F-040

文脈に応じて行動価値を分割する Q-learning

Q-learning of distributing action-values according to the context

武石　真登 †　　濱上 知樹 †
Masato Takeishi　Tomoki HAMAGAMI

1　はじめに

強化学習は，学習の主体であるエージェントが環境との相互作用を通じて自律的に行動を獲得する手法である [1]．人が立ち入ることが困難な環境における行動獲得や，複雑なパラメータ設定が必要なロボットの自律制御等への応用が期待されている．

強化学習の代表的なアルゴリズムとして Q-learning[2] が知られている．Q-learning は現在の状態と行動によってのみ次の状態の遷移が決定する単純マルコフモデルを想定したアルゴリズムである．しかし，実環境においては，センサの性能等や物理的な誤差により環境が十分に観測できず，環境の状態遷移が単純マルコフモデルとみなせない場合が多い．その結果，Q-learning には，不完全知覚問題 [3] 等の学習問題が生じる．

この問題を回避する方法として，状態や行動の系列である「文脈」を利用して学習を行う方法が広く研究されている [4]〜[7]．本稿においても，多重マルコフモデルを扱うことのできる学習方法として，文脈を考慮した手法について提案する．

多重マルコフモデルを単純マルコフ過程として扱う場合，過去の $n-1$ 状態と現在の状態を組み合わせ，n 重状態として新たな単純マルコフ過程を定義するのが簡単である．しかし，一般に考慮すべき n は明らかでなく，状態数の爆発が起こり，学習が収束しない問題が生じる．

一方，ロボットなどでは，観測されうる状態空間よりロボットがとりうる行動空間のほうが次元が少ない場合が多い．また，未知の環境の状態数は明らかではないが，行動数はエージェントの属性として既知であることが多い．よって，学習に用いる n 重マルコフ過程を状態の列で定義するのではなく，エージェントの行動の系列を用いて定義することで，行動価値関数の獲得が容易になると想定される．

そこで，本稿では，行動価値を過去の行動の履歴に応じて分割し，行動選択を行うときに，分割された行動価値を参照する方法を提案する．この方法により，過去の行動の履歴により，同じ状態でも行動価値が異なって見えるため，文脈に適した行動選択が可能となる．また，行動価値同士の関連をネットワークとみなし，重みづけをしながら価値を分配することにより，行動の連続性を考慮した価値分配を行う．このように，行動価値同士のネットワークを用いて文脈の繋がりやすさを考慮することで，学習速度の向上と学習すべき行動価値数の増加を抑制する．

2　文脈に応じて行動価値を分割する Q-learning

Q-learning のアルゴリズムの中に，文脈に応じて行動価値を分割し，状態遷移の際に，分割された価値を単層のネットワークを介して分配・伝搬する機能を付加する．この手法を，本稿では，netQ-learning(netQL) と呼ぶ．

2.1　状態価値の分割

netQL では，n ステップ前の行動を参照して行動選択を行う．そのために，行動価値を参照する n ステップ前までの行動の組の個数 $|\mathcal{A}|^n$ だけ行動価値を分割して記憶する．ここで，$|\mathcal{A}|$ は行動空間の大きさである．一般に状態空間の大きさ $|\mathcal{S}|$ に比べ $|\mathcal{A}|$ は既知であることが多く，その大きさも小さい．そのため，学習すべき状態数を爆発させることなく行動価値を学習することが期待できる．

2.2　文脈に沿った価値更新を行うアルゴリズム

netQL では，価値更新時に，時刻 t における行動価値と，n ステップ前の状態における $|\mathcal{A}|^n$ 個の行動価値の間にネットワークを構築し，ネットワークで連結された価値に対して文脈を評価する関数を用いて価値を分配する．この文脈を評価する関数を，「価値伝搬制御関数」と呼ぶ．価値伝搬制御関数は，価値伝搬の際に分割された行動価値に対し，文脈に沿った行動選択が行われる場合に価値をより多く分配する関数である．この価値伝搬方法を用いることで，文脈に沿った行動遷移をする行動選択が行われやすくなる．つまり，行動の連続性を考慮して，適切でない行動が選択されないようにすることによって，学習速度の向上が期待できる．

2.3　価値更新のアルゴリズム

netQL の価値更新のアルゴリズムは，Q-learning に前述の価値伝搬制御を行うアルゴリズムを付加したものである．今，時刻 t における環境の状態を s_t，選択した行動を a_t，報酬を r_t，行動空間の大きさを $|\mathcal{A}|$ とする．また，時刻 t から，n ステップ前までの行動の系列を，$\boldsymbol{a}_t = \{a_t, a_{t-1}, \cdots, a_{t-n}\}$ とする．さらに，可能な行動系列を列挙したときの，k 番目の行動系列を $\boldsymbol{a}_k$ とする．ただし，$1 \leq k \leq |\mathcal{A}|^n$ である．netQL では，ある状態 s_t で行動 a_t を行い，状態 s_{t+1} 移った時，s_t における分割された $|\mathcal{A}|^n$ 個の行動価値に対して，価値伝搬制御関数 $c(t, \boldsymbol{a}_k)$ を用いて価値更新を行う．

$$\begin{aligned} netQ(s_t, a_t, \boldsymbol{a}_k) &\leftarrow (1-\alpha) netQ(s_t, a_t, \boldsymbol{a}_k) \\ &+ \alpha c(t, \boldsymbol{a}_k) \left\{ r_{t+1} + \gamma \max_{a'} netQ(s_{t+1}, a', \boldsymbol{a}_t) \right\} \\ &(k = 1, 2, \cdots, |\mathcal{A}|^n) \end{aligned} \tag{1}$$

ただし，α は学習率，γ は割引率である．このように価値伝搬制御関数 $c(t, \boldsymbol{a}_k)$ は，時刻 t における見かけ上の行動と，分割された行動価値のうち，k 番目の行動価値の文脈の繋がりを評価するために，$|\mathcal{A}|^n$ 個の netQ 値に対して価値 $r_{t+1} + \gamma \max_{a'} netQ(s_{t+1}, a', \boldsymbol{a}_t)$ を文脈に応じて比例分配をしている．図 1 に netQL の価値更新の様子を示す．

価値伝搬制御関数により，次に示す効果が期待される．

- 見かけ上の行動と，k 番目に定義される行動価値の文脈上の繋がりの評価
- 文脈上無理のある行動を生起する行動に対する価値伝搬の抑制

これらの効果により，文脈の制約を効率よく利用した価値伝搬が行われ，学習速度の向上が期待される．

2.4　価値伝搬制御を行うことによる netQ 値の意味

Q-learning の価値更新のアルゴリズムによって得られる Q 値は，将来得られる期待報酬の和である．よって，学習終了時に各状態において最も Q 値の高い行動を選択することで，エージェントは期待報酬を最も多く得る行動を選択することができる．対して，netQL では式 (2) により価値伝搬制御を行

† 横浜国立大学大学院工学府

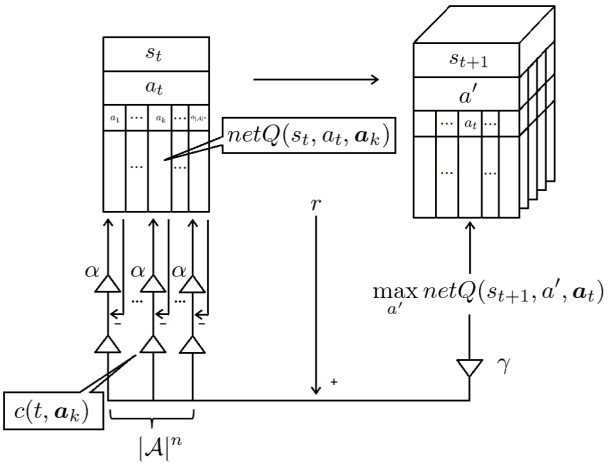

図 1　netQ

うため，ある方策 π における netQ 値は次式で定義される．

$$\begin{aligned} netQ^{\pi}(s_t, a_t, \boldsymbol{a}_k) &= E^{\pi}[c(t)R(s_t, a_t, s_{t+1}) \\ &\quad + c(t)c(t+1)\gamma R(s_{t+1}, a_{t+1}, s_{t+2}) + \cdots] \\ &= E^{\pi}\left[\sum_{k=t}^{\infty}\left\{\left(\prod_{l=t}^{k} c(l)\right)\gamma^{k-t}R(s_k, a_k, s_{k+1})\right\}\right] \end{aligned} \quad (2)$$

ただし, $c(t)$ は時刻 t において行う価値伝搬制御, $R(s_t, a_t, s_{t+1})$ は時刻 t において状態 s_t で行動 a_t を行い状態 s_{t+1} に移ったときに得られる報酬である．式 (2) より，$netQ(s_t, a_t, \boldsymbol{a}_k)$ は今後行われる価値伝搬制御を考慮した報酬の和の期待値と言える．

2.5　行動選択のアルゴリズム

netQL では，同じ状態において，n ステップ前の行動の履歴に応じて行動価値を複数持つ．よって，同じ状態でも過去にどのような行動を行ったかによって，エージェントにとっての行動価値の見え方が変化する．

以下に n ステップ前までの行動履歴の行動列が，$\boldsymbol{a}_k$ であったときの，ϵ-greedy 方策を用いたときの行動選択方法を示す．

$$\pi(s, a, \boldsymbol{a}_k) = \begin{cases} 1-\epsilon & (a = \arg\max_a netQ(s_t, a, \boldsymbol{a}_k)) \\ \epsilon & (\text{othewise}) \end{cases} \quad (3)$$

このように，n ステップ前の行動に応じて参照する行動価値を変化させることによって，文脈を考慮した行動選択が行われる．

2.6　netQ の価値更新のアルゴリズムの導出

2.3 節で netQ 値の価値更新のアルゴリズムについて示したが，本節では netQ 値の更新式の導出を行う．(2) 式より，ある方策 π における netQ 値は次式で定義される．

$$\begin{aligned} netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) &= E^{\pi}\Big[c(t)R(s_t, a_t, s_{t+1}) \\ &\quad + c(t)c(t+1)\gamma R(s_{t+1}, a_{t+1}, s_{t+2}) + \cdots\Big] \\ &= E^{\pi}\left[\sum_{k=t}^{\infty}\left\{\left(\prod_{l=t}^{k} c(l)\right)\gamma^{k-t}R(s_k, a_k, s_{k+1})\right\}\right] \end{aligned} \quad (4)$$

さらに式 (4) を式展開して，Bellman 方程式の形に変形する．

$$\begin{aligned} netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) &= E^{\pi}\left[\sum_{k=t}^{\infty}\left\{\left(\prod_{l=t}^{k} c(l)\right)\gamma^{k-t}R(s_k, a_k, s_{k+1})\right\}\right] \\ &= E^{\pi}\left[c(t)R(s_t, a_t, a_{t-1})\right] \\ &\quad + c(t)\gamma netQ(s_{t+1}, a_{t+1}, \boldsymbol{a}_t) \end{aligned} \quad (5)$$

ここで，TD 法と同じアプローチを用いて $netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1})$ を求めることを考える．まず式 (5) を変形して，期待報酬関数 $E^{\pi}[c(t)R(s_t, a_t, s_{t+1})]$ に関する式に変形すると，次式が得られる．

$$\begin{aligned} E^{\pi}[c(t)R(s_t, a_t, s_{t+1})] &= netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad - E^{\pi}[\gamma c(t) netQ^{\pi}(s_{t+1}, a_{t+1}, \boldsymbol{a}_t)] \end{aligned} \quad (6)$$

さらに，価値伝搬制御を含めた期待報酬 $E^{\pi}[c(t)R(s_t, a_t, s_{t+1})]$ と，価値伝搬制御を含めた報酬 $c(t)r_t$ 間の二乗誤差 $\Delta\{c(t)R(s_t, a_t, s_{t+1})\}$ を求めると，式 (7) が得られる．

$$\begin{aligned} &\Delta\{c(t)R(s_t, a_t, s_{t+1})\} \\ &= \frac{1}{2}\Big\{c(t)r_{t+1} - E^{\pi}[c(t)R(s_t, a_t, s_{t+1})]\Big\}^2 \\ &= \frac{1}{2}\Big\{c(t)r_{t+1} - netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad + E^{\pi}[\gamma c(t) netQ^{\pi}(s_{t+1}, a_{t+1}, \boldsymbol{a}_t)]\Big\}^2 \end{aligned} \quad (7)$$

よって，価値伝搬制御を含めた強化学習の問題は，$netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1})$ を更新して，誤差 $\Delta\{c(t)R(s_t, a_t, s_{t+1})\}$ を 0 に近づける問題になる．最急降下法を用いて $netQ^{\pi}(s_t, a_t \boldsymbol{a}_{t-1})$ の更新式を求めると，次式の更新式が得られる．

$$\begin{aligned} netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) &\leftarrow netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad - \alpha' \frac{\partial}{\partial netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1})}\Delta\{c(t)R(s_t, a_t, s_{t+1})\} \\ &= netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad + \alpha\Big\{c(t)r_{t+1} - netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad + E^{\pi}[\gamma c(t) netQ^{\pi}(s_{t+1}, a_{t+1}, \boldsymbol{a}_t)]\Big\} \end{aligned} \quad (8)$$

netQL では,Q-learning と同様に $E^{\pi}[netQ^{\pi}(s_{t+1}, a_{t+1}, \boldsymbol{a}_t)] = \max_{a'} netQ^{\pi}(s_{t+1}, a', \boldsymbol{a}_t)$ とおいて式 (9) で価値更新を行う．

$$\begin{aligned} netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) &\leftarrow netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1}) \\ &\quad + \alpha\Big[c(t)\Big\{r_t + \gamma \max_{a'} netQ^{\pi}(s_{t+1}, a', \boldsymbol{a}_t)\Big\} \\ &\quad - netQ^{\pi}(s_t, a_t, \boldsymbol{a}_{t-1})\Big] \end{aligned} \quad (9)$$

以上より，式 (9) を用いることで価値伝搬制御を扱った netQL の更新を行うことができる．

3　実験および考察

シミュレーションの実験により，提案手法の性能を確認する．図 2 に示すシミュレーション環境において，netQL を用いて自律移動エージェントがスタートエリアからゴールエリアへ自律移動するタスクを行った．エージェントの行動の定義は図 3 で定義される．また，環境側の制約として，1 ステップ前の速度 $\boldsymbol{v}$ は $h\boldsymbol{v}$ に減衰する．ここで h は減衰定数であり $0 < h < 1$ である．このシミュレーション環境において，エージェントは速度を保ちながら，急角度での方向転換を避けるよう自律行動を学習する．すなわち，エージェントは観測状態のみならず一連の行動の系列を踏まえた次行動を選択することが要求される．本稿では，このような行動を「滑らかな行動」と呼ぶ．

実験 1 では，本タスクを Q-learning と netQL についてそれぞれが獲得した経路を調べ，滑らかな行動の獲得結果を比較した．また，netQL の収束性を調べるために，各エピソード毎

のステップ数についての収束性を Q-learniing，netQL で評価した．

実験 2 では，学習結果が，異なる文脈の獲得にも有効に働くかどうかを調べた．具体的には，単純マルコフモデルでの学習を行い，段階的に速度依存性のある環境に変化させた場合に，効率的に学習が可能であることを調べた．また，収束性についても評価を行った．

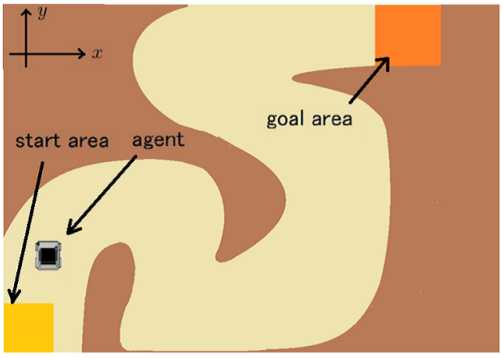

図 2　シミュレーション環境

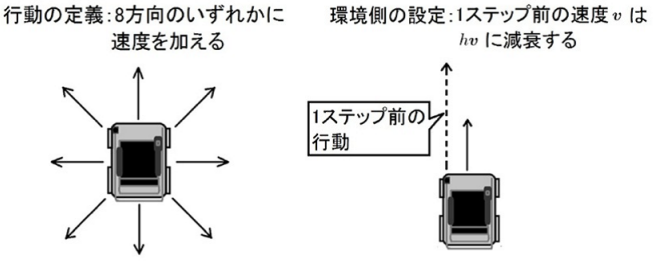

図 3　行動の定義

3.1　価値伝搬制御関数の設定

本実験では，netQL の基本的な性能を確認するために $n = 1$ とし，直前の行動と結びついた文脈に応じて行動価値が分割される設定とした．

時刻 t における自律移動エージェントの進行方向のベクトルを $\boldsymbol{v}_t$，1 ステップ前において k 番目に定義された行動 a_k を行うことにより加えられる速度ベクトルを $\boldsymbol{v}_k$ とする．今回は，$\boldsymbol{v}_t$ と $\boldsymbol{v}_k$ を用いて価値伝搬制御関数を次式で設定する．

$$c(t, a_k) = \frac{f(\boldsymbol{v}_t, \boldsymbol{v}_k)}{\sum_{l=1}^{|\mathcal{A}|} f(\boldsymbol{v}_t, \boldsymbol{v}_l)} \tag{10}$$

ただし，

$$f(\boldsymbol{v}_t, \boldsymbol{v}_k) = \begin{cases} \frac{\boldsymbol{v}_t}{\|\boldsymbol{v}_t\|} \cdot \frac{\boldsymbol{v}_k}{\|\boldsymbol{v}_k\|} & \left(\frac{\boldsymbol{v}_t}{\|\boldsymbol{v}_t\|} \cdot \frac{\boldsymbol{v}_k}{\|\boldsymbol{v}_k\|} > 0\right) \\ 0 & (\text{othewise}) \end{cases} \tag{11}$$

このような $c(t, a_k)$ を設定することにより，急角度の方向転換をしない行動を選ぶ netQ 値ほど価値が伝搬しやすくなる．

3.2　実験 1: 提案手法の有効性についての検証

3.2.1　実験方法

図 2 の環境において，状態 s_t は座標 (x, y) として，Q 値，netQ 値を RBF ネットワーク [9] を用いて関数近似することにより求めた．スタートエリアからゴールエリアまでエージェントが自律移動するタスクについて，Q-learning，　netQL の 2 つのアルゴリズムを用いて学習を行い，その結果の比較を行った．報酬はエージェントがゴールについたときのみ与え，$r = 100$ とした．表 1 に実験パラメータを示す．行動選択は ϵ-greedy 方策を用いて，表 2 に示すスケジューリングで学習を行った．

表 1　実験 1 パラメータ

パラメータ	設定値
代表点の個数 N_μ	280
netQ 値の初期値	0
重みの学習率 α_ω	0.3
基底関数の分散 α_σ	0.3
代表点の学習率 α_μ	0.3
割引率 γ	0.9
分散の初期値 σ_s	20
報酬 r	100
行動の数 $\|\mathcal{A}\|$	8
減衰定数 h	1/2
参照する行動列の大きさ n	1

表 2　実験 1 スケジューリング

episodes e	ランダム行動選択確率 ϵ
$0 \le e < 10$	0.9
$10 \le e < 60$	0.8
$60 \le e < 100$	0.7
$100 \le e < 120$	0.6
$120 \le e < 150$	0.4
$150 \le e < 170$	0.2
$170 \le e$	0

3.2.2　実験結果と考察

図 4 に，学習により得られた経路を示す．図 5 に，学習後における自律移動エージェントの単位時間あたりの進行方向のベクトル間のなす角の分布を示す．これらの結果より，提案手法により，文脈が考慮された「滑らかな行動」が獲得できていることが明らかとなった．

さらに，急な行動転換をせずに，速度を出す行動が行われたかを評価するための定性的な評価指標として次の評価値を導入する．

$$\text{value} = \frac{1}{N_s - 1} \sum_{t=1}^{N_s} \boldsymbol{v}_t \cdot \boldsymbol{v}_{t+1} \tag{12}$$

ただし，v_t はある時刻 t における速度，N_s はスタートからゴールまでのステップ数である．この評価値 value は，速度を上げつつ，各ステップごとの回転角度が小さい場合に，大きな値をとる．この評価値を用いて学習を行った結果を表 3 に示す．

この結果から，提案手法は，滑らかな行動が獲得できることが明らかとなった．Q-learning を用いた学習では，各状態ごとに期待報酬が最大になるように行動を学習するが，状態遷移を考慮した行動選択が行われないため，急角度で曲がる行動や壁に衝突する行動が獲得される．対して提案手法では，将来行われる価値伝搬制御を伴う行動価値である netQ 値を用いて行動選択が行われるため，滑らかな行動が獲得されている．

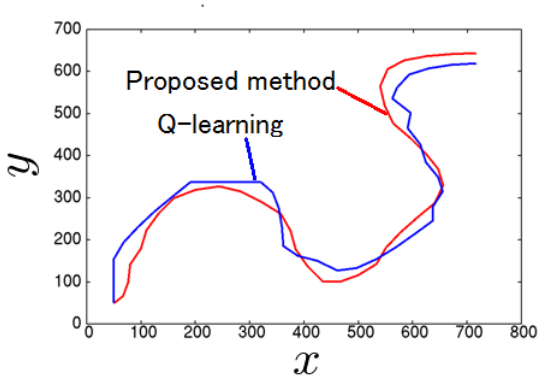

図 4　提案手法と Q-learning の経路の比較

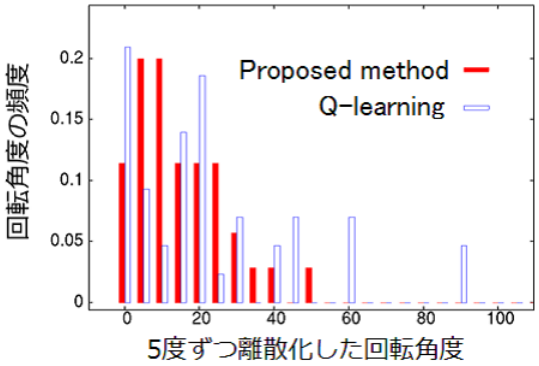

図 5　回転角度の頻度分布

表 3　実験 1 の評価値

method	value
Q-learning	1128
proposed method	1659

netQL の収束性を調べるために，各エピソードに対するゴールまでのステップ数の変化をプロットしたものを，図 6 に示す．図 6 では，(1) 提案手法を用いて学習を行った場合，(2)Q-learning を用いて学習を行った場合，(3)1 ステップ前の行動の履歴と現在の (x,y) 座標の組み合わせから状態を定義し，Q-learning を行った場合の，3 つの学習の収束過程を示している．ここで，各行動価値の空間の大きさは，(1)$|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{A}|$, (2)$|\mathcal{S}| \times |\mathcal{A}|$, (3)$|\mathcal{S}| \times |\mathcal{A}| \times |\mathcal{A}| \times |\mathcal{A}|$ となる．

以上の結果から，(1)netQL は，(2) 状態行動空間の小さい Q-learning と同等の収束性があることが明らかとなった．一方で，(3) 文脈獲得を行うために状態数を増やした Q-learning では，学習が収束しなかった．netQL を用いた場合では，価値関数の数が多いのにもかかわらず，Q-learning と同程度の収束性が得られた．これは，netQL では価値更新時に文脈が類似した場合には価値分配を行っていることで，般化の効果が得られたためと予想された．

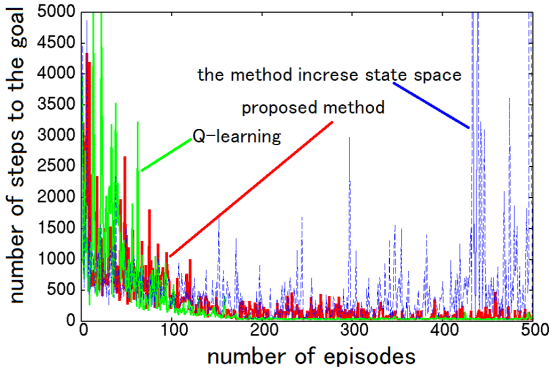

図 6　実験 1 のエピソード毎のステップ数

3.3　実験 2: 価値分配の有効性の検証

上位で述べた価値分配に伴う般化の効果を検証するための実験を行った．

まず，速度干渉のない文脈を考慮する必要のない環境から学習を始める．すなわち，単純マルコフモデルで学習できる内容で学習を行う．次に，段階的に速度干渉が起こる環境に移行し，途中までの学習結果から遅延なく文脈を考慮した学習が行われるかどうかを評価する．

3.3.1　実験方法

学習の初期段階においては，1 ステップごとに加えた速度を 0 にすることにより，1 ステップ前の速度が次の状態に干渉しないようにする．そのときに，行動選択には，次の式を用いた．

$$\pi(s,a) = \begin{cases} 1-\epsilon & (a = \arg\max_a Q'(s_t,a)) \\ \epsilon & (\text{othewise}) \end{cases} \quad (13)$$

$$Q'(s_t,a) = \sum_{k=1}^{|\mathcal{A}|} netQ(s_t,a,a_k) \quad (14)$$

netQ 値は，数ステップ前の行動の履歴に応じて分割された行動価値である．よって，式 14 を用いて分割された価値の和で行動選択を行うと，単純マルコフモデルを想定した行動選択が可能である．学習が進行すると，速度干渉を伴う多重マルコフモデルの環境に変化し，最終的には実験 1 と同様の環境となる．初期段階，学習が進んだ段階いずれにしても，価値更新のアルゴリズムは前述の netQL の方法を用いている．実験パラメータ，方策のスケジューリングは実験 1 と同様である．また，段階的に行動を変化させるスケジューリングを表 4 に示す．

表 4　実験 2 スケジューリング

episodes e	目的行動の生起確率 p
$0 \le e < 10$	0.1
$10 \le e < 60$	0.2
$60 \le e < 100$	0.3
$100 \le e < 120$	0.4
$120 \le e < 150$	0.6
$150 \le e < 170$	0.8
$170 \le e$	1.0

3.4 実験結果と考察

図 7 に実験 1, 実験 2 の提案手法で得られた経路の比較，図 8 に単位時間あたりの進行方向のベクトルのなす角の分布を示す．また，評価には実験 1 で用いた「滑らかな行動」の評価値式 (12) を用いる．実験の結果を表 5 に示す．

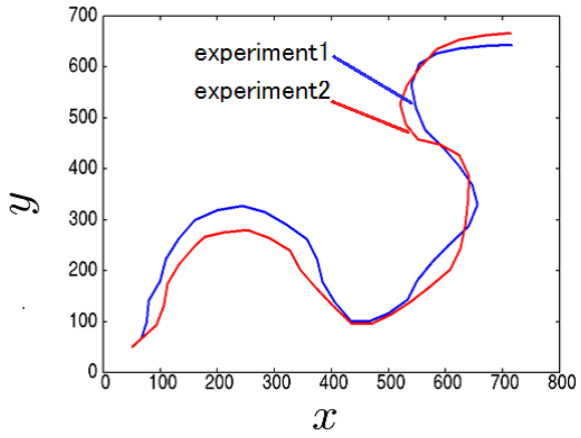

図 7　実験 2 と実験 1 の経路の比較

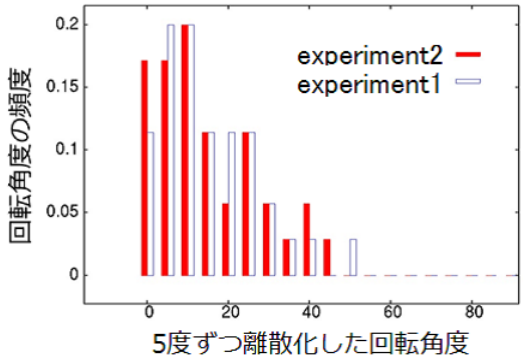

図 8　実験 2 の回転角度の頻度分布

表 5　実験 1 の評価値

method	value
Q-learning	1128
proposed method(実験 1)	1659
proposed method(実験 2)	1578

獲得された経路は，実験 1 と実験 2 で一見「滑らかさ」は変わらないが，評価値では実験 2 の評価値がより低くなっていることが分かる．これは，学習初期の段階での単純マルコフモデルを想定した学習により，学習結果が局所解に陥ってしまったためと考えられる．一方，Q-learning に比べると実験 2 の評価値が上回っていることから，価値伝搬制御関数により類似の文脈に対しても価値伝搬がなされ，学習の般化が行われていることが分かった．

学習の収束の様子を図 9 に示す．図 9 より，単純マルコフモデルにおける初期の学習段階では，学習の収束が早いことがわかる．また，Q-learning と比較して高い評価値が得られるのは，単純マルコフモデルを想定した行動であっても，複数の文脈における価値伝搬が行われ，その後の学習が容易になっているためだと考えられる．このように，経験が浅いときには慎重に行動をし，それを生かして段階的に複雑な環境での行動を獲得していく過程は，生物における探索に類似している．

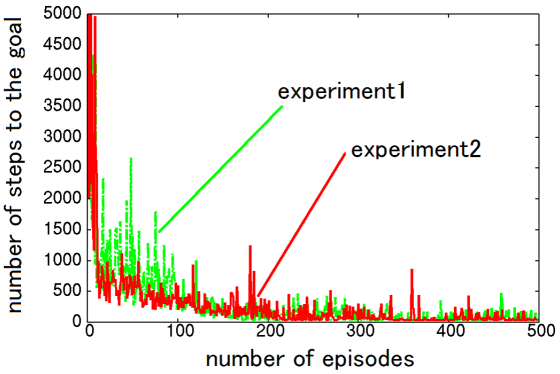

図 9　実験 2 のエピソード毎のステップ数

4　おわりに

本稿では，文脈に応じて行動価値を分割する Q-learning を提案した．

提案手法の有効性を確認するために，各状態の行動に，1 ステップ前の速度が干渉する環境における自律移動タスクを netQL を用いて学習させた．実験 1 では，Q-learning と netQL との学習結果を比較することにより行動の繋がりを考慮した学習が行われることが確認された．また，学習の収束性を (1)Q-learning, (2)netQ-learning, (3)Q-learning において文脈を考慮するために状態数を増加させた場合 の 3 つの場合において比較した．その結果，netQL は行動価値の数が多いのにも関わらず，学習の収束性能が Q-learning と同等であることが明らかになった。実験 2 では，単純マルコフモデルの行動を行いつつ，内部では文脈を考慮した価値伝搬を行うことで，類似の文脈に対しても価値伝搬がなされ，学習の般化が行われていることが分かった．その結果，実験 2 と比較して評価値は下がるものの，収束は早くなり，獲得された行動も，Q-learning と比較して評価値の高い行動が学習されることが明らかとなった．

今後はより効率の良く学習が進み，学習速度向上に効果的な価値伝搬制御関数の設定方法について検討し，強化学習の学習速度の問題の解決を行えるようにするアルゴリズムを検討する予定である．

参考文献

[1] Richard S. Sutton and Andrew G. Barto(三上貞芳, 皆川雅章 共訳): 強化学習, 森北出版株式会社, 2003.

[2] C.J.C.H. Warkins and P.Dayan: Technical note:Q-learning, Machine Learning, Vol.8, pp.279–292, 1992.

[3] S.D. Whitehead and D.H. Ballard: Learning to perceive and act by trial and error, Machine Learning, vol7, no.1, pp.45–83, 1991.

[4] 澁谷長史, 濱上知樹: 複素数で表現された行動価値を用いる Q-learning, 電子情報通信学会論文誌, vol.J91-D, no.5, pp.1286-1295, 2008.

[5] T.Shibuya, T.Hamagami: Multiplied Action Values for Complex-valued Reinforcement Learning, Proc. of the In-

ternational Conference on Electrical Engineering, I9FP0491, pp.5-8, 2009.
[6] 斉藤宋孝: 不完全知覚に対する内部メモリを用いた強化学習法に関する研究，情報処理学会，研究報告，pp.81–87, 2004.
[7] Lanzi PL :Adaptive Agents with Reinforcement Learning and Internal Memory, Proceedings of the Sixth Internationak Conference on the Simulation of Adaptive behavior, Paris, 2000.
[8] A.McCallum: Instance-baced util distinctions for reinforcement learning with hidden state, International Conference on Machine learning, pp.387–395, 1995.
[9] 渕田孝康, 前原正和, 森邦彦, 村島定行: 強化学習的手法を用いた RBF ネットワークの学習, 信学技報 TECHNICAL REPORT OF IEICE. NC99–113, pp.157–163, 2000.