学習オートマトンを用いたQ学習の高速化

小山 裕（電子情報システム工学専攻）

Accelerated Q-learning with Learning Automata

Yu Koyama
(Advanced Electronic and Information Systems Course)

Abstract

The Q-values in Q-learning, the most applied representative reinforcement learning in Markov Decision processes (MDP), will converge to the optimal values with probability one when each action is selected infinitely in any states. However, the convergence property of conventional action selection methods in Q-learning has been discussed from the perspective of ϵ-optimal (probability convergence), which is weaker than optimal (with probability one convergence). In this study, LQ-learning, which is introducing β-type learning automata with conditional optimality, is considered. Then, it is shown that the LQ-learning has so faster property than conventional action selection methods such as ϵ-greedy in Q-learning.
Keywords: Q-learning, β-type learning automata

1. はじめに

マルコフ決定過程（Markov Decision Process, 以下MDP）に従う環境において，行動とそれに対する報酬を用いた学習を繰り返して最適行動を習得する手法を強化学習という．なかでも強化学習手法の代表例であるQ学習は，各状態における行動を無限回選択する場合に，Q値が最適な値に概収束することが証明されている [1]．しかし，これまでにQ学習で提案されている行動選択法については最適行動の選択確率の収束性は保証されておらず，一部の行動選択法の学習性能が概収束よりも弱いϵ-最適 (確率収束) の概念で議論されているに過ぎない [2]．

本研究ではこの問題に着目し，ある条件を満たした確率的環境下で最適となる条件最適性を有するβ-タイプ学習オートマトンをQ学習の行動選択手法に導入するLQ学習を提案する．LQ学習では，条件最適性によりQ学習の学習速度を抜本的に改善する．そして，タクシー問題によるシミュレーション実験において，Q学習における代表的な行動選択法であるϵ-greedy法を用いたエージェントとLQ学習を用いたエージェントの最適行動選択率を比較し，本手法の優位性を検証した．また，LQ学習の一部のパラメータを変更した際の学習に及ぼす影響についても，実験により検証した．

2. Q学習と行動選択

2.1 Q学習

Q学習は，マルコフ決定過程（MDP）において用いられる強化学習手法の1つである．Q学習では，環境のある状態においてエージェントは自身の状態行動価値Qに基づいて行動選択を行う．その行動選択の結果として環境から得られる行動の評価を示す報酬をもとにQ値を更新する．この一連の操作を「試行 (step)」と呼び，この試行を繰り返すことでエージェントは最適な行動系列（最適政策）を学習する [1]．Q値の更新式は，以下の (1) 式のとおりである．$s(t)$, $\alpha(t)$ はそれぞれ時刻 t における状態とそのときの行動であり, $r(t)$ はその行動の結果, 環境から与えられる報酬である．また, η, γ は, 学習率, 割引率と呼ばれるパラメータである．

$$\begin{aligned} Q(s(t),\alpha(t)) \leftarrow\ & Q(s(t),\alpha(t)) \\ & + \eta[r(t+1) + \gamma \max_{\alpha(t+1)} Q(s(t+1),\alpha(t+1)) \\ & - Q(s(t),\alpha(t))] \end{aligned} \tag{1}$$

Q学習では，マルコフ決定過程に従う環境の各状態において，各行動を無限回選択する場合に状態行動価値Qは最適な値に収束することが証明されている [1]．すなわち，理論上は $t \to \infty$ としたとき Q 値は完全に収束し，エージェントはどの状態においても最適なQ値から漸近的に最適行動を発見できる．

2.2 ϵ-greedy 法

Q 値による行動選択の手法はいくつか種類があるが, 代表的な手法が ϵ-greedy 法である．ϵ-greedy 法とは, 確率 ϵ でランダムに行動し, 確率 $1-\epsilon$ で最も状態行動価値 Q の大きい行動を選択する手法である．この手法では, 局所解に陥ることを防ぐため常にランダムな行動 (探索)

を行う確率が存在する．一方で，十分探索を行った後でも最適行動以外の行動をとる確率が存在する．

その他の行動選択法としては，softmax 法などがあげられるが，これらの行動選択法において ϵ-最適や最適の概念からの研究はほとんど行われておらず，一部の研究で提案された高速化アルゴリズムが ϵ-最適となることが示されているに過ぎない [2]．

この問題に対し，本研究ではある条件を満たすときに最適行動への収束が保証されている β-タイプ学習オートマトンを行動選択に適用し，学習の高速化を図る．

2.3 学習オートマトンにおける評価規範

最適とは，学習個体が最適行動を選択する確率の総和が $t \to \infty$ で 1 に収束することをいい，(2) 式のように表される．ここで，$\underline{\alpha}_{opt}$ は最適行動のみを要素とする行動集合，$\phi_i(t)$ は時刻 t において最適行動 $\alpha_i \in \underline{\alpha}_{opt}$ を選択する確率を表す．

$$\lim_{t\to\infty} \sum_{\alpha_i \in \underline{\alpha}_{opt}} \phi_i(t) = 1 \quad a.s. \tag{2}$$

一方，ϵ-最適では，学習個体が最適行動を選択する確率の総和の期待値が $t \to \infty$ であっても 1 に収束することはない．ϵ 最適は (3) 式のように表される．ここで，ϵ は非常に小さい正の定数である．

$$\lim_{t\to\infty} E[\sum_{\alpha_i \in \underline{\alpha}_{opt}} \phi_i(t)] \geq 1 - \epsilon \tag{3}$$

これらの式の関係から，ϵ-最適は最適よりも収束速度が遅いという特性が知られている [3]．

3. β-タイプ学習オートマトン

特性が未知の確率的環境下で，生物と相似な振る舞いをする学習モデルの一種として学習オートマトン [4, 5] が存在する．学習オートマトンにもいくつか提案されているが，その中でも学習オートマトンの取り得る行動のそれぞれに対応づけた複数のベイズ推定器 [3] によって構成され，ベイズ推定によって環境からの応答値の推定を行う機能を有したものを β-タイプ学習オートマトンと呼ぶ．

3.1 ベイズ推定器

ベイズ推定とは，ある確率分布に従う確率変数を与えたとき，あらかじめ用意した確率分布のモデルのうち最も近いものを推定する手法である．このベイズ推定の原理を用いて，期待値を推定する形に拡張した学習モデルをベイズ推定器と呼ぶ．図 1 に一般的なベイズ推定器の入出力関係を示す．

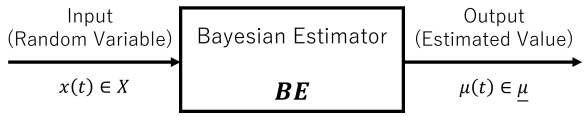

図 1 ベイズ推定器の入出力関係

このベイズ推定器は，$BE =< X, \Omega, \underline{\mu}, \boldsymbol{\lambda}(t), \mu(t) >$ により構成される．ここで，$\Omega = \{\omega_1, \omega_2, \cdots, \omega_m\} (m \geq 2)$ はベイズ推定器の取りうる状態集合，$\mu(t) \in \underline{\mu}$ はベイズ推定器が出力する実数値であり，$\underline{\mu} = \{\mu_1, \mu_2, \cdots, \mu_m\}$ は各状態 $\omega_l \quad (l = 1, 2, \ldots, m)$ に対応付けた

$$0 < \mu_1 < \mu_2 < \cdots < \mu_m < 1 \tag{4}$$

を満たす実数値の出力集合である．本研究では閉区間 [0,1] における $m+1$ 等分の分割の分点，すなわち

$$\mu_l = \frac{l}{m+1} \ (l = 1, 2, \cdots, m) \tag{5}$$

とする．また，$\boldsymbol{\lambda}(t) = (\lambda_1(t), \lambda_2(t), \cdots, \lambda_m(t))$ は時刻 t におけるベイズ推定器の状態をランダムに決定する m 次元状態確率ベクトルである．すなわち，これは BE が時刻 t において状態 $\omega(t) = \omega_l \in \Omega$ を決定する確率は $\lambda_l(t)$ であることを意味する．$t = 0$ における初期値は

$$\lambda_l(0) = \frac{1}{m} \quad (l = 1, \cdots, m) \tag{6}$$

のように一様分布として定める．

時刻 t において，ベイズ推定器は $\boldsymbol{\lambda}(t)$ を用いて状態 $\omega(t) = \omega_l$ を決定すると，その状態に対応づけた実数値 $\mu(t) = \mu_l$（$\mu_l \in \underline{\mu}$）を出力する．

つぎに，ベイズ推定器は状態確率ベクトルの各要素を (7) 式によって更新する．

$$\lambda_l(t+1) = c \cdot \lambda_l(t) \cdot \{q_l(x(t))\}^{\theta} \quad (l = 1, 2, \cdots, m) \tag{7}$$

ただし，c は $\sum_{l=1}^{m} \lambda_l(t+1) = 1$ の条件を満たすように決定される正規化定数である．また，θ は状態確率ベクトルの収束速度を決定する正定数であり，$\theta = 1$ のときは通常のベイズ学習となる．$q_l (l = 1, 2, \cdots, m)$ は推定の対象となる確率密度の有限モデルであり，状態 $\omega(t)$ の決定はその有限モデル $Q = \{q_1, q_2, \cdots, q_m\}$ 中の確率密度の一つを選択することを意味する．この有限モデル q_l には各種のものが考えられるが，本研究ではつぎの非線形な連続有限モデルを用いた [3]．

$$q_l(x) = \mu_l^x \cdot (1 - \mu_l)^{(1-x)} \quad (l = 1, 2, \cdots, m) \tag{8}$$

(8) 式を (7) に適用することで，以下の更新式が導かれる．

$$\lambda_l(t+1) = c \cdot \lambda_l(t) \cdot \{\mu_l^x \cdot (1-\mu_l)^{(1-x)}\}^{\theta} \quad (l = 1, 2, \cdots, m) \tag{9}$$

これらの過程を繰り返すことで，ベイズ推定器は入力値 $x(t) \in [0,1]$ がとる期待値に最も近い $\mu_l \in \underline{\mu}$ を出力するようになる．

3.2 β-タイプ学習オートマトン

図 2 に本研究で用いる β-タイプ学習オートマトンの構成を示す．ここで，行動 α_i は行動集合 $\underline{\alpha} = \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ 中の要素，$x(t)$ はある時刻 t における学習オートマトンの選択行動に対する評価を示す確率的環境の応答値であり，閉区間 $X = [\underline{x}, \overline{x}]$ 中の値をとる．図に示すように β-タイプ学習オートマトンは，学習オートマトンの行動 $\alpha_k \in \underline{\alpha}$ のそれぞれに対応付けた r 個のベイズ推定器 $BE < X, \Omega, \underline{\mu}, \boldsymbol{\lambda}_k(t), \mu_k(t) > (k = 1, 2, \cdots, r)$ から構成される．

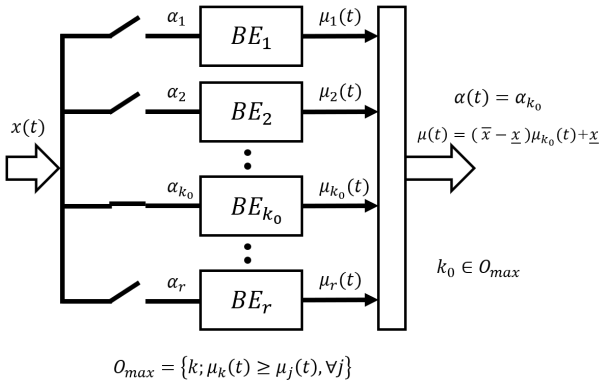

図 2 β-タイプ学習オートマトンの構造

前述した通り，β-タイプ学習オートマトンはある条件を満たすとき最適行動の選択確率が 1 に収束する条件最適性が保証されている．その条件とは，最適行動 α_{i_0} に対する期待利得 c_{i_0} と他の行動 α_i に対する期待利得 c_i の間に常に

$$c_{i_0} - c_i > \min_k(\mu_{k+1} - \mu_k) \tag{10}$$

という関係が成り立つことである．いま，(5) より出力する推定値を閉区間 [0,1] において m 等分しているため，(10) の不等式はつぎのように表される．

$$c_{i_0} - c_i > 1/m \quad (= \Delta) \tag{11}$$

すなわち，状態数 m を十分大きくし，c_{i_0} と他の c_i との差が小さくなるような推定精度が得られるとき，β-タイプ学習オートマトンの最適性が保証される．これにより，$t \to \infty$ で学習オートマトンが最適行動を選択する最適行動選択確率が確率 1 に概収束する．

4. 提案手法

本研究では，β-タイプ学習オートマトンを行動選択法として用いる手法を提案する．本手法を，LQ 学習と命名する．

LQ 学習では図 3 のように，β-タイプ学習オートマトンをマルコフ決定過程に従う環境の状態数 n だけ，ベイズ推定器を各状態において取りうる行動数 r だけ用意する．エージェントは観測した状態に対応する β-タイプ学習オートマトンに状態行動価値 Q を入力値 $x(t)$ として与え，β-タイプ学習オートマトン内部のベイズ推定器によって Q 値の推定が行われる．そして，得られた出力値のうち最大のものを出力したベイズ推定器に対応した行動を β-タイプ学習オートマトンが決定する．一方，各状態では通常の Q 学習と同様に Q 値のテーブルを保持し，学習オートマトンによる行動選択で得られた Q 値を格納する．

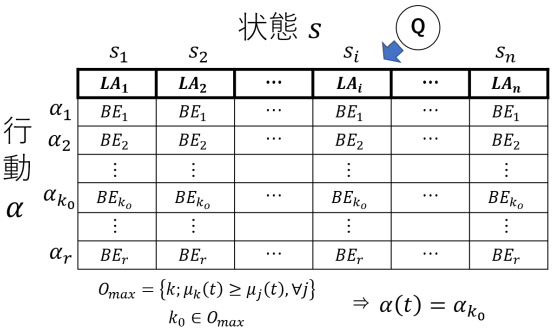

図 3 LQ 学習における行動選択法

本手法のアルゴリズムを以下に示す．

Step 0 : まず，各 β-タイプ学習オートマトン $LA_i (i = 1, 2, \cdots, n)$ 中の各ベイズ推定器 $BE_k (k = 1, 2, \cdots, r)$ について，その出力集合 $\underline{\mu}$ を (5) 式，状態確率ベクトル $\boldsymbol{\lambda}_k(0)$ を (6) 式に従って初期化する．また，$t \leftarrow 0$ とする．

Step 1 : 時刻 t において，エージェントは環境の状態 $s(t)$ を認識する．$s(t) = s_i$ であったとき，その状態に対応した β-タイプ学習オートマトン LA_i の内部にある各ベイズ推定器 $BE_k (k = 1, 2, \cdots, r)$ は独立に $\boldsymbol{\lambda}_k(t)$ から自身の状態 $\omega_k(t) = \omega_{l_k} \in \Omega$ を定め，推定値 $\mu_k(t)$ を出力する．

Step 2 : β-タイプ学習オートマトン LA_i は各 BE_k の出力 $\mu_k(t)$ を精査し，その中の最大値をもとに行動を決定する．すなわち，$O_{max} = \{k | \mu_k(t) \geq \mu_j(t), \forall_j\}$, $k_0 \in O_{max}$ であるような k_0 に対して

$$\begin{cases} \alpha(t) = \alpha_{k_0} \in \underline{\alpha} \\ \mu(t) = (\overline{x} - \underline{x})\mu_{k_0} + \underline{x} \end{cases} \tag{12}$$

のように LA_i は時刻 t における行動 $\alpha(t)$ と出力 $\mu(t)$ を決定する．ただし，$|O_{max}| \geq 2$ のときは，k_0 を集合 O_{max} 中からランダムに 1 つ決める．その後，エージェントは LA_i により決定された行動 $\alpha(t)$ を環境に対して実行する．

Step 3： エージェントは行動 $\alpha(t)$ に対する環境からの報酬 $r(t+1)$ を受け取り，(1) 式から状態行動価値 Q を更新する．更新後，その Q 値を応答値 $x(t)$ として（すなわち $x(t) = Q(s(t), \alpha(t))$），**Step 2** で行動決定を行った β-タイプ学習オートマトン LA_i へ渡す．ただし，学習の序盤や終盤では Q 値があらかじめ決めた応答値の最小値 $\underline{x}$ を下回る，あるいは最大値 $\overline{x}$ を上回るといった事象が発生することが考えられる．したがって，LA_i へ渡す応答値 $x(t)$ は以下の (13) ように決める．

$$x(t) = \begin{cases} \underline{x} & if \quad x(t) \leq \underline{x}, \\ \overline{x} & if \quad x(t) \geq \overline{x}, \\ Q(s(t), \alpha(t)) & else. \end{cases} \tag{13}$$

応答値 $x(t)$ を受け取った LA_i は行動 α_{k_0} に対応するベイズ推定器 BE_{k_0} の状態確率ベクトル $\boldsymbol{\lambda}_{ik_0}(t)$ のみを (9) 式を用いて更新する．ただし，ベイズ推定器は応答値 $x(t)$ は閉区間 [0,1] 中の値であることが前提となっているため，(9) 式の更新の際は以下の (14) 式で定義される $x'(t)$ を応答値 $x(t)$ の代わりとして用いる [6]．

$$x'(t) = \frac{x(t) - \underline{x}}{\overline{x} - \underline{x}} \tag{14}$$

Step 4： $t \leftarrow t+1$ として，**Step 1** に戻る．

5. 実験

提案手法の，従来の行動選択法に対する優位性を検証するため，タクシー問題 [7] を対象とするシミュレーション実験を行った．そして，各行動選択法においてエージェントが最適行動を選択する確率を求め，最適行動への収束の速さやその精度について比較した．また，LQ 学習において (9) 式中のパラメータ θ を変更することによる，学習特性の変化についても実験を行い検証した．

5.1 タクシー問題

タクシー問題とは，タクシー運転手が 3 都市を回り，集客を行う際の最適行動を考える問題のことをいう．タクシー運転手 (エージェント) は都市 $i(i = 1, 2, 3)$ にいる状態を状態 i として認識する．状態 1, 3 においてはつぎの（1）, (2) , (3）の行動を選択することができ，状態 2 においては（1）, (2）の行動を決定することができる．

(1) 手をあげる通行人を拾うために流して回る

(2) 最も近いタクシースタンドに行き，順番を待つ

(3) 無線連絡による呼び出しを待つ

各状態と決定に対する状態遷移確率と利得は表 1 の通りに定める．この問題の最適行動はすでに調べられており，行動（2）がどの状態においても最適行動であることが明らかになっている [7]．

表 1 タクシー問題の設定

状態	行動	状態遷移確率			利得		
		状態 1	状態 2	状態 3	状態 1	状態 2	状態 3
1	1	0.5	0.25	0.25	10	4	8
	2	0.0625	0.75	0.1875	8	2	4
	3	0.25	0.125	0.625	4	6	4
2	1	0.5	0	0.5	14	0	18
	2	0.0625	0.875	0.0625	8	16	8
3	1	0.25	0.25	0.5	10	2	8
	2	0.125	0.75	0.125	6	4	2
	3	0.75	0.0625	0.1875	4	0	8

5.2 実験 1

本実験では，先に述べたタクシー問題の環境に対してエージェントの行動選択法を適宜変更し，最適行動の選択率を求めた．比較する手法は，(1)ϵ 値を固定値とした ϵ-greedy 法 (ϵ 固定法)， (2)ϵ 値を更新する形にした ϵ-greedy 法 (ϵ 更新法)， (3) 提案手法 (LQ 学習) の 3 つとした．(2) の行動選択法における ϵ 値の更新式は，(15) 式のように定めた．表 2 に実験環境を, 表 3 にシミュレーションの際に設定した各パラメータを示す．

シミュレーションにおいて 1 エピソード 300 試行とし，これを 100 エピソード繰り返した．この 100 エピソードの過程を 50 回行い，エピソードに対するエージェントの最適行動の選択率の平均を結果とした．

$$\epsilon(t+1) = k \cdot \epsilon(t) \qquad (0 < k < 1) \tag{15}$$

表 2 実験環境

OS	Microsoft Windows 10 Pro 22H2
CPU	AMD Ryzen 5700G(3.8GHz)
コア数	8
スレッド数	16
メモリ	32GB
使用言語	Python 3
台数	1

表 3 設定したパラメータ

パラメータ	記号	値
学習率	η	0.1
割引率	γ	0.9
ϵ 固定法の探索率	ϵ_{fix}	0.1
ϵ 更新法の探索率の初期値	$\epsilon(0)$	0.1
ϵ 更新法の ϵ の変化率	k	0.999
状態数	m	200
応答値の最大値	$\overline{x}$	150
応答値の最小値	$\underline{x}$	100
学習速度を決定する正定数	θ	1

5.3 実験 1 の結果

図 4 に実験 1 の結果を示す．グラフにおいて横軸がエピソード数，縦軸が最適行動選択率を表している．図より，従来手法と比べ LQ 学習を適用したエージェントの方が最適行動の選択率が最も早い段階で 1.0 に近づいていることが確認できた．また，探索が十分となるエピソード後半においても，ϵ 更新法と僅差ではあるが LQ 学習の最適行動選択率が最も 1.0 に近いことが確認できた．

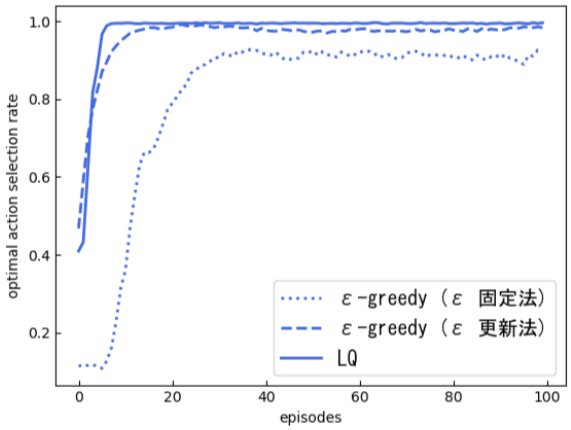

図 4 各行動選択法でのシミュレーション結果

5.4 実験 2

本実験では，LQ 学習における学習速度を決定するパラメータ θ を変更したときの学習特性の変化を，実験 1 と同様のシミュレーションにより検証した．パラメータ θ を大きく設定することで，最適行動に対する状態確率ベクトルの更新が速まり，より早い段階で最適行動へ収束することが期待される．本実験の実験環境およびパラメータは，θ の値以外は表 2，3 と同条件に設定した．

シミュレーションにおける試行数や，結果の算出方法も実験 1 と同様にし，1 エピソード 300 試行 ×100 エピソードの過程を 50 回行ったときの，エピソードに対するエージェントの最適行動の選択率の平均を結果とした．

5.5 実験 2 の結果

図 5 に実験 2 の結果を示す．結果から，$\theta = 1, 2, 5$ とした場合ではエージェントの最適行動選択率には大きな差は見られなかった．一方で，$\theta = 10$ とした場合では他の θ の値のものに比べ，最適行動の選択率は 1.0 からは最も遠く，収束も遅くなっていることが確認できた．

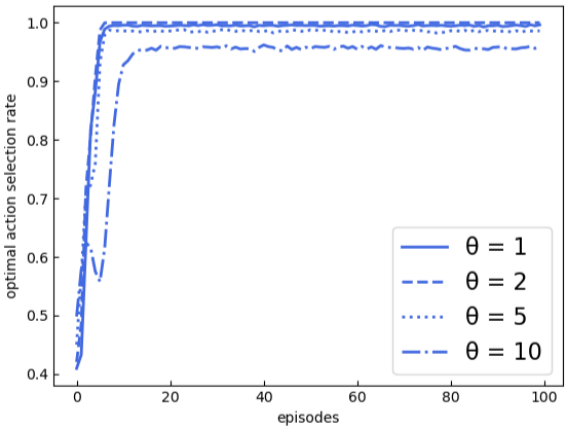

図 5 $\theta = 1, 2, 5, 10$ でのシミュレーション結果

6. 考察

図 4 に示される実験の結果から，提案する β-タイプ学習オートマトンを用いた LQ 学習が，ϵ-grddy 法を用いた Q 学習に比べて高速な収束特性を有することが確認できた．これは，β-タイプ学習オートマトン内部のベイズ推定器が未知確率分布の有限モデルを用いたベイズ推定を行っており，マルコフ決定過程に従う環境の持つ未知確率分布を高速に推定できていることが理由である．また，(11) 式を満たすような十分な推定精度が得られる状態数 m を決めたことで条件最適な特性が発揮されたことも高速化の理由であると考えられる．

また，エピソード後半においても最適行動選択率が最も高いのは LQ 学習を用いたエージェントであった．これは理論上，$t \to \infty$ としたとき LQ 学習においては学習オートマトンが最適行動を選ぶ確率が 1 に概収束することが証明されているため，この性質が結果として現れたと考えられる．

一方，(9) 式において学習速度を決定するパラメータ θ の挙動についても実験を行った．しかし，図 5 に示される結果から，θ の値による学習への影響は小さいことが確認できた．また，$\theta = 10$ のように θ を大きくしすぎると，タクシー問題においては誤学習が発生し，$\theta = 1$ 以外のパラメータでは良い結果が得られなかった．これは，$\theta = 1$ のベイズ推定そのものが既に高速であり，数値計算上，実数値データの有効桁数が大きくとれないことが原因である．

2 つの実験全体に言えることとして，ベイズ推定は未知確率分布と推定した確率分布間の二乗誤差の減少率の点で最適であることが知られている [6]．このことから，学習速度の高速性が説明でき，さらなる高速化（パラメータ θ の調整）は限界があることが予想される．

7. おわりに

本研究では，行動選択法の観点から Q 学習の高速化を図るため，行動選択に β-タイプ学習オートマトンを用いた LQ 学習を提案した．そして，その収束速度を検証するため，タクシー問題を対象としたシミュレーションにより従来の行動選択法による Q 学習の学習速度と比較した．結果から，LQ 学習によって最適行動への収束が高速化されることを確認した．また，LQ 学習のパラメータ θ を変更した際の学習の挙動についても調査した．こちらは，θ の値によって学習速度が更に向上する様子は見られず，むしろ θ を大きくしすぎることで誤学習を引き起こすことを確認した．

本手法の課題点もいくつかあげられる．本手法では，学習に必要な要素として通常の Q 学習で用いる Q 値を保存する機構 (Q テーブル) に加え，環境の状態数 n に対応付けた n 個の β-タイプ学習オートマトン，各 LA_i が内部に持つ行動数 r に対応付けた r 個のベイズ推定器，各 BE_k の状態数 m だけ要素を持つ状態確率分布 $\lambda_k(t)$ が存在する．これにより，単純積で $n \times r \times m$ だけ実数値データの計算資源を確保する必要があり，大規模な環境を対象とした学習を行う場合は計算資源が枯渇してしまう恐れがある．そのため，より規模の大きい環境に対してはより効率的に計算資源を利用できるような手法へ改良することが課題となる．

また，LQ 学習ではベイズ推定の原理に基づいた推定を行うが，このベイズ推定は環境の確率的パラメータが既知であることを前提とする「モデルベース型」の学習手法の一種である．本研究では，状態数 m，応答値の最大値 $\overline{x}$，応答値の最小値 $\underline{x}$ が環境の確率的パラメータに該当する．一方で，Q 学習は環境の確率的パラメータが未知の場合でも適用可能な「モデルフリー型」の学習手法である．そのため，LQ 学習ではベイズ推定のモデルベース性によって Q 学習の特長であるモデルフリー性が損なわれており，確率的パラメータが未知な環境では適用が難しい．この問題に対して，アルゴリズムを改良し，モデルフリーな学習を実現することも課題点としてあげられる．

本研究の成果が，Q 学習の高速化に対して，その一助となることを期待したい．

参考文献

1) CJCH.Watkins, P.Dayan: ”Q-Learning”, *Machine Learning*, Vol.8, pp.279-292, 1992.
2) M.G.Azar, et.al.: ”Speedy Q-Learning”, Neural Information Processing Systems, 2011.
3) 原元司，阿部健一: ”条件最適な学習オートマトン”，計測自動制御学会論文集，Vol.28，No.6，742-749，1992.
4) 馬場則夫: ”学習オートマトン理論の最近の話題”, 計測と制御, Vol.26, No.9,pp.901-808(1987).
5) K.S.Narendra: ”Learning Automata:An Introduction”, Prentice Hall(1989).
6) 原 元司，阿部 健一: ”確率的ネットワークにおける最短経路問題への分散学習オートマタアプローチ”，電子情報通信学会論文誌，Vol.J76-D-，No.9，pp.2116-2125(1993).
7) 小河原正巳，坂本武司: ”マルコフ過程”，共立出版株式会社，1967.