よく適応する強化学習法

林　　佑宜† 　　中村　篤祥†

† 北海道大学 工学部 情報エレクトロニクス学科　〒 060–0814 北海道札幌市北区北十四条西 9 丁目
E-mail: †{yhayashi,atsu}@ist.hokudai.ac.jp

あらまし　強化学習は逐次的意思決定問題に使われる手法であり、定常環境を仮定することが多い。しかし、現実世界では環境の定常性は保証されない。そのため、定常環境を仮定した強化学習では、環境の変化が起こるとそれへの適応に時間がかかる場合がある。そのような環境変化に対処する強化学習についての研究はいくつかあるが、環境全体の変化に適応するものが主である。環境の部分的な変化のみが発生する場合、全体を適応させるよりその部分だけを対処させた方が効率の良い適応が可能である。本論文では、そのような一部分だけ変化する環境に迅速に適応するモデルベースの強化学習の手法を提案し、実験により効果の検証を行う。

キーワード　強化学習、変化点検知、モデルベース強化学習、非定常環境

1　概　　要

強化学習 [1] は、逐次的意思決定問題を解決するために用いられる手法であり、試行錯誤しながら行動を最適化する理論的枠組みである。

強化学習では、環境が定常的であるという仮定 (状態遷移や報酬が時間と共に変化しない) を前提にすることが多い。この仮定と異なる環境では、標準的な強化学習のアルゴリズムはうまく機能しない場合がある。しかし、強化学習のアルゴリズムを現実の問題に適用する際、定常的な環境が保証されているとは限らない。そのため、環境が変化したことを検知し、適応する仕組みが必要になる。

ある環境において方策を最適化できた後、環境全体が変化する場合、最適な方策は変化する。そういった状況で現在のモデルをそのまま利用していくと、環境が変化した後の最適な方策に収束するのに時間がかかる。そのため、モデルを新しく作り最初から学習し直す方法が効率よく学習できる。しかし、環境が一部分のみ変化する場合、現在のモデルを全く利用できないわけではないため、現在のモデルを完全に初期化してしまうのでは効率が悪い。

本稿では、環境変化が全体ではなく、部分的に変化する場合の変化点検知及び環境に適応する手法を提案する。

非定常環境には様々あり、一般的な非定常環境に対処するのは非常に困難である。そのため、本論文では環境がほとんど変化せず、変化するときには突然変化する場合に限定する。また、環境の変化は部分的なものを想定する。この前提のもとで環境の変化に対応するモデルベースの強化学習手法を提案し、実験により効果の検証を行う。

2　関 連 研 究

非定常環境における強化学習の手法は様々研究されている。Choi らは非定常環境を持つ確率的なマルコフ決定過程 (MDP) のモデル Hidden-mode MDPs (HM-MDP) を提案した [2]。HM-MDP は、環境変化によりモードが変わるモデルであり、それぞれのモードは定常環境の MDP に対応する。状態空間と行動空間は全てのモデルで共通であるが、状態遷移確率と報酬関数は各モードによって異なる。また、エージェントはモデル間の遷移を観測できない。Choi らは HM-MDP を解くためのアルゴリズムとして、HM-MDP のためのベルマン方程式に基づいた価値反復アルゴリズムを考案した [3]。これはモデル情報が既知であることを前提にしている。

Silva らは非定常環境におけるモデルベースのアルゴリズムを考案した [4]。これは Reinforcement Learning with Context Detection (RLCD) と呼ばれ、コンテキスト検知を用いたアルゴリズムである。シミュレーションのサンプルによって、状態遷移確率と報酬関数の推定、MDP 環境の変化の検知を行う。エピソードごとにすべての観測済みのコンテキストのエラースコアが予測関数を用いて計算され、最小のものがアクティブなコンテキストとして選択される。ただし、すべてのコンテキストのエラースコアが設定した値よりも大きい場合は、新しいコンテキストが生成され、これがアクティブとなる。

文献 [5] では、RLCD を改良し、アクティブコンテキストの同定に逐次解析で開発された cumulative sum (CUSUM) を組み込んでいる [6]。この手法は RLCD よりも理論的な根拠を持たせることが出来る上に、必要なパラメータ数の削減に成功している。また、パラメータの解釈も容易で調整がしやすいという利点がある。

3　CUSUM を用いた RLCD 法

環境の変化点検知をしたときに現在のモデルを初期化し、環境に適応する従来手法である RLCD 法について説明する。

3.1　強化学習と MDP

MDP は $< \mathcal{S}, \mathcal{A}, T, R >$ の 4 つ組で表される。$\mathcal{S}$ は離散状態集合、$\mathcal{A}$ は離散行動集合、$R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ は報酬関数、$T : \mathcal{S} \times \mathcal{A} \to \Pi(\mathcal{S})$ は状態遷移確率を表す。状態 s において、

行動 a を選択し、状態が s' に遷移する確率を $T(s,a,s')$ と記述する。

状態が s でその後の状態遷移が方策 π に従う場合の割引報酬の総和の期待値を $V^{\pi}(s)$ と表記し、これを状態価値関数という。状態が s の時に行動 a を選択し、その後の状態遷移は方策 π に従う場合の割引報酬の総和の期待値を $Q^{\pi}(s,a)$ と表記し、これを行動価値関数という。

最適状態価値関数は $V^{*}(s)$ と表記し、ある状態 s から開始して最適方策に従った時に得られる割引報酬の総和の期待値を表している。全ての状態において、$V^{*}(s)$ が与えられたとき、将来の割引報酬の総和の期待値が最大になるような、状態から行動への写像を最適方策 π^{*} と書く。

MDP の要素全てが既知であるのならば、価値反復や方策反復のようなアルゴリズムを用いることで最適方策を求めることが出来る。しかし、状態遷移確率 T や報酬関数 R が未知である場合、強化学習の手法が用いられる。環境からの報酬のフィードバックを得ることで、最適方策を学習する。環境が定常の場合、強化学習により学習された方策は最適方策に収束することが保証されている。本論文では非定常環境を扱うため、MDP モデルや強化学習の手法を直接適用することが出来ない。

強化学習における学習方法は、一般にモデルフリーとモデルベースの二つのアプローチに分けられる。モデルフリーのアルゴリズムでは、エージェントは環境がどのように動くのかという情報を利用しない。環境から得られたサンプルを直接利用するため、誤差は小さくなるが、サンプル効率が悪く、方策が収束するまでの試行回数が多い。これに対して、モデルベースのアルゴリズムはエージェントは環境についての情報を利用する。環境を仮定して学習するため、仮定が間違っていると誤差が大きくなるが、サンプル効率が良く、収束までの時間が短い。今回は Prioritiezed Sweeping と呼ばれるモデルベースの手法を用いる [1]。

方策には、ε の確率でランダムな行動を選択し、$1-\varepsilon$ の確率で状態行動価値が最大の行動を選択する ε -greedy 法を用いる。

3.2 モデルの学習

モデルベースの手法では、状態遷移確率と報酬関数が必要になる。しかし、強化学習の問題ではこれらの値が明示的に提示されていない。そのため、状態 s で行動 a をとり、状態 s' と報酬 r を観測する度に以下のように更新することにより状態遷移確率と報酬関数を推定する [4]。

$$\Delta T(\kappa) = \begin{cases} \frac{1-T(s,a,\kappa)}{N(s,a)+1} & \kappa = s' \\ \frac{0-T(s,a,\kappa)}{N(s,a)+1} & \kappa \neq s' \end{cases} \quad \forall \kappa \in \mathcal{S} \tag{1}$$

$$T(s,a,\kappa) = T(s,a,\kappa) + \Delta T(\kappa) \tag{2}$$

$$\Delta R(s,a) = \frac{r - R(s,a)}{N(s,a)+1} \tag{3}$$

$$R(s,a) = R(s,a) + \Delta R \tag{4}$$

$N(s,a)$ は状態 s で行動 a をとった回数を表している。状態 s で行動 a をとる度に過去 M 回のみ考慮して以下のように更新する。

$N(s,a) = \min(N(s,a)+1, M)$

初期値は任意の状態、行動において、$T(s,a,\kappa) = \frac{1}{|\mathcal{S}|}, R(s,a) = 0, N(s,a) = 0$ とする。

3.3 CUSUM を用いた環境変化の検知

CUSUM [6] を用いた RLCD 法では、次のようにして、環境がいつ変化をしたのか検知する [5]。2 つの MDP を $M_0 = (\mathcal{S}, \mathcal{A}, T_0, R_0)$ と $M_1 = (\mathcal{S}, \mathcal{A}, T_1, R_1)$ とし、これらの値は既知であると仮定する。ある未知の時間ステップで環境がコンテキスト M_0 から M_1 へと変化する場合を考える。$(s_0, a_0, s_1, a_1, s_2, ..., s_t, a_t, s_{t+1})$ を観測履歴とし、それから計算される CUSUM スコアを S_t^T とする。時間ステップ $t \geqq 1$ において S_t^T は以下の式で計算する。

$$S_t^T = \max\left(0, S_{t-1}^T + \ln \frac{T_1(s_t, a_t, s_{t+1})}{T_0(s_t, a_t, s_{t+1})}\right) \tag{5}$$

ただし、$S_0^T = 0$ とする。CUSUM 法では、S_t^T の値を設定した閾値 $c^T > 0$ よりも大きいかを比較する。もし、$S_t^T \geqq c^T$ ならば遷移確率が変化したと判断する。報酬関数の変化も同様に検知できるが、今回は遷移確率の変化に限定して検知を行う。CUSUM 法の直感的な考え方としては、最近の履歴を生成した可能性 (尤度) が M_0 よりも M_1 の方が有意に高い場合、環境が変化したと判断するというものである。

4 提 案 法

文献 [4] と [5] では、環境の変化を検知して、モデルの変更を行っている。これを部分的な環境の変化を検知するように変更する。

$(s_0, a_0, s_1, a_1, s_2, ..., s_t, a_t, s_{t+1})$ を観測履歴とし、それから計算される CUSUM スコアを $S_t^T(s,a)$ とする。時間ステップ $t \geqq 1$ において、$S_t^T(s,a)$ は以下の式で計算する。

$$S_t^T(s,a) = \begin{cases} \max\left(0, S_{t-1}^T(s,a) + \ln \frac{T_{uni}(s_t, a_t, s_{t+1})}{T_{cur}(s_t, a_t, s_{t+1})}\right) & ((s,a) = (s_t, a_t)) \\ S_{t-1}^T(s,a) & ((s,a) \neq (s_t, a_t)) \end{cases} \tag{6}$$

ただし、$S_0^T(s,a) = 0$ ($\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$) とする。$T_{cur}(s,a,s')$ は現在のモデルの状態遷移確率、$T_{uni}(s,a,s')$ は一様な状態遷移確率であり、以下のように表される。

$T_{uni}(s,a,\kappa) = \frac{1}{|\mathcal{S}|}$, $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$, $\forall \kappa \in \mathcal{S}$

直近の状態遷移の割合が現在のモデルの状態遷移確率よりも、一様分布に近い場合、部分的な環境が変化したと判定する。アルゴリズムの全体像は Algorithm 1 に記述する。

Algorithm 1 partial RLCD with Sequential Change-Point Detection

Require: $c > 0$ (CUSUM score threshold), $M > 0$ (window size)

$T_{cur}(s,a,\kappa) \leftarrow \frac{1}{|\mathcal{S}|}, R(s,a) \leftarrow 0, S(s,a) \leftarrow 0\ (\forall s \in \mathcal{S},\ \forall a \in \mathcal{A},\ \forall \kappa \in \mathcal{S})$
$s \leftarrow s_0$ (initial state)
while do
  Select action a indicated by $\pi(s)$
  Observe next state s' and reward r
  $S(s,a) \leftarrow max\left(0, S(s,a) + \ln \frac{T_{uni}(s,a,s')}{T_{cur}(s,a,s')}\right)$
  if $S(s,a) > c$ **then**
    $T_{cur}(s,a,\kappa) \leftarrow \frac{1}{|\mathcal{S}|},\ \forall \kappa \in \mathcal{S}$
    $S(s,a) \leftarrow 0$
  end if
  Update T_{cur} according (1) (2)
  Update R_{cur} according (3) (4)
  $N(s,a) \leftarrow min(N(s,a)+1, M)$
  $s \leftarrow s'$
end while

5 評価実験

本節では、今回の手法の効果を [4] の実験で用いられたボールキャッチ問題を部分的に環境が変化する非定常環境において評価する。

5.1 実験方法

トーラス状のグリッド環境において、猫が移動するボールをキャッチすることを目標にする。グリッド (マス) の大きさは 15 × 15 である。猫の位置から見たボールの位置をエージェントの状態とする。猫は毎時間ステップにおいて、1 マス上下左右のどれかに移動するかまたは同じ位置に留まるかという 5 つの行動から一つを選択する。ボールは上下左右のうち、一つの方向に動き続ける。本実験では左へ動き続ける設定にした。猫が行動を選択した後にボールは 1 マス動く。報酬は各行動に対して-1 であり、ボールをキャッチすると 10 得られる。1 エピソードにおけるステップ数の最大値は 100 であり、猫がボールをキャッチするか 100 ステップを超えるとそのエピソードは終了する。各エピソードでは、猫はグリッドの中心から、ボールはランダムな位置から始める。100 エピソードで 1 バッチとする。このような環境で以下の 2 つの実験を行った。

実験 1 として、複数回の部分的環境の変化に対しての効果の検証を行った。5 から 9 バッチでは相対位置が (1,0),(2,0) の場合には、ボールを右に 1 マス動かすように変更する。10 から 14 バッチでは相対位置が (1,1),(1,14) の場合には、ボールを右に 1 マス動かすように変更する。つまり、以下の図 1 の赤い範囲にボールがある場合、ボールは右に移動する。その他のバッチでは、通常の環境で実行している。これを 20 バッチ実行した。

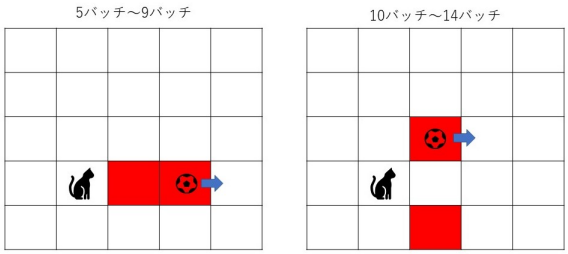

図 1　部分的環境変化

実験 2 として、環境全体の変化を行った。10 バッチごとにボールの動きを右、左、上、下と変化させた。これを 40 バッチ実行した。

5.2 実験結果

実験 1 での結果は図 2 のようになった。縦軸は 1 バッチの平均ステップ数を表している。ハイパーパラメータは割引率 γ=0.9、ε -greedy 法における探索率 ε =0.05、Prioritiezed Sweeping の学習率 α=0.2、反復回数は 10 回で実験を行った。文献 [5] の手法である RLCD with SCD では、環境の一部分が変化したときは検知できていないが、本手法では部分的な変化を検知して適応できている。

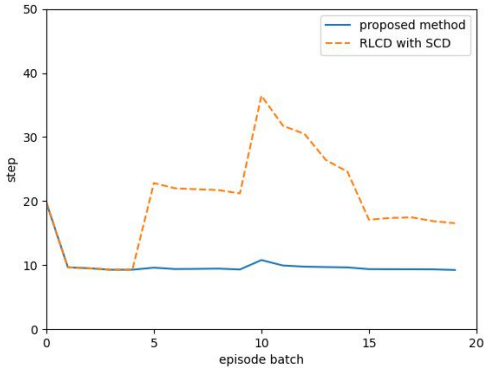

図 2　実験 1 の結果

実験 2 の結果は図 3 のようになった。全体の環境が変化している 5、10、15 バッチにおいて提案法に比べて、既存法が早く適応できている。

6 考　　察

実験 1 の結果より、環境が変化した際、提案法ではモデルの初期化をしないため、それまでに学習した情報を活用できており、効率的に方策の部分的な再学習が出来ていることが分かる。以下の図 4 に相対位置が (1,0)、(2,0) での部分的環境変化前後でのそれぞれの相対位置における最大の Q 値の値とその行動をヒートマップで表している。環境が変化した後、相対位置が (1,0)、(2,0)、つまり、猫の右 2 マスにボールがある場合、ボールが右に移動するため、ボールは猫から遠ざかってしまう。そのため、ボールの転がる行に猫が存在する場合、猫は上か下に

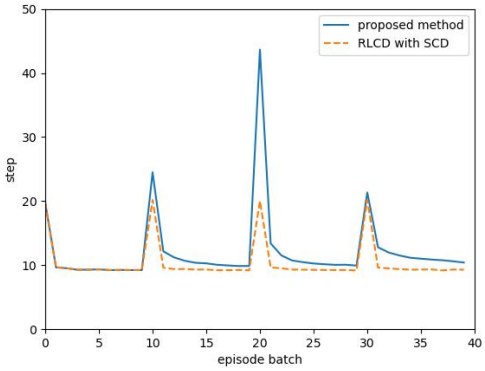

図 3　実験 2 の結果

ずれる必要がある。この図 4 を見ると、既存法では、環境変化前の Q 値に引きずられ、適切に Q 値の更新が出来ていない。これに対して、提案法では環境変化前の Q 値を利用しつつ、新しい環境に適応できていることが分かる。つまり、猫は上か下にずれるということが学習できている。

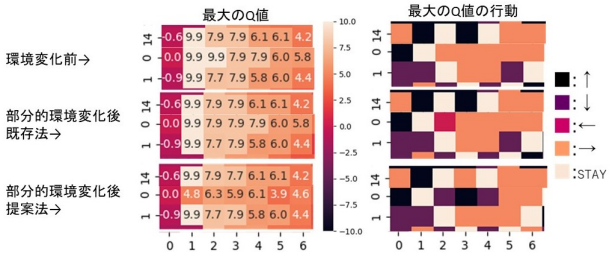

図 4　それぞれの相対位置における Q 値のヒートマップ図

以下の図 5 に相対位置が (2,0) において、提案法と既存法の最大 Q 値のパスを示している。提案法では一度下の行に移動してから、ボールが右上に来た時にキャッチしに行く。これに対して、既存法では (2,0)、(4,0) を繰り返すことになる。既存法でも、追加で実行を繰り返すと、やがて提案法と同じ軌道を学習するが、場合によっては適応がとても遅い。

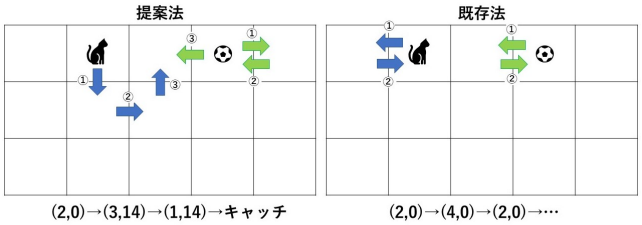

図 5　最大 Q 値のパス

実験 2 では、ボールの振る舞いが 10 バッチ毎に変化する。つまり、全ての状態、行動における状態遷移確率は変化する。結果を見ると、環境全体が変化する場合では、提案法では部分的に状態遷移確率を修正していくため、従来法に比べて適応速度が遅いことが分かる。しかし、バッチ数を重ねるごとに、ステップ数にほぼ差はなくなっているため、環境の変化に適応は出来ていると考えられる。

従来法では、複数のモデルを保持し、同じ環境に遭遇した場合、モデル集合から該当するモデルを選択することが出来る。そのため、環境全体が変化し、同じ環境が出現する場合では従来法の方が有用性が高い。これに対し、環境変化が部分的に発生し、同じ環境が繰り返されない場合は本手法が有効である。

7　ま　と　め

CUSUM 法を用いることにより、部分的に変化した環境に素早く適応できるモデルベース強化学習手法を開発した。本手法は、従来法で適応できていない部分的な環境の変化に対して対応することが出来るというメリットがあるが、環境全体が変化した場合では、順応速度は遅いというデメリットがある。

今後の課題として、状態遷移が決定的な場合での変化の検知だけでなく、状態遷移が確率的でも検知できるか、報酬関数の変化も対応できるか、実験環境を変えて実験を行いたい。また、全体の環境変化と部分的環境変化を別々に検知できるか検証したい。本実験では大部分の状態遷移確率が同じで、一部分のみ変化した場合、従来の手法では適応できない場合があることを示したが、現実のアプリケーションでもでも同じような環境があるか調査したい。

文　　献

[1] Richard S. Sutton and Andrew G. Barto. 2018. Reinforcement Learning: An Introduction (2nd ed.). MIT Press, Cambridge,MA.
[2] Samuel P. M. Choi, Dit-Yan Yeung, and Nevin Lianwen Zhang. 2000. “An environment model for nonstationary reinforcement learning“ In Advances in Neural Information Processing Systems. 987–993.
[3] Samuel P. M. Choi, Dit-Yan Yeung, and Nevin L. Zhang. 2000. “Hidden-mode Markov decision processes for nonstationary sequential decision making.“ In Sequence Learning. Springer, Berlin, 264–287.
[4] Bruno C. da Silva, Eduardo W. Basso, Ana L. C. Bazzan, and Paulo M. Engel. 2006. “Dealing with non-stationary environments using context detection“ In Proceedings of the 23rd International Conference on Machine Learning. 217–224.
[5] Emmanuel Hadoux, Aurélie Beynier, and Paul Weng. 2014. Sequential decision-making under non-stationary environments via sequential change-point detection. In Learning over Multiple Contexts. Nancy, France.
[6] Basseville, M., Nikiforov, I.V. 1993. “Detection of Abrupt Changes: Theory and Application.“ Prentice-Hall