学習空間を全体空間とその部分空間で二重化する強化学習法

松井 博和^{*1} 西澤 智恵子^{*2} 野村 由司彦^{*1}

Reinforcement Learning by Using Dual Q-table for the Whole Space and a Partial Space

Hirokazu Matsui^{*1}, Chieko Nishizawa^{*2} and Yoshihiko Nomura^{*1}

In this paper, we propose a Q-learning method by using dual Q-table. Concretely, the proposed method has two Q-tables: “Whole Q-table” is larger, based on the whole space (in enough detail for learning optimal actions) of the environment and “Partial Q-table” is smaller, based on a subspace (rough for learning outline actions) of the whole space. The two Q-tables simultaneously learn the environment based on a selected action. The action is selected by using the more learned Q-table out of the two Q-tables by each step. We simulated the proposed method, comparing with conventional ones, under three conditions of the learning environments, where the partial Q-table can learn optimal actions at the highest rate, middle rate and the lowest rate of the situations in the environment. As a result, we indicated that the proposed method can learn the optimal actions at any rates. As the rate is higher, it converges earlier. Even if at the lowest rate, the proposed method is almost as effective as conventional one. And we indicated the proposed method to be effective by using mathematical analysis. Furthermore, we verified that the proposed method was effective under an actual environment.

Key Words: Reinforcement Learning, Q-learning, Dual Q-table

1. はじめに

春, 迷いながら目的地まで到達する近道を見つけ通ったとする。夏, それぞれの場所で思い出しながら, その道を再び通る。秋, 紅葉や落ち葉があっても, その道を再び辿る。冬になり, その道を選んでも, 雪により通れない部分があると, 春夏秋は通れるが冬は辿れない場所があることを知り, 遠回りでも別の道を通る。次の年, 春夏秋はその近道を選ぶが, 冬は遠回りの道を選ぶ。これは人が季節を適切に考慮して場所ごとに目的地へとつながる適正行動を学習できることを示している。我々は, これを場所のみで表す状態(場所状態)に対する行動学習と季節と場所の組み合わせで表す状態(季節・場所状態)に対する行動学習をうまく組み合わせることで効率的に学習するからと考える。この組み合わせがなく, 場所状態に対する行動学習のみでは, 春夏秋と冬で異なる道を学習できなくなる。また, 季節・場所状態に対する行動学習のみでは, 春の学習結果が夏秋冬に, 夏のものが秋冬に, 秋は冬に利用できず学習効率が悪くなる。本論文は, 季節・場所状態に対する行動学習を全体空間に対する行動学習, 場所状態に対する行動学習を部分空間に対する行動学習と一般化し, 二つの行動学習を同時に学習し, 二つのうちの学習が進んでいると考えられるものに基づき行動選択することで, ロボットが効率良く学習するアルゴリズムを提案する。

学習とは, エージェントが置かれる状況(以下, 環境とよぶ。)ごとに, 適正行動を対応付けて記録する過程である。それぞれすべての環境に対応する最適行動を求めるためには, それぞれすべての環境において, 選択可能なすべての行動を試さなければならない。ロボットが未知環境の中で適正行動を獲得する方法の一つとして, 強化学習[1]とよばれる手法が用いられる。強化学習法とは, 様々な行動を選択し, その結果の報酬に基づいて選択した行動の良し悪しを評価し, 環境と適正行動の対応付けを求めるものである。強化学習における試行回数の削減は, 実環境での学習において重要な問題となる。ここで, 文献[2]~[4]のように, 環境から得られる報酬パラメータを調整することで, 複数の学習エージェントの協調動作を学習させるものではなく, 一つの学習エージェントに対する学習試行回数の削減について従来法を以下の四つに整理し考える。(1) 全体空間から切り出した部分空間を作り, サブゴール化, あるいは, 状況に対する部分空間の適正度に応じて部分空間を切り替えたり, 学習空間の状態数の最大を決めて, その中で状態をうまく分割する仕組みにより, 学習対象の状態数が膨大になるのを防ぎ, 学習の遅延を防ぐ手法, (2) 関数を用いて, 連続・高次元空間の状態や行動の価値を近似することで, 学習対象の状態数やパラメータを減らし, 学習を促進する手法, (3) 既学習情報や設計者が与える情報を事前知識として利用したり, 環境変化のあるところで探索行動をし, 環境変化のないところでタスク達成可能性が高い行動を選択されやすくし, 学習を効率化する手法, (4) 複数の学習空間を同時並列に学習させ, ある一定期間の行動経験により得られた報酬を基に, 複数の学習空間の価値を推定, 比較してより価値の高い空間に切り替えることで, 学習を促進す

原稿受付 2017 年 10 月 3 日

^{*1}三重大学大学院工学研究科

^{*2}トヨタテクニカルディベロップメント(株)

^{*1}Graduate School of Engineering, Mie University

^{*2}Toyota Technical Development Corporation

■ 本論文は有用性(要素分野)で評価されました。

る手法がある。

(1) としては、複雑なタスクを一気に学習させるのではなく、目標達成付近の単純な状況から徐々に複雑な状況へと移行していく [5] や、簡単なタスクから段階的に複雑なタスクへと移行していく [6]、連続状態空間を離散化する場合、タスクに応じた適切な状態数を維持することで、過剰な状態が分割されることによる学習の遅延を防ぐ [7]、制御対象の将来の状態のための予測モデルと、制御出力を学習する強化学習コントローラの組を、学習空間を分割するように準備し、それらを予測モデルの誤差に基づいて切り替え、または組み合わせる [8]、状態空間は最初は粗く分けて学習が進まない場合にさらに分割、行動空間は学習が進むに従い連続空間に切り替えていく [9]、複雑なタスクを細かいサブタスクに分解した強化学習モジュールを階層的に並べて学習する階層型強化学習 [10] [11] がある。また、大きく複雑なマルコフ決定過程を木構造で表現する algebraic decision diagrams (ADDs) を入力とし、ゴールまでの行動回数を基に、大域的最適性を失わずに階層的クラスをつくり、ゴールまでの最短行動を探索しやすくする [12] がある。ただし、これは環境情報を学習しながら得るわけでないため、学習による環境情報の更新による木構造の改変は考慮していない。これらの研究は、環境分割の必要性を適宜確認しながら学習空間をつくりだすことにより、あるいは、全体の学習空間を分割した学習空間を複数用意して切り替えることにより、各学習空間の状態数が膨大となるのを防ぎ、効率良く学習する手法である。

(2) としては、正規化ガウス関数を基底として状態の評価関数を近似し、近似の精度により基底関数を動的に増やす [13]、状態価値関数近似に用いる要素に位置と勾配情報をもたせ、その要素を動的に追加・削除・再配置することで状態価値関数の形状の複雑さに即した近似をする [14]、ニューラルネットワーク出力層で、行動価値を分散表現することで行動空間の連続化を実現する [15] がある。これらは、連続空間をそのまま扱うことで、離散化の際に学習空間の状態数が膨大になるのを防ぎ、さらに、近似のためのパラメータを必要に応じて追加・削除したり、近似結果の出力方法を工夫することで学習パラメータが膨大になるのを防ぎ、効率良く学習する手法である。

(3) としては、回避行動はほぼすべてのタスクで共通に達成させる動作知識という考えに基づき、回避行動をタスク達成のための学習空間とは別の学習空間にて学習させ、別のタスクでもその学習済み空間を利用する [16]、学習済みの空間を利用して行動を選択し、試行回数に対するタスク成功率を測定し、タスク成功率の低下した環境のみ再学習する [17] や、行動選択則の温度パラメータを状態ごとに調整する [18]、強化学習で得られた学習結果から成功事例と失敗事例を収集し十分な数の失敗事例がたまる環境の変化を知覚し政策を変更する [19]、TD 誤差がある閾値を超えた場合に環境の変化を知覚する [20]、設計者が与えた基礎知識を利用して行動を選択し、学習初期のランダム行動を抑制する [21]、過去の成功・失敗経験に基づいて探索戦略を変化させる [22] [23]、学習初期に事前知識を用い、それが間違っている場合に対応するため、経験した回数に応じて忘却する [24] がある。これらは、既学習情報や設計者が与える情報を事前知識として利用し、環境変化のあるところで探索行動をし、環境変化のないところでタスク達成可能性が高い行動を選択されやすくし、学習を効率化する手法である。

(4) としては、学習法やパラメータ数の異なる学習空間を複数並列化してもち、目標達成ごとに得られた報酬から価値を推

定、比較して切り替える [25] や、自分の学習結果から同時に学習する他者の状態価値を推定し、比較してより高いほうを自分の行動学習に利用する [26] がある。これらは、ある一定期間の行動経験により得られた報酬を基に、複数の学習空間の価値を推定して比較し、より価値の高い空間に切り替える、あるいは、その空間の情報を利用することにより、学習を効率化する手法である。

(3) (4) を組み合わせる手法として、全状態を対象とする完全空間と学習エージェントの知覚情報を粗視化して状態数を減らす制限空間を並列に学習、学習初期は制限空間を用い、ある程度学習が進んだ後に完全空間に切り替える [27] がある。これは、学習初期は状態を粗く見て状態数を減らして制限空間で学習を進めることで、学習は速く進むが制限空間のみでは性能が悪いため、完全知覚空間でも並列に学習を進め、適当なタイミングで切り替えて学習を効率化する手法である。

これらの手法に対して、本提案手法では、学習の全体空間を使用しない (1) とは違い、すべてを学習対象とした全体空間を用いる。全体空間に加えて、学習空間の部分空間を並列化して学習を進め、学習進捗度に応じてそれら空間を切り替えることで学習の効率化を図る。部分空間は、全体空間より小さいため、学習自体が速く進む。全体空間の学習が進むまでは、学習が進んでいる部分空間を用いて行動を選択し、全体空間が進んでくると、全体空間を用いて行動を選択するようになる。つまり、全体空間をそのまま扱うことで大域的最適性を失わずに、学習初期のランダム行動を、部分空間の学習結果に基づいて、抑制することができる。(2) の学習空間を関数表現する方法は、環境の状況を状態に区切ることなく、その状況の値を関数のパラメータ調整により再現し、別の状況においても、関数に状況を代入すれば、状態価値等の妥当な値を得られるようにするものである。しかし、早く学習するようにパラメータ数を限定すれば近似精度が悪くなり、精度良くしようとすれば、多くのパラメータを必要とし、学習が遅くなる。この関数近似する手法に対して、本提案手法は優劣を比較するものではなく、技術的にまったく異なる手法であり、相補間的に組み合わせで性能を高めることが期待できる。例えば、本提案手法に関数近似法を導入すると、パラメータの多いものと少ないもののそれぞれの関数近似表現をもち、パラメータ学習をそれぞれ同時にする手法となる。本提案手法では、学習の効率化に用いる部分空間が全体空間と同時学習が可能のため、(3) と違い、事前知識や既学習情報を準備する必要はない。また、本手法では、学習進捗度の比較による学習空間の切り替えは、環境の状態を認識して行動選択する度に実施するため、ある一定期間の行動経験が必要な (4) と違い、1 回の行動経験により変化する学習進捗度にも即座に対応できる。本提案手法の全体空間と部分空間を並列にもつ考え方は、(3) (4) を組み合わせた文献 [27] の完全空間と制限空間を並列にもつ考え方と似ているが、文献 [27] では、制限空間から完全空間へ閾値を用いて切り替えること、切り替わった後は、完全空間のみを使う点が異なる。本手法では、全体空間と部分空間のそれぞれの学習進捗度を比較して選択するため、閾値は必要ない、また、各状態で選択するため、目標達成のために学習進捗に応じて全体空間と部分空間を行き来できる。そのため、環境変化が起きた場合においても、全体空間のみの場合よりも部分空間がより速く環境変化に対応し、学習を効率化する可能性をもつ。整理すると、本提案手法は、環境に対する学習空間を二つ用意し、これを二つとも学習させる。一つは、適正

行動を学習するのに十分な状態数をもつ学習空間（全体学習空間）を用いる。もう一つは、全体学習空間を圧縮する部分空間（部分学習空間）を用いる。全体空間は、状態数が多いので学習は遅いが、環境に対して適正行動を正確に対応づけできる。部分空間は、状態数が少ないので学習は速いが、環境に対して適正行動を粗く対応づけする。行動選択をする度に、この二つの学習空間のどちらかを使い、更新することで、学習が速く環境にも細かく対応づけする学習を目指す。我々は、これまでに全体空間と部分空間を同時に用いることで、学習効率化の可能性を検証してきた [28]。本論文では、正確で効率良く目標を達成するための行動学習手法を提案し、文献 [27] では示されなかった部分空間の有用性が異なる代表的な環境とその変化において、提案手法の有効性を検証する。具体的には、部分空間だけで学習できる環境、部分空間だけでは、半分しか学習できない環境、部分空間はまったく役に立たない環境において検証する。その結果、どのように部分空間を全体空間から作っても、本提案手法は、全体空間学習に対して有効であることを示す。さらに、実機を用いて学習実験をし、実機においてもシミュレーションと同様であることを検証する。ただし、本論文では、どのように全体空間から部分空間を圧縮すると効率が良いのかについては議論しない。

2. 提案手法

本研究の提案手法を工学的に述べる。本提案手法は、強化学習法の一つである Q-learning [29] を拡張するものである。提案手法は、行動選択数は同じであるが、状態数の異なる二つの Q-table をもつ。一つは、行動選択を環境に対して正確に割り当てるのに十分に細かい状態をもつ全体空間 Q-table、もう一つは、行動選択を環境に対して割り当てるのに最低必要とされる粗い状態をもつ部分空間 Q-table である。行動選択は、いずれか一方の Q-table を用いる。二つの Q-table を更新することで最適行動を学習する。これにより、学習速度が速く環境に対しても細かく対応付けできる学習を目指す。

一般の Q-learning は、Q-table という状態ごとにそれぞれの行動に対する評価に対応する Q 値をもつテーブルを一つだけもつ。それぞれの状態は、一般に環境空間を量子化し割り当てるものである。いかなる環境においても正確に最適行動に割り当てようと、Q-table の状態数を必要以上に多くしても、正確に状態ごとの最適行動を学習できる。しかし、状態ごとに最適行動を割り当てるために学習時間がかかる。その逆に、学習速度を早くするために、Q-table の状態数を必要以下に少なくすると、最適行動が確率的にしか割り当てられなくなり、正確に学習できなくなる。一般に、Q-table の最適な状態の割り当ては、学習する前には分からない。本研究は、十分に細かい状態をもつ状態数の多い全体空間 Q-table と最低限必要な粗さの状態をもつ状態数の少ない部分空間 Q-table を適宜用いることで、学習速度が速く環境に対しても細かく対応付けできる学習を目指す。

2.1 学習空間

提案手法は、二つの Q-table を用いる。一つは、すべての状態で適正行動が割り当てられるように十分細かく区切られた状態空間をもつ全体空間 Q-table であり、もう一つは、少なくともいくつかの状態で適正行動が割り当てられるように区切られた粗い状態空間をもつ部分空間 Q-table である。それぞれの状態割り当ては、同じ環境 env に対して、下式のように表現できる。

Fig. 1 Whole Q-table and Partial Q-table

$$p_s = func_p(env) \quad (1)$$

$$w_s = func_w(env) \quad (2)$$

ここで、 p_s , w_s は、部分空間と全体空間の状態をそれぞれ表し、 $func_p$, $func_w$ は、エージェントの置かれている環境を部分空間 Q-table と全体空間 Q-table のそれぞれどの状態に対応するかを表す関数である。本研究では、部分空間 Q-table の状態 p_s は、全体空間 Q-table の状態 w_s のいくつかの集合で表される。すなわち、部分空間の状態に対してそれが表す環境の範囲を rep_p 、全体空間の状態のものを rep_w とすると、下式のようになる。

$$rep_p(p_{s_i}) = rep_w(w_{s_j}) + rep_w(w_{s_k}) + \dots \quad (3)$$

Fig. 1 に全体空間 Q-table と部分空間 Q-table の例を示す。環境パラメータは、対象物体の色とその位置であり二次元空間中の点である。図の例では、全体空間 Q-table の状態は、環境である位置を 4 分割、色を 4 分割で、16 状態に分割しており、行動の選択肢が、4 とおりあるので、Q 値は 64 個になる。部分空間 Q-table の状態は、環境である位置のみ 4 分割するだけで、4 状態に分割しており、行動の選択肢は、同じ 4 とおりなので、Q 値は 16 個になる。この例では、部分空間 Q-table の一つの状態に対応する環境の範囲は、全体空間 Q-table の四つの状態に対応する環境の範囲と同じである。

2.2 Q-table の選択

全体空間 Q-table と部分空間 Q-table の二つの Q-table を学習しているので、行動選択をどのような方法で決めるかが問題になる。提案手法では、環境を全体空間 Q-table と部分空間 Q-table においてそれぞれの状態で表現し、それぞれの状態における行動の Q 値をみて、どちらの Q-table が有用かを判断、有用な Q-table を用いて行動選択する。ここで、Fig. 2 に、ある状態における行動の Q 値の例をあげる。有用な Q-table は、高い報酬を得るためにどの行動をとるべきかについて、確実性の高い情報をもつものと考え。Fig. 2 の中では、(a) が一番高い Q 値であるが、Q 値がすべて同じであるため、行動の選択確率もすべて同じとなり、どの行動をとるべきかが分からない。それに対して、(b) (c) は行動の Q 値が異なることで、行動の選択確率に差があり、どの行動をとるべきかが分かる。本論文では、より有用な Q-table は、(a) よりも (b) (c) のような Q-table であり、さらに行動選択確率に偏りがあればあるほど、どの行動をとるべきかについて確実性の高い情報をもつと考える。そこで、行動の選択確率の偏りを評価する指標として下式の平均情報量（情報エントロピー）を用いる。平均情報量は、状態 s における行動選択確率の偏りが大きくなればなるほど小さくなる。

$$H(s) = \sum_{a \in ACT} p(a | s) \log_2 \frac{1}{p(a | s)} \quad (4)$$

Fig. 2 Three examples of distribution of Q-values by each action at a state

Fig. 3 NS chart of the proposed method

すなわち、行動選択に全体空間 Q-table と部分空間 Q-table のどちらを使うかは、それぞれの状態における平均情報量が小さいほうを用いる。

2.3 Q-table の更新

Q-table の更新は、一般の Q-table の更新アルゴリズムと同様である。ここで、 r は報酬、 α は学習率、 γ は割引率を示す。

$${}^iQ(s_t, a) \leftarrow (1 - \alpha){}^iQ(s_t, a) + \alpha(r + \gamma V({}^i s_{t+1}))$$

$$i \in \{p : \text{partial}, w : \text{whole}\}$$

(5)

ただし、全体空間用、部分空間用の状態価値関数 $V({}^i s)$ は下式を用いる。部分空間用の価値関数は、一般のものと同等であるが、全体空間用は、学習速度を速めるため、有用ならば部分空間のものを使う。

$${}^pV({}^p s) = \max_{a \in ACT} {}^pQ({}^p s, a)$$

$${}^wV({}^w s) = \max_{i \in \{p, w\}} \max_{a \in ACT} {}^iQ({}^i s, a)$$

(6)

2.4 学習アルゴリズム

前節までで説明した学習アルゴリズムの概念を Fig. 3 に示す NS チャートを基に整理して説明する。

(1) 環境の設定：

シミュレーションで使う学習環境や学習パラメータなどを設定する。また、学習エージェントが採るべき最適行動を定義する。学習エージェントは、報酬を得ることで環境を知覚することができる。

(2) Q-table の初期化：

学習エージェントが使う全体空間 Q-table および部分空間 Q-table の Q 値を初期化する。

(3) Q-table の選択：

学習エージェントが全体空間 Q-table と部分空間 Q-table のどちらを用いて行動選択するかは、式 (4) で計算できる平均情報量 $H(s)$ を用いて判断する。その状態における全体空間 Q-table と部分空間 Q-table の平均情報量を計算し、

Fig. 4 An environment of the simulation

平均情報量が低い Q-table を行動選択のために選ぶ。

(4) 行動選択：

学習エージェントは、平均情報量により選択された iQ -table ($i \in \{p, w\}$) に対し、下式の Boltzmann 選択を使い時刻 t の行動 a_t を選択する。

$$p(a_t | {}^i s_t) = \frac{\exp(\frac{{}^iQ(s_t, a_t)}{T})}{\sum_{a \in ACT} \exp(\frac{{}^iQ(s_t, a)}{T})}$$

(7)

(5) 状態価値関数：

式 (6) に従い、それぞれの状態価値を求める。

(6) それぞれの Q-table の更新：

Q 値の更新関数式の式 (5) を式 (6) で求めた次状態の状態価値を用いてそれぞれ更新する。

上記アルゴリズムを報酬を得るまで繰り返し、それを 1 エピソードとして、何エピソードも繰り返すことで Q 値を更新し、各環境における適正行動を学習する。

3. シミュレーション実験

本章では、提案手法の有効性をシミュレーション実験により確認する。本提案手法は、全体空間 Q-table と部分空間 Q-table の二つの Q-table を用いる。その二つの Q-table の行動の数と種類は互いに同じであるが、状態の数と種類は異なる。全体空間 Q-table は、対象となる実験環境を十分正確に学習できるほど細かい状態をもち、部分空間 Q-table は、全体空間の状態を統合して学習に最低限必要な粗い空間表現をする状態空間をもつ。本論文では、対象の環境に対してどのような全体空間 Q-table と部分空間 Q-table を用意するかを議論するのでなく、用意した全体空間 Q-table と部分空間 Q-table に対して、シミュレーション環境がどのような場合に学習効率がどうなるかをシミュレーション実験により検証する。具体的には、シミュレーション環境が、(ケース 1) 部分空間 Q-table の状態表現の粗さでも十分表現できる場合、(ケース 2) 部分空間 Q-table では粗すぎ、全体空間 Q-table では細かすぎる場合、(ケース 3) 全体空間 Q-table の状態表現の細かさがないと学習できない場合の三つの場合において本提案手法の有効性を調べる。ここでは、二つの Q-table を用いる本提案手法の有効性を比較するために、全体空間 Q-table のみと部分空間 Q-table のみの単純な従来手法二つもシミュレーション実験をする。

3.1 シミュレーション環境

シミュレーションに用いる環境として、部分空間がどれだけ環境を正確に表現しているか分かりやすい恣意的な環境を用意する。具体的には、Fig. 4 に示す環境を考える。この環境では、Cell N 個のセルが円環状に並べられる。学習エージェントは、時刻 $t \in \{0, 1, 2, \dots\}$ が変わるとに一つずつセル $cell_t$ を時

計回りに移動する。学習エージェントは、時刻 t において、必ず 0 から $N-1$ の整数のいずれかの値 n_t の札を上げる。学習エージェントからみえるところに、 $ClrN$ 種類の旗があり、時刻ごとに掲げる旗 clr_t がランダムに切り替わる。Fig. 4 では、FlagA が立てられている状況である。学習エージェントは、そのときの立てられている旗の種類とセルの場所であらかじめ決められる行動、すなわち数 $n(cell_t, clr_t)$ の札を上げると、その場で段を上がり、そうでない場合、一番下の段へ下りる。そして、 $StageN$ 段分上ると報酬がもらえる。学習エージェントが、時刻 t のときにいる $cell_t$ は、0 時刻のときに、0 セルにいるとすると、 $cell_0 = 0$ と記述され、時刻 t の $cell_t$ では、下式で記述される。

$$cell_t = t \bmod CellN \quad (8)$$

また、時刻 t の報酬 R_t は、下式で記述される。

$$R_t = \begin{cases} 1 & \text{if } (\sum_{i=t-m+1}^t tf_i) = StageN \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$tf_i = \begin{cases} 1 & \text{if } n(cell_i, clr_i) = n_i \\ 0 & \text{otherwise} \end{cases}$$

上記の環境において、 $StageN=4$ として、4 連続で適正行動を実行できると報酬がもらえるものとし、全体空間と部分空間を以下のように定義する。

・全体空間 Q-table：

学習エージェントの場所 (20) と旗 (4) と学習エージェントの行動 (4) の 3 軸で表現される。

・部分空間 Q-table：

学習エージェントの場所 (20) と学習エージェントの行動 (4) の 2 軸で表現される。

この環境は、全体空間 Q-table を用いれば、どの位置にいても、どの旗が揚がっても、各状態ごとに適正行動を学習することができる。しかし、部分空間 Q-table のみを用いると、どの位置にいても、同じ場所なら旗の種類にかかわらず適正行動が一意に決められていないと、各状態ごとに適正行動を学習することができない。同じ場所でも、掲げる旗により適正行動が異なると、確率的にしか適正行動を学習することができない。

そこで、本研究では、同じ場所において、一意に適正行動の決まる環境から旗の種類ごとに適正行動の異なる環境まで、部分空間 Q-table の有効率を変更することで、本提案手法の Dual Q-table の有効性がどのように変化するかを観測する。

Q 値の初期値を 0.0、報酬 r を 1.0、学習率 α を 0.08、割引率 γ を 0.8 に設定する。行動選択手法として Boltzmann 選択を用い、温度 T を 0.10 に設定する。

3.2 比較のための変更条件

環境のすべてが部分空間 Q-table のみで学習できるケース 1、環境の半分を部分空間 Q-table のみで学習できるケース 2、環境を部分空間 Q-table では学習できないケース 3 の合計 3 種類の環境でシミュレーション実験をし、本提案手法の特性をみる。三つの環境とも、1,000 エピソード目から環境が変化する。

ケース 1：学習環境のすべてを部分空間のみで表現できる場合（部分空間による推論が完全に有効な場合）

この環境では、学習エージェントの適正行動のすべてが部分空間のみで表現できる。具体的には、Fig. 5 に示すように、偶数セルにいるときは、行動 0、奇数セルにいるときは、行動 1 をとればよく、学習エージェントの適正行動はすべて、旗に関係なく

Fig. 5 Case 1: An environment that the partial Q-table is valid for all the states

Fig. 6 Case 2: An environment that the partial Q-table is valid for the half states

学習エージェントの位置情報のみで決まる。また、環境変化後も、偶数セルと奇数セルの設定が入れ替わるだけで、位置情報のみで決まる。言い替えると、エージェントの適正行動はすべて状態をセル位置のみで記述する部分空間 Q-table のみで表現できる。この場合、部分空間を用いる推論が完全に有効であると考えらる。

ケース 2：学習環境の半分を部分空間のみで表現できる場合（部分空間による推論が半分有効な場合）

この環境では、学習エージェントの適正行動のうち、半分が部分空間のみで表現でき、残り半分は部分空間では確率的にしか表現できない。具体的には、Fig. 6 に示すように、学習エージェントが偶数セルにいるときは、行動 0、奇数セルにいるときは、旗により、行動 0 か、行動 1 が適正行動であると決まる。言い替えると、学習エージェントの適正行動のうち、半分は部分空間 Q-table のみで表現できるが、残り半分は、部分空間 Q-table のみでは確率的にしか決められず全体空間 Q-table を必要とする。ここでの環境変化は、部分空間 Q-table に対しては変化がなく全体空間 Q-table に対してのみ変化のあるものとする。この場合、部分空間 Q-table による推論のうち、半分は有効であると考えられ、提案手法がほかの二つの従来法と比べて有効であることが期待できる。

ケース 3：学習環境を部分空間では表現できない場合（部分空間による推論がまったく有効でない場合）

この環境では、学習エージェントの適正行動は部分空間のみでは表現できない。具体的には、Fig. 7 に示すように、学習エージェントの適正行動はすべて、学習エージェントがいるセルの位置と旗により決まる。ここでの環境変化は、部分空間のみでは学習できない別の環境へ変化する。言い替えると、学習エージェントの適正行動はすべて部分空間 Q-table のみで表現できず、全体空間 Q-table を必要とする。この場合、部分空間 Q-table による推論はまったく役に立たないと思われる。

3.3 シミュレーション実験結果と考察

部分空間 Q-table と全体空間 Q-table の二つの Q-table をもつ提案手法と部分空間 Q-table のみの従来法と全体空間 Q-table のみの従来法との比較シミュレーションの実験結果を示す。ケース 1 から 3 までのそれぞれの実験結果のグラフは、1,000 エピソード目に環境を変更する 0 から 1,990 エピソード目までの

Fig. 7 Case 3: An environment that the partial Q-table is not valid

Fig. 8 The results of Case 1

Fig. 9 The results of Case 2

1,000 試行シミュレーション実験の平均の結果である。

Fig. 8 にケース 1 のシミュレーション実験結果を示す。ケース 1 で、提案手法と部分空間 Q-table のみの従来法、および全体空間 Q-table のみの従来法の 3 手法とも適正な値を得ることができた。部分空間 Q-table のみの従来法と提案手法はほぼ同じ早さで、全体空間 Q-table のみの従来法よりも早く収束した。部分空間 Q-table が環境に対して適しているため、効率良く環境を学習することができた。全体空間 Q-table は同じ状態とみなして良い環境に対して余分に行動を試すため、効率良く環境を学習することができなかった。提案手法は、初期状態からの学習は、グラフ上で部分空間 Q-table のみの従来法と区別がつかないほどの速度で収束した。しかし、環境変化に関しては、全体空間 Q-table のみの従来法よりは良いものの、部分空間 Q-table のものよりも幾分遅くなった。この結果より、部分空間の推論が完全に正しい。ケース 1 において、提案手法は、全体空間 Q-table のみの従来法よりも、部分空間のみの学習に近く有効であることを示した。

Fig. 9 にケース 2 のシミュレーション実験結果を示す。ケース 2 で、提案手法と全体空間 Q-table のみの従来法の 2 手法は最適値を得ることができた。しかし、部分空間 Q-table のみの従来法は、最適値を得ることができなかった。これは、半分

Fig. 10 The results of Case 3

の環境は部分空間で表現できるため、学習エージェントは適正行動を決められるが、残り半分の環境は全体空間でないと表現できないため、学習エージェントは適正行動を決められずランダムに行動を選択した結果である。提案手法は、学習初期から部分空間 Q-table に頼り知識から推論し、行動を選択するため、全体空間 Q-table のみの従来法よりも早く収束した。そして、全体空間 Q-table でしか表現できない環境は全体空間 Q-table に頼り、行動を選択するため、最適値をとることができた。ケース 2 において、提案手法は、全体空間 Q-table のみの従来法よりも、効率良く学習できることを示した。

Fig. 10 にケース 3 のシミュレーション実験結果を示す。ケース 3 で、提案手法と全体空間 Q-table のみの従来法の 2 手法は最適値を得ることができた。しかし、部分空間 Q-table のみの従来法は、最適値を得ることができなかった。これは、すべての環境は全体空間でないと表現できないため、学習エージェントは適正行動を決められずランダムに行動を選択した結果である。提案手法は、部分空間 Q-table より全体空間 Q-table を多く採用することで環境を学習することができた。ほとんど同じ学習曲線を描く、提案手法と全体空間 Q-table 手法との間で、環境変化において、提案手法が全体空間 Q-table よりも、騙されるピークを低くできたのは、部分空間 Q-table のおかげであると考えられる。これは、変化後の環境に適応している途中は、常に偽を答える Q-table よりもランダムに真と偽を答えるほうが役に立つためであると考えられる。ケース 3 において、提案手法は全体空間 Q-table の従来法とほぼ同じくらい有効であることを示した。

環境設定を代表する三つのケース、すなわち、学習環境のすべてを部分空間で表現できるケース、学習環境の半分は部分空間で表現でき残り半分は確率的でないと表現できないケース、学習環境のすべてを全体空間でないと表現できないケースにおいて、全体空間 Q-table のみの手法よりも、提案手法が劣ることはないことが検証できた。これにより、用意する部分空間 Q-table にかかわらず提案手法が全体空間 Q-table のみの手法よりも有用であることをシミュレーションにおいて示せた。

4. 数 理 解 析

本章では、提案手法の学習の収束速度に着目し、3 章で用いた三つの学習空間において従来手法と比較する。ケース 1 では部分空間のみの従来手法に対してまったく遅れがないこと、ケース 3 では全体空間のみの従来手法に対してほとんど遅れがないこと、ケース 2 ではケース 1 とケース 3 の間になることを説明する。本解析では、ケース 1 と 2 についてはアルゴリズムの流

れからの説明のみとし、ケース3のみ数理的に説明する。また、本論文では学習環境の変化が起きる前までについてを解析する。

4.1 ケース1：学習環境を部分空間のみで表現できる場合

ケース1における提案手法の学習の収束速度が、部分空間のみの従来手法に対して、まったく遅れないことをアルゴリズムの流れから説明する。

i, j は、適切な整数、 k は、 j とは異なる整数とする。式(3)で示すように、部分空間のある一つの状態 $P S_i$ が、全体空間の複数の状態 $W S_j, W S_k, \dots$ に対応すると考える。ケース1の場合、部分空間の状態 $P S_i$ と全体空間の状態 $W S_j, W S_k, \dots$ の適正行動が一致する。ゆえに、部分空間と全体空間のそれぞれの状態において、 Q 値の期待値は、その状態を経験する回数と同じならば一致する。また、 Q 値の期待値は、経験する回数が多いほど、 Q 値の収束値に近づく。経験する回数が多いほど、その状態における適正行動の Q 値が大きくなり、 Q 値の偏りが大きくなる。部分空間の状態 $P S_i$ を経験する回数は、全体空間のそれぞれの状態 $W S_j, W S_k, \dots$ を経験する回数の和に等しい。ゆえに、部分空間の状態 $P S_i$ の経験回数は、どの全体空間の状態 $W S_j, W S_k, \dots$ よりも常に多く Q 値の偏りが大きい。 Q 値の偏りが大きくなると、 Q 値から計算する行動の選択確率も偏りが大きくなる。提案手法においては、行動選択確率の偏りが大きい空間を用いて行動を選択するため、ケース1の環境では、常に部分空間が行動選択に用いられるので、実質部分空間のみの従来法と同じ枠組みになり、同じ速さで収束する。

4.2 ケース3：学習環境を部分空間では表現できない場合

ケース3における提案手法の学習の収束速度が、全体空間のみの従来手法に対してほとんど遅れないことを説明する。ケース3の学習環境において、部分空間はすべての状態において適正行動が学習できないため、部分空間に誤誘導される分だけ、提案手法は全体空間のみからなる従来手法よりも学習の収束が遅くなる。ここでは、提案手法の行動選択空間が部分空間から従来手法と同じ全体空間に切り替わり、その間の誤誘導が数エピソード程度であることを数理的に説明する。

4.2.1 学習空間

解析に用いる学習空間は3章と同様とするが、簡単化のため、セル数 $CellN$ は1個、各セル旗の種類 $ClrN$ は2種類、報酬取得のために上がる段数 $StageN$ は1段、すなわち、状態に対する適正行動を1回すると報酬が取得できる。Fig. 11に示すように、旗により学習エージェントの適正行動が異なるため、部分空間では適正行動を学習できない。学習時の Q 値の初期値を0、報酬 r を1、学習率 α を0.08、割引率 γ を0.8、行動選択手法に Boltzmann 選択を用い、温度 T を0.2に設定する。

本解析では、提案手法とは異なるが、簡単化のため以下の三つを実施する。一つ目は、本設定では任意の状態から1ステップの適正行動で報酬が与えられるので、ほとんど影響のない Q 値の更新式の次状態の価値を0とする。二つ目は、誤誘導を維持することとなるが、部分空間の誤誘導により選択行動が失敗しても選択確率を変更しない。もちろん、実際には行動選択失敗により Q 値は低下し、平均情報量は変化するので、行動選択

空間の切り替えは発生する。三つ目は、行動選択空間の切り替わり判断は、平均情報量で比較する本手法より悪く見積もり、部分空間における各行動の目標達成回数が一致するときとする。

4.2.2 切り替わり時の Q -table

1 エピソード目は、全体空間、部分空間とも全 Q 値が初期値で平均情報量が同じため、提案手法における行動選択には部分空間 Q -table が用いられる。本解析では、1 エピソード目に FlagA の状況にて目標達成した後、2 エピソード目以降で、誤誘導されながら行動選択空間が部分空間から全体空間に切り替わるエピソード回数の期待値を算出する。1 エピソード目で FlagB の状況にて目標達成した場合においても、以降の説明で FlagA と FlagB が入れ替わるのみで算出方法は同じである。

提案手法においては、2 エピソード目以降の目標達成するまでの旗の系列は、行動選択空間の切り替わり方により三つに分類される。三つの系列は、正規表現である。“.” は一つの記号、“+” は直前の記号が1回以上、“*” は0回以上続くことを示すものを用いると、FlagA が立つ状況のみを経験して目標達成する ($A+$ 系列)、FlagB が立つ状況を1回は経験して最後は FlagA で目標達成する ($. * B . * A$ 系列)、最後は FlagB で目標達成する ($. * B$ 系列) と表現される。 $(A+)$ 系列の場合、部分空間にて高い Q 値をもつ行動0の失敗が一度も起こらず、部分空間の平均情報量は変化しないため、目標達成時の行動選択空間は部分空間であり、全体空間の Q 値のみでなく、部分空間の Q 値が強化される。 $(. * B . * A)$ 系列の場合、FlagB の状況にて部分空間で高い Q 値をもつ行動0の失敗が起き、部分空間の平均情報量が増加するため、目標達成時の行動選択空間は全体空間である。この場合、部分空間の Q 値は更新されず、FlagA の状況、つまり全体空間の一部の状態において行動選択空間が全体空間に切り替わる。そのため、これ以上 FlagB の状況により誤誘導となるように部分空間が強化されることはない。その後、FlagB において目標達成するとき、部分空間の行動選択により誤誘導される。 $(. * B)$ 系列の場合、部分空間と比べて、FlagB の状況において全体空間には高い Q 値をもつものもなく Q 値のバラつきは小さいため、目標達成時の行動選択空間は部分空間である。この場合、部分空間の Q 値が更新され、行動1の Q 値が行動0に近づきバラつきが減る。この3種類の系列の繰り返しにより、やがては、行動選択空間が部分空間から全体空間に切り替わる。具体的には、1 エピソード目は、必ず ($. * A$ 系列) であり、FlagA の状況で部分空間の Q 値が強化されなくなるのは、エピソード系列： $(. * A)$ 系列 ($A+$ 系列) $* ((. * B . * A)$ 系列) $| (. * B)$ 系列) のときである。ここで、“|” は“または”を意味する。このエピソード系列の後、1 エピソード目とエピソード系列中の ($A+$ 系列) の回数分だけ ($. * B$ 系列) で報酬を得るまでは、FlagB の状況において行動選択空間が部分空間から全体空間へ切り替わらず、誤誘導される可能性がある。具体例として、2 エピソード目終了時の Q -table 更新結果を Fig. 12 に示す。図中の $Q(N_{Goal})$ は、 N_{Goal} 回目目標達成して更新された Q 値を、 P_{Goal} (旗の系列) は、旗の系列を経験して目標達成する確率を示す。図中の ($. * A$ 系列) ($A+$ 系列) 後では、部分空間の Q 値は強化され、FlagB の状況で2回報酬を得るまで行動選択空間が部分空間になり、FlagB の状況において行動0が適正行動であると誤誘導される。さらに、3 エピソード目が ($A+$ 系列) だと、FlagB の状況で3回報酬を得るまで誤誘導される。 $(. * A)$ 系列 ($. * B . * A$ 系列) 後では、FlagA の状況においては、行動選択空間が全体空間に切り

Fig. 11 An environment for the mathematical analysis

Fig. 12 Updating the dual Q-table

替わる。これにより、部分空間はもうこれ以上誤誘導となる方向へ強化されず、FlagB の状態で 1 回報酬を得れば誤誘導されなくなる。(*A 系列) (*B 系列) 後では、FlagA と FlagB のいずれの状況においても、行動選択空間が全体空間に切り替わる。ゆえに、もう誤誘導されない。各系列で目標達成する確率は無限級数の公式を用いて、式 (10)、式 (12)、式 (13) のように算出できる。

$$P_{Goal}(A+) = P_A P(0|A) \sum_{k=0}^{\infty} \{P_A(1 - P(0|A))\}^k \quad (10)$$

$$= \frac{P_A P(0|A)}{1 - P_A(1 - P(0|A))}$$

$$P_{Goal}(*A) = P_A P(0|A) \sum_{k=0}^{\infty} (1 - P_A P(0|A) - P_B P(1|B))^k \quad (11)$$

$$= \frac{P_A P(0|A)}{P_A P(0|A) + P_B P(1|B)}$$

$$P_{Goal}(*B, *A) = P_{Goal}(*A) - P_{Goal}(A+) \quad (12)$$

$$P_{Goal}(*B) = \frac{P_B P(1|B)}{P_A P(0|A) + P_B P(1|B)} \quad (13)$$

$$P_{Goal}(B+) = \frac{P_B P(1|B)}{1 - P_B(1 - P(1|B))} \quad (14)$$

ただし、 P_A は各ステップにおいて FlagA が立つ確率、 P_B は FlagB が立つ確率、 $P(0|A)$ は FlagA の状況にて適正行動 0 をとる確率、 $P(1|B)$ は FlagB において適正行動 1 をとる確率である。ここでは、 $P_A = P_B = 0.5$ とする。

4.2.3 切り替えエピソード回数の期待値

部分空間による誤誘導は、行動選択空間が全体空間に切り替わるまで起こる可能性がある。そこで、誤誘導されるエピソード回数の期待値を、簡単化のため、初回から全体空間に切り替わるまでのエピソード回数の期待値として多めに見積もる。切り替えは、1 エピソード目と (A+) 系列でゴールし続けるエピソード回数の和 N_{AG} に対して、(*B) 系列で N_{AG} 回ゴールするエピソード数が必要である。これは、(A+) 系列でゴールし続けないと、FlagA での行動決定空間が全体空間に切り替わり、その後は、(*B) 系列で最大同数ゴールすれば、FlagB での行動決定空間も全体空間に切り替わるので、全体空間だけで学習する従来手法と同じになるまでにかかるエピソード回数を求めるものである。 P_{Goal} (系列) の確率は、 N_{AG} の数にも依存するので、ここから P_{Goal} (系列, N_{AG}) と記述する。す

ると、(*B) 系列で N_{AG} 回ゴールするのに期待されるエピソード数の概数 N_{exBG} は、 $N_{exBG} = \frac{N_{AG}}{P_{Goal}(*B, N_{AG})}$ とかける。求めるべき切り替えのエピソード回数の期待値 N_{ep} は次式となる。

$$N_{ep} = \sum_{N_{AG}=1}^{\infty} P_{contA}(N_{AG}) \left(N_{AG} + \frac{N_{AG}}{P_{Goal}(*B, N_{AG})} \right) \quad (15)$$

ここで、 $P_{Goal}(*B, N_{AG})$ の代わりに $P_{Goal}(B+, N_{AG})$ を用いて多めに見積る。 $P_{Goal}(*B, N_{AG}) = P_{Goal}(B+, N_{AG}) + P_{Goal}(*A, *B, N_{AG})$ であり、 $P_{Goal}(*A, *B, N_{AG})$ は、学習が進むと FlagA の状況でゴールするようになって 0 に近づく。また、 $P_{contA}(n)$ は、1 エピソード目で (*A 系列) でゴールし、2 エピソード目以降に連続で (A+ 系列) のみで n エピソード目までゴールし、 $n+1$ エピソード目は、(A+ 系列) 以外でゴールする確率とする。これは $P_{Goal}(A+, n)$ を用いて計算できる。

$$P_{contA}(n) = (1 - P_{Goal}(A+, n+1)) \prod_{k=2}^n P_{Goal}(A+, k) \quad (16)$$

ただし、(A+ 系列) 以外でゴールする確率 $(1 - P_{Goal}(A+, n+1))$ は、 $P_{Goal}(*B, *A, n+1) + P_{Goal}(*B, n+1)$ である。 $P_{contA}(n)$ は指数関数的に減り、ここでの設定パラメータでは $P_{contA}(1) = 0.63$, $P_{contA}(2) = 0.22$, $P_{contA}(3) = 0.087$ となり、 $P_{contA}(20)$ では 2.9×10^{-7} と非常に小さくなる。そこで、 N_{AG} を 1 から 19 までの和 Nep_{1-19} と 20 以降の和 Nep_{20-} に分けて計算する。 Nep_{1-19} は 1 エピソードずつ式 (14)、式 (15)、式 (16) を用いて数値計算すると $Nep_{1-19} = 7.9$ となる。 Nep_{20-} は式 (15) を展開して求めていく。 $P_{contA}(N_{AG})$ は、式 (10) より $P_{Goal}(A+)$ の最大が $1/2$ であることから、少なくとも 1 エピソードごとに $1/2$ 倍ずつされていく。 $P_{Goal}(*B, N_{AG})$ は、行動選択確率をボルツマン選択を用いて算出することから最小値をもつ。今回のパラメータにおける最小値 6.6×10^{-3} を用いる。式 (15) の第 1 項は、第 2 項に比べて $1/1,000$ 程度と十分小さいため無視できる。式 (15) の第 2 項は次式に展開でき、無限級数の和の公式を用いて求めると 0.0029 となる。

$$Nep_{20-} \approx \sum_{N_{AG}=20}^{\infty} P_{contA}(N_{AG}) \frac{N_{AG}}{P_{Goal}(*B, N_{AG})} \quad (17)$$

$$= P_{contA}(20) \frac{20}{P_{Goal}(*B, N_{AG})} \left(1 + \frac{21}{20} \frac{P_{contA}(21)}{P_{contA}(20)} + \dots \right)$$

$$\leq 2.9 \times 10^{-7} \frac{20}{6.6 \times 10^{-3}} \left\{ 1 + \left(\frac{1.05}{2} \right)^1 + \left(\frac{1.05}{2} \right)^2 + \dots \right\}$$

$$= 0.0029$$

ゆえに、 $Nep = Nep_{1-19} + Nep_{20-} \approx 7.9$ となる。ここで、Q-table の学習収束を適正行動の選択確率が 99% 以上になることとすると、今回の学習条件において、その確率が 99% 以上になるまで Q 値が上がるには、FlagA と FlagB の各状況にて 30 回ずつ目標達成する必要がある、学習収束には 60 エピソードが最低限必要となる。それに対する切り替えエピソード回数の概算期待値は 13% 程度である。すなわち、提案手法において誤誘導の可能性をもつエピソード回数は、学習初期の数エピソード程度であることを数理的に確認できた。

ここから、行動数 N_{act} と圧縮状態数 N_c が 2 より大きいとき、行動選択空間の切り替えエピソード数 N_{ep} がどうなるかを簡単に述べる。式 (15) より N_{ep} の値は、部分空間を強化し続けたエピソード回数 N_{AG} と N_{AG} 回 FlagB でゴールして FlagB で部分空間から全体空間に行動選択 Q-table を切り替えるエピソード回数の期待値の和と $P_{contA}(N_{AG})$ の生起確率を掛け合わせるものの総和である。行動数 N_{act} を大きくするほど、 $P(0|A)$ の値は小さくなり、 $P_{contA}(N_{AG})$ の値は $N_{AG} = 1$ のとき、より大きく、 $N_{AG} > 1$ では、より小さくなる。それに対して、 $P_{Goal}(*B, N_{AG})$ の値は、 N_{act} に反比例する程には小さくならないので、行動選択が N_{act} 倍に広がり、収束が遅くなることを考えれば、割合として、 N_{ep} は小さくなる。圧縮状態数 N_c を部分空間のすべてで適正行動が学習できない必要条件である $N_c \leq N_{act}$ の範囲で大きくすると、やはり $P_{contA}(N_{AG})$ の値は $N_{AG} = 1$ のとき、より大きく、 $N_{AG} > 1$ では、より小さくなる。それに対して、各エピソード回数は、FlagB だけでなく FlagC, D, ... でもゴールする期待値を考えなくてはならない。しかし、これらを考慮してもエピソード回数は、最大でも N_c 倍程度にもならないので、学習空間が N_c 倍に広がることを考えれば、割合としては、 N_{ep} はやはり小さくなる。

4.3 ケース 2：学習環境を部分的に部分空間で表現できる場合

ケース 2 における提案手法の学習の収束速度が、ケース 1 とケース 3 の間になることを簡単に説明する。

部分空間の状態 pS_i が全体空間の状態 wS_j , wS_k , ... に対応するとする。また、状態 wS_j のほうが、 wS_k より先に学習するとする。ケース 2 においては、部分空間 pS_i と全体空間の状態 wS_j に対応する状況で学習した結果が、部分空間 pS_i と全体空間の状態 wS_k に対応する状況において必ずしも適正行動となるわけではない。適正行動でない場合は、全体空間の状態 wS_k で再学習する必要がある、学習効率はケース 1 より下がる。また、部分空間 pS_i と全体空間の状態 wS_j に対応する状況で学習した結果が、部分空間 pS_i と全体空間の状態 wS_k に対応する状況において必ずしも不適正な行動となるわけでもない。適正行動となる場合は、部分空間の学習結果を用いて、全体空間の状態 wS_k で適合行動が学習強化され、学習効率はケース 3 より上がる。ゆえに、学習速度はケース 1 と 3 の間になる。

5. 実機実験

本章では、実機実験により提案手法の有効性を確認する。コンピュータシミュレーションにおいて、実機で起こる現象をシミュレートしきれているわけではない。本章では提案手法がコンピュータシミュレーションと同様の振る舞いをみせることを確認することで、提案手法が実環境でも有効であることを示す。前章のシミュレーション実験と同様に、提案手法は、全体空間 Q-table と部分空間 Q-table の二つを用い、それぞれの Q-table は行動の数と種類が同じで状態の数と種類が異なる。シミュレーション実験のケース 2 のように、学習環境の一部が部分空間のみで表現できる場合について実験する。全体空間 Q-table のみと部分空間 Q-table のみの単純な従来手法の二つと提案手法の学習効率を比較し、シミュレーションの結果と同様に提案手法の有効性を示す結果が得られるかを検証する。

5.1 実機環境

実機は、Fig. 13 に示す自律移動ロボット MieC を用いる。MieC は三重大学大学院工学研究科機械工学専攻知能ロボティク

Fig. 13 The autonomous mobile robot: MieC

Fig. 14 The environment of the experiments

Fig. 15 The movement of the color target

Fig. 16 Photograph of an experimental environment

ス研究室で開発された自律移動ロボットで、移動機構として 2 本の無限軌道を用いる。2 本の無限軌道は、二つのモータにより、それぞれ独立に駆動される。外部センサとして CCD カメラを搭載し、外部通信には無線 LAN を用いる。さらに、CPU カードと FPGA カードを搭載しており、画像処理などは CPU カードが処理を担当し、モータ制御などの処理は FPGA カードが担当する。ロボットが動作する環境は、大きさ $1.20 [m] \times 0.91 [m]$ のプラ舟を用い、内側には白色のプラスチック版を張り付ける。タスクは、ロボットの動作環境に表示された色をもつターゲットを、ロボットのカメラ画像の指定位置で認識することを目標とする。ターゲットは、Fig. 14 のようにプロジェクターと鏡を用いて、動作環境に投影することで表示し、動作環境に一つのみ存在する。ロボットが目標達成すると、ターゲットの投影位置は規則的に、色は不規則に変化させる (Fig. 15)。ターゲットの投影位置はロボット動作環境の四隅、色は赤・青・緑・黄色の 4 色のいずれかである。Fig. 16 に、実験の様子を実機の上部から撮影した写真を示す。報酬の取得状態について説明する。報酬は、ロボットのカメラ画像において、ターゲットの中

Fig. 17 States on camera image

Fig. 18 States of goals

Fig. 19 Action set of the autonomous mobile robot

心位置がカメラ画像を分割したうちのあらかじめ決められる位置となると、ロボットに与えられる。Fig. 17のように、カメラ画像の分割数は縦3×横5とし、左上から順に番号を割り振る。実機実験においても、学習途中での目標達成条件を変更して環境を変化させる。849エピソード目までの環境変更前は、赤色・青色のターゲットは6番の位置に、緑色・黄色のターゲットは8番の位置に映ったときに、850エピソード目以降の環境変更後は、赤色・青色のターゲットは8番の位置に、緑色・黄色のターゲットは6番の位置に映ったときに、ロボットは報酬を得る (Fig. 18)。カメラ画像の横方向の分割については、カメラ画像取得処理の遅れ等で各色の目標位置が隣り合う状態と誤認識されないように、つまり、7番の位置を確実に認識させるため、横幅をその左右の3倍としている。縦方向の分割については、上下方向動作の学習難易度を上げるため、目標位置6, 8番を含む5~9番の高さをその上下の1/2としている。ロボットの行動集合は、Fig. 19の矢印のパターンで示すように、八つの行動で構成する。動作速度は一定とし、その行動により、環境が別の状態に移るまで続け、移るまでを一ステップとする。上記の環境において、全体空間と部分空間を以下のように定義する。全体空間と部分空間共に位置情報として画像分割数の15に1を加えるのは、ターゲットが画像に映らない状態を表現するためである。

・全体空間 Q-table :

ロボットのカメラ画像にターゲットが映る位置 (15+1) とターゲットの色 (4)、ロボットの行動 (8) の3軸で表現する。

・部分空間 Q-table :

ロボットのカメラ画像にターゲットが映る位置 (15+1)、ロボットの行動 (8) の2軸で表現する。

前章のシミュレーション実験と同様、全体空間を用いれば、各状態ごとに適正行動を学習することができるが、部分空間のみを

Fig. 20 Actual experimental results from 0th to 1700th episode

Fig. 21 Actual experimental results from 0th to 300th episode

用いると、確率的にしか適正行動を学習できない。また、ターゲットの目標位置は色が4色のあるうちの2色ずつを同じ位置にしておき、どの色も画像の中心位置の左右どちらかの位置にしているため、目標達成のために、ターゲットをカメラ画像の周辺から中心方向に移動させる行動戦略は、部分空間のみで表現できる条件となる。4色それぞれを区別する全体空間では細かすぎ、色を区別できない部分空間では粗すぎる条件である。本実験では、前章のシミュレーション実験のケース2と同等の条件において、提案手法の振る舞いが同様であることを確認することで、実環境での有効性を検証する。学習空間のQ値の初期値を0.0、報酬 r は正の報酬を1.0、負の報酬を-1.0、学習率 α を0.1、減衰率 γ を0.8に設定する。負の報酬はロボットが動作環境であるプラ舟の壁に衝突し続けるのを防ぐため、一定時間行動しても状態が推移しない場合に与えられる。行動選択手法としてBoltzmann選択を用い、温度 T を0.1に設定する。

5.2 実機実験結果と考察

全体空間 Q-table と部分空間 Q-table の二つの Q-table をもつ提案手法と部分空間 Q-table のみの従来法と全体空間 Q-table のみの従来法との比較結果を示す。グラフは、それぞれの学習法で1,700エピソード目まで3試行分実施し、ステップ数の積算値を平均した結果である。環境変化として、850エピソード目からターゲットの目標認識位置の変更を実施した。

Fig. 20~Fig. 22に実験結果を示す。横軸はエピソード数、縦軸は行動ステップ数の累積を示す。Fig. 20が全エピソード分、Fig. 21が0から300エピソード分、Fig. 22が850から1,700エピソード分、かつ、学習状況を比較しやすくするために850エピソード目からステップ数の積算値をカウントしたグラフである。ステップ数の積算値のため、グラフの最終的な傾きが目標達成に必要なステップ数、傾きが一定になるまでが学習収束の速さを示す。まず、環境変化前の849エピソード目までの結

Fig. 22 Actual experimental results from 850th to 1700th episode, when counting the steps from 850th episode

果を考察する。最終的なグラフの傾きより、提案手法と全体空間 Q-table のみの従来法の 2 手法は最適値を得ることができたが、部分空間 Q-table のみの従来法は、速く収束するものの最適値を得ることができなかった。これは、ターゲットの色に関係なくカメラ画像の中心方向、目標位置近くに移動させる適正行動は速く獲得できるものの、色により指定された目標位置に移動させる適正行動が確率的にしか決められないためである。提案手法は学習初期は部分空間を用い、全体空間でしか表現できない環境は全体空間を用いることで、全体空間のみの従来法よりも速く収束し、かつ、最適値を得ることができた。次に環境変化後の 850 エピソード目以降の結果を考察する。ターゲットの色を区別しない部分空間 Q-table のみの従来法には、今回の環境変化が起こってもその環境変化を認識できない、つまり関係ないため、グラフの傾きに変化はなかった。提案手法は環境変化後も環境変化前と同様に、全体空間 Q-table のみの従来法よりも速く収束し、かつ、最適値を得ることができた。環境変化前と後のどちらも、収束の速さと結果の正確性について、提案手法と二つの従来手法は、前章のシミュレーション実験のケース 2 と同様の結果となった。つまり、実機実験においても、提案手法は、有用性があるといえる。

6. おわりに

本論文では、状態を細かく表現する全体空間 Q-table と粗く表現する部分空間 Q-table を組み合わせて効率良く正確に環境を学習する Q-learning を提案した。一般に、状態を粗く表現する Q-table では、環境に対する学習速度は速いが、正確に適正行動を学習できず、状態を細かく表現する Q-table では、正確に適正行動を学習できるが、環境に対する学習速度が遅い。本提案手法では、全体空間 Q-table とその状態空間を圧縮する部分空間 Q-table を同時に学習し、学習が進んでいる Q-table をその都度選択し、それを行動選択に用いることで、学習速度が速く、正確に適正行動を学習できるものを目指した。本提案手法の挙動確認のため、部分空間が学習環境を完全に表現しているもの、一部有用なもの、まったく有用でないものの代表的な三つの学習環境においてシミュレーション実験をした。提案手法は、有用な部分空間を用いれば、学習効率があがることはもちろんだが、まったく有用でない部分空間を用いても学習効率がほとんど下がらないことを示した。また、この学習効率の効果を静的で環境が変化しないものではあるが、数理解析においても検証した。さらに、簡単な環境ではあるが、実機実験においても有効であることを示した。本研究では、全体空間と部分空

間そのものの作成の仕方は検討していない。しかしながら、全体空間は正確な適正行動学習のために十分に細かい状態が表現できるように、部分空間は明らかに必要な状態を区別できるように、というような方針で作成すれば、提案手法が利用できる。今後は、この結果を用いて、三つ以上の Q-table を用いるものや構造化をするものなどに発展させたい。

謝辞 本研究の一部は科研費(16K12788)の助成を受けた。

参考文献

- [1] R.S. Sutton and A. Barto: Reinforcement Learning: An Introduction. A Bradford Book, The MIT Press, 1998.
- [2] 今福啓: “環境の変化に適応するマルチエージェントの学習法”, 人工知能学会論文誌, vol.21, no.2, pp.153-166, 2006.
- [3] 森山甲一, 沼尾正行: “環境状況に応じて自己の報酬を操作する学習エージェントの構築”, 人工知能学会論文誌, vol.17, no.6, pp.676-683 2002.
- [4] S. Mikami, Y. Kakazu and T.C. Fogarty: “Co-operative Reinforcement Learning By Payoff Filters, in Proc,” 8th European Conference on Machine Learning ECML-95(Lecture Notes in Artificial Intelligence 912), pp.319-322, 1995.
- [5] 浅田稔, 野田彰一, 俵積田健, 細田耕: “視覚に基づく強化学習によるロボットの行動獲得”, 日本ロボット学会誌, vol.13, no.1, pp.68-74, 1995.
- [6] 前田陽一郎, 花香敏: “Shaping 強化学習を用いた自律エージェントの行動獲得支援手法”, 知能と情報, 日本知能情報ファジィ学会誌, vol.21, no.5, pp.722-733, 2009.
- [7] 濱上知樹, 小坪成一, 平田廣則: “適応的な状態分割を行う Q-Learning における状態数の調整方法”, 電子情報通信学会論文誌, vol.J86-D-1, no.7, pp.490-499, 2003.
- [8] 鮫島和行, 片桐憲一, 銅谷賢治, 川人光男: “複数の予測モデルを用いた強化学習による非線形制御”, 電子情報通信学会論文誌, vol.J84-D-2, pp.2092-2106, 2001.
- [9] M. Nagayosi, H. Murao and H. Tamaki: “Developing reinforcement learning for adaptive co-construction of continuous high-dimensional state and action spaces,” Journal of Artificial Life and Robotics, pp.204-210, 2012.
- [10] 高橋泰岳, 浅田稔: “階層型学習機構における状態行動空間の構成”, 日本ロボット学会誌, vol.21, no.2, pp.164-171, 2003.
- [11] 内部英治, 銅谷賢治: “複数報酬のもとでの階層強化学習”, 日本ロボット学会誌, vol.22, no.1, pp.120-129, 2004.
- [12] J.L. Barry, L.P. Kaelbling and T. Lozano-Perez: “DetH*: Approximate Hierarchical Solution of Large Markov Decision Processes,” 22nd International Joint Conference on Artificial Intelligence, IJCAI-11, AAAI Press, pp.1928-1935, 2011.
- [13] 森本淳, 銅谷賢治: “強化学習を用いた高次元連続状態空間における系列運動学習—起き上がり運動の獲得—”, 電子情報通信学会論文誌, vol.J82-D-2, no.11, pp.2118-2131, 1999.
- [14] 小林祐一, 湯浅秀男, 新井民夫: “自律系分散系の適応アルゴリズムによる強化学習のための関数近似”, 計測自動制御学会論文集, vol.38, no.2, pp.219-226, 2002.
- [15] 小林高彰, 鎌谷長史, 森田昌彦: “選択的不感化ニューラルネットを用いた連続状態行動空間における Q 学習”, 電子情報通信学会論文誌, vol.J98-D, no.2, pp.287-299, 2015.
- [16] 山口明彦, 杉本徳和, 川人光男: “回避行動の再利用メカニズムを備えた強化学習手法と多関節ロボットの全身運動学習への応用”, 日本ロボット学会誌, vol.27, no.2, pp.209-220, 2009.
- [17] 港隆史, 浅田稔: “環境の変化に適応する移動ロボットの行動獲得”, 日本ロボット学会誌, vol.18, no.5, pp.706-712, 2000.
- [18] 小堀訓成, 鈴木健嗣, パトヨハルトノ, 橋本周司: “尤度情報に基づく温度分布を用いた強化学習法”, 人工知能学会論文誌, vol.20, no.4, pp.297-305, 2005.
- [19] 松井藤五郎, 犬塚博久, 世木博久, 伊藤英則: “強化学習結果の再構築への概念学習の適用”, 人工知能学会論文誌, vol.17, no.2, pp.135-144, 2002.
- [20] 阪口豊, 高野光雄: “環境変化への適応と文脈切替え”, 第 16 回生体・生理工学シンポジウム論文集, pp.157-160, 2001.
- [21] 片山謙吾, 奥石尚宏, 成久洋之: “強化学習エージェントへの階層化意志決定法の導入—追跡問題を例に—”, 人工知能学会論文誌, vol.19, no.4, pp.279-291, 2004.

- [22] 増山岳人, 山下淳, 浅間一: “変換不変性を用いた経験の抽象化と内発的動機づけに基づく強化学習”, 日本機械学会論文集 C 編, vol.79, no.798, pp.289–303, 2013.
- [23] 齋藤雅矩, 瀬古沢照治: “エージェントの行動履歴を活用した Q-learning アルゴリズムの提案”, 電気学会論文誌 C, vol.136, no.8, pp.1209–1217, 2016.
- [24] K. Terashima, H. Takano and J. Murata: “Acceleration of Reinforcement Learning by Controlled Use of Options Given as Prior Information,” SICE Journal of Control, Measurement, and System Integration, vol.6, no.4, pp.252–258, 2013.
- [25] 内部英治, 銅谷賢治: “重点サンプリングを用いた複数強化学習器の同時学習”, 電子情報通信学会技術研究報告, NC, pp.179–184, 2003.
- [26] 高橋泰岳, 田村佳宏, 浅田稔: “価値システムに基づく他者行為観察と自己行動学習の循環的発達”, 日本知能情報ファジィ学会誌, vol.21, no.5, pp.640–652, 2009.
- [27] 伊藤昭, 金満満: “知覚情報の粗視化によるマルチエージェント強化学習の高速化—ハンターゲームを例に—”, 電子情報通信学会論文誌, vol.J84-D-1, no.3, pp.285–293, 2001.
- [28] 日置智恵子, 松井博和, 野村由司彦: “未知環境において行動指針を利用する強化学習法—同一空間上に対して複数の学習テーブルを用いる方法—”, 第 44 回 人工知能基礎論研究会, 2001.
- [29] C.J.C.H. Watkins and P. Dayan: “Technical Note: Q-Learning,” Machine Learning, vol.8, pp.279–292, 1992.

松井博和 (Hirokazu Matsui)

1992 年名古屋工業大学工学部電気情報工学科卒業。1994 年名古屋大学大学院工学研究科博士前期課程情報工学修了。1997 年神鋼電機株式会社入社。1998 年三重大学工学部助手。2000 年名古屋大学 (工学博士), コミュニケーション, 自律移動や学習支援に関する知能ロボットの研究に従事。現在三重大学大学院工学研究科助教。電子情報通信学会会員。

(日本ロボット学会正会員)

西澤 (旧姓: 日置) 智恵子 (Chieko Nishizawa)

1999 年三重大学工学部機械工学科卒業。2001 年同大学大学院工学研究科修了。同年株式会社トヨタコミュニケーションシステム入社。2006 年会社再編成により, トヨタテクニカルディベロップメント株式会社に移動。工学博士 (三重大学)。

(日本ロボット学会正会員)

野村由司彦 (Yoshihiko Nomura)

1976 年名古屋大学工学部機械工学科卒業。1978 年名古屋大学大学院工学研究科修了。同年 NTT 電機通信研究所入所。1990 年名古屋大学工学部助教授。1997 年三重大学工学部教授。コンピュータビジョン, 知能ロボットの研究に従事。工学博士 (東工大)。日本機械学会, IEEE 等各会員。土木学会論文奨励賞。電子通信学会学術奨励賞受賞。

(日本ロボット学会正会員)