強化学習における Support Vector Machine を用いた状態一般化法

後藤　　亮†　　松尾　啓志†

State Generalization Method with Support Vector Machines in Reinforcement Learning

Ryo GOTO† and Hiroshi MATSUO†

あらまし　従来の強化学習は離散的な状態空間を対象としたものが多く，連続的な状態を扱うには状態を離散化する必要がある．しかし単純な離散化は，状態の次元の増加とともに状態数を指数的に増加させてしまうため，学習に要する時間や必要となるメモリ量が増大する．本論文では，Support Vector Machine（SVM）を用いて多次元連続な状態を一般化する手法を提案する．本手法は，SVM を用いることで未知の状態における最適な行動を推定し，より少ない試行で環境に適応することが期待できる．また，ロボットをゴールまで移動させるタスクを想定し，シミュレーション上で従来手法との比較実験を行った．その結果，本手法がより少ない試行で環境に適応することを確認した．

キーワード　強化学習，Support Vector Machine，状態一般化

1. まえがき

近年，ロボット技術の発展により，自律的に環境に適応する学習ロボットの重要性が高まりつつある．そうした自律的な制御規則獲得のための有効な手法として，強化学習が注目を集めている．強化学習とは，環境から得られる報酬信号を基に自動的に制御規則を獲得する学習の枠組みである．従来の強化学習研究は離散状態空間を対象として行われたものが主であった．離散状態空間を対象とした手法を連続状態空間に適用する場合には，状態空間を離散化する必要がある．しかし，単純な離散化では入力状態の次元が上がるにつれ状態数が指数的に増加し，学習に要する時間や必要となるメモリ量が著しく増加してしまう．実際の環境では状態入力が連続的な多次元ベクトルで与えられることは多々あると考えられる．そのため実問題への応用を考えたとき，状態の離散化に伴う問題を解決することが重要となる．

連続状態空間を対象とした手法として，価値関数を連続関数で近似するもの [1], [2] などが提案されている．これらの手法では，未経験の状態に対しても性質の似た状態での経験に基づいて行動できるなどの利点がある．しかしこれらの手法は報酬を少しずつ伝播させることで状態の価値を評価するため，価値の予測が困難な複雑な環境に適応するには多数の試行が必要となる．その他のアプローチとして，状態空間を適応的に分割し自律的に状態空間を構成する手法 [3]〜[6] などが提案されている．しかし，これらは状態空間を線形に分割するもの [3], [4] や，楕円体モデルを用いて構成するもの [5], [6] など単純なモデルによるものが多いため，その汎化能力は低いと考えられる．

本論文では，Support Vector Machine（SVM）[7] を用いて多次元連続状態を一般化し，少ない試行から環境に適応する手法を提案する．SVM は 2 クラスのパターン認識手法であり，非線形のデータ分離が可能なことや，汎化誤差が次元に依存しないことなどから近年注目されている．本手法では，SVM の特徴である柔軟な非線形データ分離を用いて，従来法に比べ汎化能力の高い状態一般化を行う．

以下，**2.** で強化学習について，**3.** で SVM について概説する．**4.** で本研究で提案する状態一般化手法について述べる．**5.** では自律移動ロボットのシミュレーションにより提案手法の評価を行い，考察を述べる．**6.** で本論の結論を述べる．

† 名古屋工業大学電気情報工学科，名古屋市
Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466–8555, Japan

2. 強化学習

強化学習は教師無し学習の一種であり，エージェントには目標とする行動出力は与えられない．その代わりに一連の行動の結果として報酬が与えられ，エージェントは試行錯誤を通じて目標出力を獲得する．

エージェントは制御対象である環境の現在の状態を観測した後，その状態入力をもとに行動を選択し，出力する．その結果，エージェントは環境から報酬を受け取る．これらのインタラクションの繰り返しにより，エージェントは環境に対する知識を獲得する．通常，エージェントは目標の状態に到達したときのみ報酬を獲得するため，行動を実行した直後の報酬を見るだけでは，その行動が正しかったかどうかを判断できないという困難を伴う．

強化学習における代表的な手法である TD 学習 [8] や，それを拡張した Q-learning [9] では，報酬を基に状態価値の評価を行う．状態価値とは，ある状態において将来にわたって獲得できる報酬の総和の期待値である．Q-learning などの離散状態空間を対象とした手法では，すべての状態の状態価値を記憶しておく必要がある．

価値関数の連続関数による近似により，連続状態空間を効果的に扱うことのできる手法も提案されている．森本らは Normalized Gaussian Network（NGnet）[10] を拡張した Incremental NGnet（INGnet）を用いて高次元連続状態空間での学習を行う手法を提案した [2]. NGnet は正規化ガウス関数を基底関数とし，それらを組み合わせることにより連続関数を表現するネットワークであり，INGnet は基底関数を適応的に追加できるようこれを拡張したものである．森本らの手法では，INGnet を Actor-Critic [8] に適用し連続状態空間における価値関数を学習する．

3. Support Vector Machine

Support Vector Machine（SVM）は 2 クラスのパターン認識手法であり，L 次元の特徴ベクトル $\mathbf{x}_n$（$\mathbf{x}_n \in \mathbf{R}^L$）と，それに割り当てられた正負のラベル y_n（$n = 1, 2, \ldots, N$）で表される N 個の観測データから，正負クラスを判定する識別関数 $f(\mathbf{x})$ を求める．

3.1 線形 SVM

SVM による観測データの線形分離を図 1 に示す．なお，すべての観測データは線形に分離可能なものとする．図中の $H_0 : \mathbf{w}\cdot\mathbf{x}+b=0$ は正のデータと負のデータを分離する分離超平面である．ただし $\mathbf{w}$ は超平面の法線ベクトル，b は定数である．$H_1 : \mathbf{w}\cdot\mathbf{x}+b=1$, $H_2 : \mathbf{w}\cdot\mathbf{x}+b=-1$ は H_0 に平行な超平面で，H_0 から最も近い正の特徴ベクトルは H_1 上にあり，最も近い負の特徴ベクトルは H_2 上にある．H_1 と H_2 の間の距離はマージンと呼ばれる．SVM はマージンを最大化する分離超平面 H_0 を求めるアルゴリズムである．

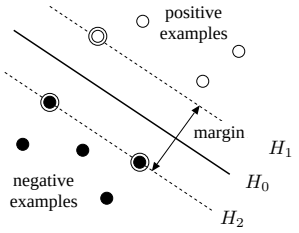


図 1　線形 SVM におけるデータの分離
Fig. 1　The separation of linear SVM

すべての観測データが誤りなく分離できる場合，H_1 と H_2 の間にはデータは存在しない．よってすべての n において次式が成り立つ．

$$y_n(\mathbf{w}\cdot\mathbf{x}_n+b)-1 \geqq 0 \tag{1}$$

マージンを最大化するには $1/\|\mathbf{w}\|$ を最大化する必要がある．この問題は式 (1) の制約が付いた $\|\mathbf{w}\|^2$ の最小化問題へと定式化できる．さらに正の乗数 $\alpha_1, \ldots, \alpha_N$ を定義し Lagrange の未定乗数法を用いることで，$\|\mathbf{w}\|^2$ の最小化問題は二次計画問題に帰着される．最終的に識別関数 $f(\mathbf{x})$ は

$$f(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x}+b = \sum_{n=1}^{N}\alpha_n y_n \mathbf{x}_n\cdot\mathbf{x}+b \tag{2}$$

となる．なお，ここでは観測データは線形分離可能なものと仮定したが，線形分離が不可能な場合でも識別関数は式 (2) と同じ形になる．

二次計画問題を解くかわりに，Sequential Minimal Optimization（SMO）[11] を用いて高速に $\alpha_1, \ldots, \alpha_N$ を求めることができる．本研究ではこの SMO を用いた．

3.2 非線形 SVM

非線形 SVM では，$\mathbf{R}^L$ から，高次元空間 $\mathcal{H}$ への写像 $\Phi : \mathbf{R}^L \mapsto \mathcal{H}$ を用いて，$\mathcal{H}$ 上で線形 SVM を適用することで非線形分離を行う．このときの識別関数を

$$f(\mathbf{x}) = \sum_{n=1}^{N}\alpha_n y_n K(\mathbf{x}_n, \mathbf{x}) + b \tag{3}$$

とする．ただし $K(\mathbf{x},\mathbf{y}) = \Phi(\mathbf{x})\cdot\Phi(\mathbf{y})$ である．写像 $\Phi(\mathbf{x})$ を求めるのは困難であるが，$\mathcal{H}$ 上での内積 $K(\mathbf{x},\mathbf{y})$ が直接計算できれば $\Phi(\mathbf{x})$ を知る必要はない．関数 $K(\mathbf{x},\mathbf{y})$ はカーネル関数と呼ばれ，いくつかの関数が知られている．本研究では次式で表されるガウシアンカーネルを用いた．

$$K(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{4}$$

ただし σ は適当な定数である．

4. SVM を用いた状態一般化法

連続状態空間では状態が無数に存在するため，状態の一般化を行う必要がある．しかし従来手法は設計者が状態空間を分割し状態を離散化するものや，楕円体モデルなどの単純なモデルに基づいて一般化するものが多い．そのため，複雑な環境では状態空間の構成が複雑になり，未知の状態に対する推定効果も弱くなる．

そこで本研究では，SVM を用いて状態の一般化を行い，未知の状態における最適な行動を推定する手法を提案する．本手法は SVM の高い汎化能力を活用することにより，従来の強化学習よりも少ない試行で環境に適応することができる．本論文で提案する一般化は次の 3 つの手順から成る．

（1） 行動経験を収集する．

（2） SVM を用いて状態の価値を推定する．

（3） 各行動について，行動後に価値の高い状態へと遷移する状態を，再び SVM を用いて一般化する．

以下，各手順の詳細と行動選択について述べる．

4.1 行動経験の収集

状態を一般化するためには，状態とそれらの状態の性質を知る必要がある．そこでランダム行動により，行動前の状態 $\mathbf{s}$，行動 a，行動後の状態 $\mathbf{s}'$ の 3 つの組で表される行動経験 $(\mathbf{s},a,\mathbf{s}')$ を収集する．（本論文では状態 $\mathbf{s}$, $\mathbf{s}'$ はすべてベクトルとする．）しかし収集されたすべての状態に対し SVM を適用すると，訓練データ数が膨大になり SVM の計算コストが増大する．そこで次のような制限を設ける．

- 各試行において，ゴール状態に到達したステップからさかのぼって T ステップ分の行動経験のみを収集する．これは，価値の推定がゴール状態に近い状態から行われていくことから，ゴール状態付近での行動経験がより重要となると考えられるためである．
- 新しく得られた行動経験とすでに収集されている行動経験の行動が同じで，かつ行動前の状態間の距離と行動後の状態間の距離が閾値 d_{th} より小さい場合，新しく得られた行動経験は収集しない．

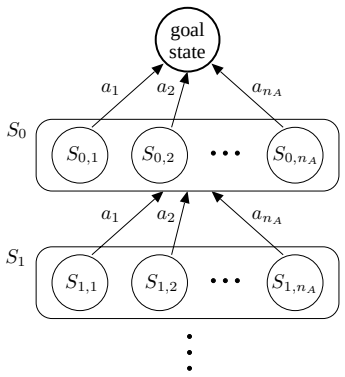


図 2 状態価値の評価
Fig. 2 Estimation of state value

4.2 状態価値の推定

本提案手法では，ある行動を実行した結果，現在の状態よりも価値の高い状態クラスに遷移するかどうかを推定する．そのため，収集された状態の価値を求める必要がある．図 2 に状態価値の評価方法を示す．報酬がゴール状態に到達したときのみに得られる環境において，状態価値はゴール状態までの最適ステップ数の指標と考えられる．そこで本手法はゴール状態に 1 行動で到達した状態を一般化して状態クラスを生成し，さらにそのクラスに 1 行動で到達した状態を一般化するという手順を繰り返すことで状態空間を構成し，状態価値を推定する．以下にアルゴリズムを示す．

（1） 行動集合を A，行動経験の集合を D で表す．$k = 1,\ldots,n_A$（$n_A = |A|$）とし，集合 D_k を集合 D に含まれる行動経験の中で行動が $a_k \in A$ であるものの集合とする．また変数 j を用意し，$j \leftarrow 0$ で初期化する．

（2） 集合 D_k 中の i 番目の行動経験を $\delta_k^i = (\mathbf{s}^i, a_k, \mathbf{s}'^i)$ で表す．なお，$\mathbf{s}^i$ は行動前の状態，a_k は行動，$\mathbf{s}'^i$ は行動後の状態である．δ_k^i において，$\mathbf{s}'^i$ がゴール状態であれば $\mathbf{s}^i$ を正事例，$\mathbf{s}'^i$ がゴール状態でなければ $\mathbf{s}^i$ を負事例とする．すべての $\delta_k^i \in D_k$ について $\mathbf{s}^i$ の正負判定を行い，ガウシアンカーネルを使用した SVM により正事例集合と負事例集合の分離を行う．ここで $\mathbf{s}^i$ が正事例である場合，δ_k^i を集合 D_k から取り除いておく．求めた識別関数を $f_{j,k}(\mathbf{s})$ とし，すべての k について $f_{j,k}(\mathbf{s})$ を求める．また，SVM の識別関数の計算を高速化するため，あまり重要でない

訓練データを削除する．すなわち式 (3) において α_n が閾値 α_{th}（α_{th} は 0 に近い正の数）以下である場合，その項は識別関数 $f(\mathbf{x})$ の値にはそれほど影響を与えないため，n 番目のデータ $(\mathbf{x}_n, y_n)$ を削除する．

（3） 識別関数 $f_{j,k}(\mathbf{s})$ の符号が正となる状態の集合を $S_{j,k} = \{\mathbf{s}|f_{j,k}(\mathbf{s}) > 0\}$ と表す．各 k で求められた $S_{j,k}$ の和集合 $S_j = \bigcup_{k=1}^{n_A} S_{j,k}$ を求める．

（4） 行動経験 $\delta_k^i = (\mathbf{s}^i, a_k, \mathbf{s}'^i) \in D_k$ において，$\mathbf{s}'^i \in S_j$ ならば $\mathbf{s}^i$ を正事例とし，$\mathbf{s}'^i \notin S_j$ ならば負事例とする．以降（2）と同様に，すべての k について $f_{j+1,k}(\mathbf{s})$ を求め，$\mathbf{s}^i$ が正事例であれば δ_k^i を集合 D_k から取り除く．

（5） $j < M-1$ ならば $j \leftarrow j+1$ として（3）へと戻り，$j \geqq M-1$ ならば終了する．ただし，M は求めるべき状態クラスの数である．これにより，S_0 から S_{M-1} までのクラスを求める．(図 2 では S_0, S_1 などの集合が独立しているが，実際には同一の要素を持つこともある.)

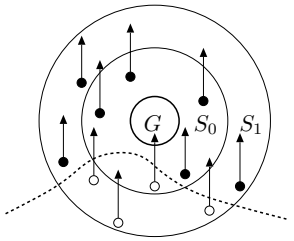
このアルゴリズムは One-against-the-Rest Classifier [12] と類似し，1 つのクラスとそれ以外のクラスの識別を複数回行うことにより，マルチクラス識別を行っている．しかし，ゴール周辺から逐次的に SVM を適用する点，一度正事例となったデータはその後は訓練データとして用いない点が One-againtst-the-Rest Classifier と異なる．

4.3 2 クラス法による状態一般化

4.2 では状態の価値に基づき状態空間を構成したが，この段階では状態空間が細かく構成されており，大域的な行動推定ができない．そこで本手法ではさらに，行動後に価値の上がる状態を SVM を用いて一般化し，全状態空間を行動ごとに 2 つのクラスで構成することで，より大域的な状態空間を構成する．

この状態一般化では，行動 a の正のクラスに属する状態は「行動 a により価値の高い状態に遷移すると推定される状態」を表す．SVM は汎化能力の高い手法であるため，価値関数が緩やかに変化するような比較的価値の予測がしやすい環境であれば，従来の状態一般化法に比べより大域的に最適行動を推定できると考えられる．

行動 a_k に対する状態一般化の例を図 3 に示す．図中の点と矢印は行動前の状態と行動 a_k による状態遷移を表す．**4.2** と同様に，集合 D_k 中の i 番目の行動経験を $\delta_k^i = (\mathbf{s}^i, a_k, \mathbf{s}'^i)$ で表す．$\mathbf{s}^i$ を要素にもつ集合 S_j の中で j が最小のもの（すなわち推定価値が最も高いもの）を S_v とする．また，$\mathbf{s}'^i$ を要素にもつ集合 S_j の中で j が最小のものを $S_{v'}$ とする(注1)．このとき $v > v'$，もしくは S_v が存在せず $S_{v'}$ が存在するならば行動前の状態より行動後の状態の方が価値が高いと推定されるため，$\mathbf{s}^i$ を正事例とする．反対に $v \leqq v'$，もしくは S_v が存在し $S_{v'}$ が存在しないならば $\mathbf{s}^i$ を負事例とする．行動 a_k のすべての行動経験について同様の処理を行い，ガウシアンカーネルを使用した SVM によって識別関数を求める．この識別関数によりすべての状態を正の状態クラス，または負の状態クラスへと分類することができる．これをすべての $a_k \in A$ について行う．



○ positive example ● negative example

図 3 2 クラス法による状態一般化
Fig. 3 State generalization by the 2-class method

本手法は基本的には MDP を対象問題クラスとしているが，より具体的には状態遷移に関する不確実性の少ない問題を想定している．本手法が想定する行動制御などの問題では，同一の行動によって行動後の状態が大きく異なるような確率的状態遷移をすることは少なく，この仮定は妥当であると考えられる．

4.4 行動選択

本手法では，少数の試行による行動経験から学習を行うため，一般化による誤差の影響が少なくない．そこで，ある確率 ϵ でランダムに行動を選択する．そして残りの確率で入力状態を正の状態クラスに分類する行動を選択する．ただし，異なる複数の行動の正のクラスに含まれる場合には，その中からランダムに決定する．また，入力がすべての行動において負の状態クラスに含まれる場合は，すべての行動の中からランダムに選択するものとする．

(注1)：$S_v, S_{v'}$ は価値推定の際に知ることができるので，ここでは識別関数を用いて計算などを行う必要はない．

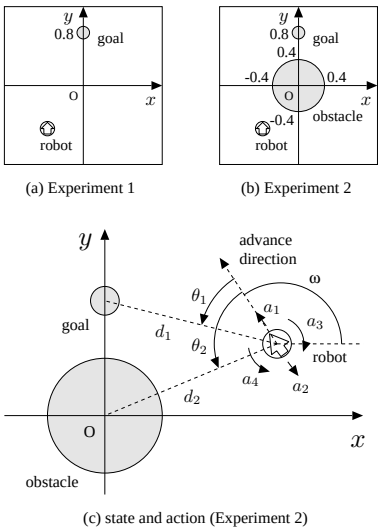


図 4　想定するタスク
Fig. 4　The assumed task

5. 実　　験

5.1　想定するタスク

提案手法の有効性を検証するため，計算機シミュレーションによりロボットナビゲーションの実験を行った．図 4 に想定するタスクを示す．なおこのタスクは [4] で定義されているタスクとほぼ同様のものである．ロボットの学習すべきタスクは唯一のゴールへの到達であり，ロボットがゴールに到達した時のみに正の報酬が得られるものとした．ロボットの初期位置は $-0.5 \leqq x \leqq 0.5$, $y = -0.7$ の範囲でランダムに与えた．また，x 軸の正の方向からロボットの進行方向までの角度を ω [rad]（$-\pi < \omega \leqq \pi$）で表し，初期の進行方向は $\omega = \pi/2$ とした．有効範囲は $-2.0 \leqq x \leqq 2.0$ かつ $-2.0 \leqq y \leqq 2.0$ の範囲とし，ロボットが有効範囲外に出た場合，強制的に初期位置に戻されるものとした．ロボットの行動は前進（進行方向に距離 0.1 移動），後退（進行方向とは逆に距離 0.1 移動），右回転（進行方向を -0.2π[rad] 回転），左回転（進行方向を 0.2π[rad] 回転）の 4 種類とした．

- 実験 1：2 次元状態

ゴールは中心 $(x, y) = (0.0, 0.8)$，半径 0.1 の円とした．ロボットの位置からゴールの中心までの距離を d_1，ロボットの進行方向からゴールの中心までの角度を θ_1[rad]（$-\pi < \theta_1 \leqq \pi$）と表し，状態は (d_1, θ_1) の 2 次元とした．タスクの成功・失敗に関しては，ロボットが有効範囲内から出ることなくスタートから 60 ステップ以内にゴールに到達できたときをタスク成功，それ以外をタスク失敗とした．

- 実験 2：4 次元状態

ゴールは実験 1 と同様に配置し，さらに中心 $(x, y) = (0.0, 0.0)$，半径 0.4 の円形の障害物を配置した．ロボットから障害物の中心までの距離と角度をそれぞれ，d_2, θ_2[rad]（$-\pi < \theta_2 \leqq \pi$）とし，状態は $(d_1, \theta_1, d_2, \theta_2)$ の 4 次元とした．また，ロボットが障害物にぶつかる行動を実行した場合にはロボットは移動せずにその場所にとどまるものとした．ただし，ロボットを点と考えるため，回転行動は必ず実行される．実験 2 では，ロボットが有効範囲内から出ることなくスタートから 100 ステップ以内にゴールに到達できたときをタスク成功，それ以外をタスク失敗とした．

表 1　提案手法のパラメータ
Table 1　Parameters for the proposed method

パラメータ	値
1 試行で収集する行動経験数 T	300
価値推定処理におけるクラス数 M	30
状態間の距離の閾値 d_{th}	0.1
識別関数のパラメータ α の閾値 α_{th}	0.1
行動選択における確率 ϵ	0.25

5.2　パラメータ設定

提案手法におけるパラメータ設定を表 1 に示す．また，カーネル関数のパラメータは，**4.2** で述べた価値推定では $\sigma = 0.2$ とし，**4.3** で述べた状態一般化においては，より大域的に一般化を行うため $\sigma = 0.5$ とした．

5.3　性能の比較と評価

5.3.1　ステップ数とタスク成功率の比較

提案手法，森本ら [2] の手法（以下，Actor-Critic with INGnet），Q-learning，3 層の階層型ニューラルネットワークを用いた誤差逆伝搬法（以下，BP 法）を使用して実験 1，実験 2 を行った．Q-learning では，あらかじめ状態空間を分割し入力を離散値で与えた．実験 1 におけるニューラルネットワークの入力層は 3 ユニット，中間層は 7 ユニット，出力層は 1 ユニットである．実験 2 において，入力層は 5 ユニット，中間層は 11 ユニット，出力層は 1 ユニットとした．また実験 1，実験 2 ともに学習回数は 5000 回とした．入力層は状態ベクトルの各成分を [0,1] に正規化した値を入力とするユニットとバイアスユニットで構成し，中間層（内 1 ユニットはバイアスユニット）の数，学習

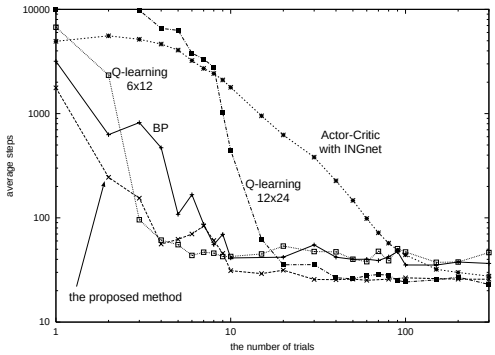


図 5　実験 1 における平均所要ステップ数
Fig. 5　The average number of steps in experiment 1

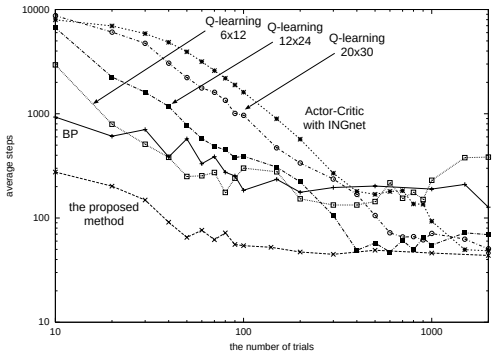


図 7　実験 2 における平均所要ステップ数
Fig. 7　The average number of steps in experiment 2

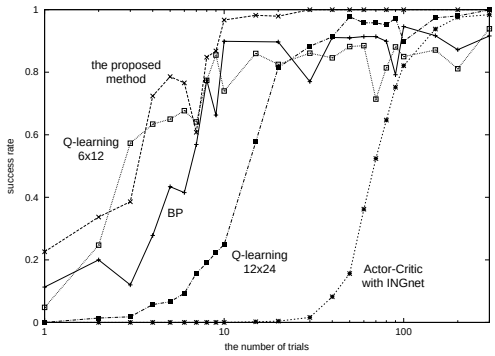


図 6　実験 1 における成功率
Fig. 6　Success rate in experiment 1

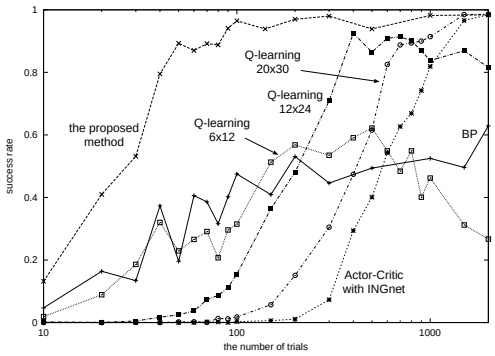


図 8　実験 2 における成功率
Fig. 8　Success rate in experiment 2

回数などのパラメータは予備実験による最適値を用いた．各ニューロンの出力関数には出力値が [0,1] のシグモイド関数を用いた．また，BP 法は状態の一般化のみに適用し，状態価値の推定には提案手法を用いた．

a） 実験 1 の結果

実験 1 における各手法でのゴール到達までの平均ステップ数の変化を図 5 に，タスク成功率の変化を図 6 に示す．このグラフの平均ステップ数と成功率は，ゴール状態に到達するまでに要したステップ数の 1000 回の平均値である．ただし，100 回ごとに学習結果を初期化し乱数の種を変えて再び決められた試行数だけ学習を行い測定した．なお Q-learning 6x12 は，状態入力の距離に関するパラメータの 0 から 3.5 までを 6 段階に等分割し，角度に関するパラメータの $-\pi$ から π までを 12 段階に等分割して状態空間を 72 状態に離散化し Q-learning により学習したものである．同様に Q-learning 12x24 は距離を 12 分割，角度を 24 分割，計 288 状態に離散化したものである．

Q-learning 6x12 は状態空間の分割が不足しているため性能が悪くなっているが，その他の Q-learning では最終的に 25 ステップ程度に収束し，成功率も 1 に近い値が得られている．しかし Q-learning 12x24 では収束に 100 試行程度，Actor-Critic with INGnet では 200 試行程度要しているのに対し，提案手法はおよそ 30 試行で収束しており，提案手法は従来手法に比べ非常に速く収束することが確認された．また提案手法では 10 試行以内の学習初期において平均ステップ数が大きく減少している．BP 法も学習初期において平均ステップ数が大きく減少しているが，最終的なステップ数，成功率ともに提案手法に劣っている．BP 法が SVM と同じ訓練データを用いているにもかかわらず，このような差異が生じることから，提案手法において SVM による一般化が有効に働いていると考えられる．

b） 実験 2 の結果

実験 2 における平均ステップ数の変化を図 7 に，タ

表 2　状態・基底数の比較
Table 2　Comparison of the number of states (bases)

	実験 1	実験 2
Q-learning 6x12	72	5184
Q-learning 12x24	288	82944
Q-learning 20x30	600	360000
Actor-Critic with INGnet	113 (200)	1001 (1500)
提案手法	861 (40)	4476 (140)

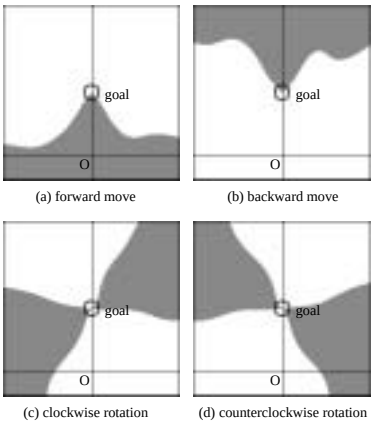
スク成功率の変化を図 8 に示す．平均ステップ数と成功率は，実験 1 と同様の方法で測定した 1000 回の平均値である．また実験 1 で用いた比較対象の他に Q-learning 20x30 を追加した．Q-learning 20x30 は，距離を 20 分割，角度を 30 分割して学習を行ったものである．実験 2 では入力が 4 次元となるため，Q-learning 6x12 で 5184 状態，Q-learning 12x24 で 82944 状態，Q-learning 20x30 で 360000 状態となる．

Q-learning 6x12 と Q-learning 12x24 では学習が進むにつれタスク成功率が低下する傾向が見られた．これは状態空間の分割不足により不完全知覚が発生し，誤った学習が行われたことが原因と考えられる．Q-learning 20x30 では高い性能が得られているが，状態数が多いため学習速度は他の Q-learning に比べ遅くなっている．収束に要する試行数は Q-learning 20x30, Actor-Critic with INGnet が共に約 1500 試行であったのに対し，提案手法では 150 試行程度でこれらの手法とほぼ同じ収束値が得られた．また，実験 1 よりも状態入力の次元数が増え，構成すべき状態空間がより複雑になり BP 法の性能が悪くなっている．これらの結果から，本提案手法は実験 2 のような多次元の状態入力をもつ環境にも高速に適応できることが確認された．

5.3.2　メモリ量の比較

Q-learning 用に離散化した状態空間の全状態数，Actor-Critic with INGnet において基底関数の中心となるベクトル数，提案手法の状態一般化において最終的に必要となるベクトル数を表 2 に示す．Actor-Critic with INGnet と提案手法の値は 10 回の実験による平均値であり，括弧内の値は学習試行数である．

実験 1，2 ともに Actor-Critic with INGnet が提案手法よりも少ない状態ベクトル数で環境に適応した．これは Actor-Critic with INGnet が連続的な行動空間でも対応できる手法であるのに対し，提案手法は行動毎に状態空間を構成するためであると考えられる．Q-learning では，実験 2 のような 4 次元状態空間を



(a) forward move　(b) backward move
(c) clockwise rotation　(d) counterclockwise rotation

図 9　実験 1 における人工的な行動経験集合からの状態空間の構成（グレー領域は行動後に高価値状態へ遷移すると推定される領域）
Fig. 9　State space construction by the artificial experience set in experiment 1

離散化した場合，5000 状態程度では十分な性能が得られず，高い性能を得るためには非常に多くの状態が必要となった．一方 Actor-Critic with INGnet や提案手法では実験 2 でもそれほど多くのベクトルを必要とせず環境に適応した．なお BP 法については，今回状態一般化のみを BP に置き換えたため，SVM を用いた提案手法とほぼ同量のメモリ量が必要となる．これらのことから提案手法は学習速度やメモリ量の観点から多次元連続状態空間に対して有効な手法であるといえる．

5.4　状態空間の構成

提案手法における状態空間の構成を，実験 1 と実験 2 の環境で視覚的に確認した．

5.4.1　人工的な行動経験による構成

人工的に行動経験集合を生成することにより，本手法における理想的な状態空間構成を調査した．行動経験集合は次のように生成した．

- ロボットの位置を $x = -1 + 0.08i$（$i = 0, \ldots, 25$），$y = -0.92 + 0.08j$（$j = 0, \ldots, 27$）とし，進行方向を $\omega = -17\pi/18 + k\pi/9$（$k = 0, \ldots, 17$）に設定する．ただしゴールや障害物の範囲内に含まれるものは除外する．
- 設定された位置からある行動を実行し，その行動経験を収集する．
- すべての i, j, k について，すべての行動を実行し行動経験集合を生成する．

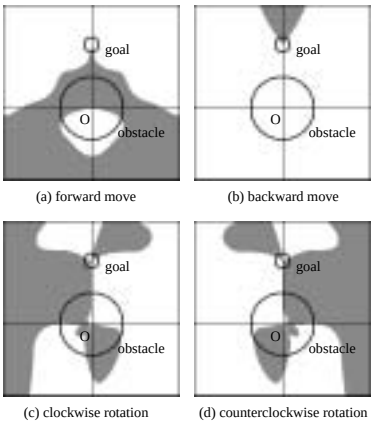


図 10　実験 2 における人工的な行動経験集合からの状態空間の構成（グレー領域は行動後に高価値状態へ遷移すると推定される領域）
Fig. 10　State space construction by the artificial experience set in experiment 2

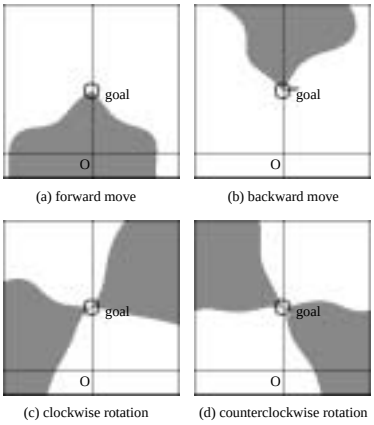


図 11　実験 1 における状態空間の構成例（グレー領域は行動後に高価値状態へ遷移すると推定される領域）
Fig. 11　An example of state space construction in experiment 1

5.4.2　状態空間構成に対する考察

実験 1，実験 2 における人工的な行動経験集合からの状態空間の構成をそれぞれ図 9，図 10 に図示する．また実験 1 において 40 試行学習後の状態空間の構成の様子を図 11 に，実験 2 において 140 試行学習後の構成の様子を図 12 に，実験 2 において BP 法を用いた場合の 140 試行学習後の左回転に対する構成の様子を図 13 に示す．これらの図のグレー領域は，ロボットが真北（$\omega = \pi/2$ の方向）を向いているときの各行動における正の状態クラス，すなわち行動によってより高価値の状態に遷移すると推定される領域を表している．

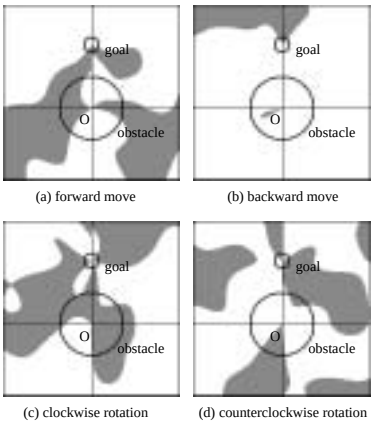


図 12　実験 2 における状態空間の構成例（グレー領域は行動後に高価値状態へ遷移すると推定される領域）
Fig. 12　An example of state space construction in experiment 2

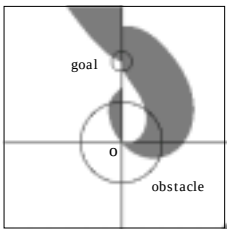


図 13　実験 2 における BP 法を用いた場合の左回転に対する状態空間の構成例（グレー領域は行動後に高価値状態へ遷移すると推定される領域）
Fig. 13　An example of state space construction for counterclockwise rotation in experiment 2 with BP

図 11 は図 9 とほぼ同じ構成であることから，実験 1 において 40 試行学習後に提案手法における理想の状態空間構成が得られたことが確認された．図 11 から，ロボットから見て前方にゴールがあれば前進する，右やや前方にゴールがあれば右回転するなどといったことが学習されたことがわかる．一方，図 12 を図 10 と比較すると，実験 1 の時に比べ理想とは少し異なる構成となった．これは，状態がより高次元になったため，収集された状態の密度が低くなり誤分類が生じたためであると考えられる．しかし図 12 から，障害物がロボットの前方にあれば前進をせず，前方やや左にあるならば右回転し，やや右にあるならば左回転するといった障害物を迂回する行動が学習されたことが確認できる．また，図 12(d) と図 13 を比較すると，BP 法は SVM に比べて，行動経験を獲得することが困難

なゴールから離れた部分において，ゴールに到達するに適切な状態空間を構成できていない．一方で SVM は行動経験を獲得することが困難な部分でも合理的な決定を行い，さらに障害物付近においても好ましい状態空間の構成の獲得が可能であることが確認できる．

6. むすび

本論文では，Support Vector Machine を用いて連続状態を一般化する強化学習的手法を提案した．実験により提案手法が従来法に比べ少ない試行で環境に適応できることを示した．また提案手法により環境に適応した状態空間構成が得られることを確認した．

4.1 で述べたように，本手法は重要な行動経験のみを用いることによって SVM の計算にかかるコストを削減しているが，高次元の状態を考えた場合，効果的に SVM の訓練データを削減することが必要となる．そこで今後の課題として，より効果的な行動経験集合の縮小方法を検討することなどが挙げられる．また，SVM による性能向上を確認するため，他の状態一般化手法との比較についても今後行う予定である．

文　献

[1] J.A. Boyan and A.W. Moore, “Generalization in Reinforcement Learning: Safely Approximating the Value Function,” in G. Tesauro, D.S. Touretzky, and T.K. Leen, eds., Advances in Neural Information Processing Systems 7, pp.369–376, MIT Press, 1995.

[2] 森本 淳，銅谷 賢治，“強化学習を用いた高次元連続状態における系列運動学習：起き上がり運動の獲得，”信学論（D-II），Vol.J82-D-II, No.11, pp.2118–2131, 1999.

[3] W.T.B. Uther and M.M. Veloso, “Tree based discretization for continuous state space reinforcement learning,” in Proc. AAAI-98, Madison, WI, 1998.

[4] 矢入 健久，堀 浩一，中須賀 真一，“複数行動結果を考慮した最尤推定に基づく状態一般化法，”人工知能学会誌，Vol.16, No.1, pp.130–140, 2001.

[5] 浅田 稔，野田 彰一，細田 耕，“ロボットの行動獲得のための状態空間の自律的構成，”日本ロボット学会誌，Vol.15, No.6, pp.886–892, 1997.

[6] 上野 敦志，堀 浩一，中須賀 真一，“自律エージェントのための状況認識と行動規則の同時学習，”第 30 回人工知能基礎論研究会，pp.19–24, 1997.

[7] C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, Vol.20, No.3, pp.273–297, 1995.

[8] R.S. Sutton and A.G. Barto, “Reinforcement Learning,” MIT Press, 1998.

[9] C.J.C.H. Watkins and P. Dayan, “Technical Note: Q-Learning,” Machine Learning, Vol.8, No.3, pp.279–292, 1992.

[10] J. Moody and C.J. Darken, “Fast Learning in Networks of Locally-Tuned Processing Units,” Neural Computation, Vol.1, pp.281–294, 1989.

[11] J. Platt, “Fast Training of Support Vector Machines using Sequential Minimal Optimization, in B. Schölkopf, C. Burges, and A. Smola, eds., Advances in Kernel Methods – Support Vector Learning, pp.185–208, MIT Press, 1999.

[12] J. Weston and C. Watkins, “Multi-class support vector machines,” Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, TW20 0EX, UK, 1998.

（平成 x 年 xx 月 xx 日受付）

後藤　亮

平成 13 名工大・電気情報卒．現在，同大学院博士前期過程在学中．強化学習に関する研究に従事．

松尾　啓志　（正員）

昭和 58 名工大・情報卒．昭和 60 同大学大学院修士過程了．同年松下電器産業（株）入社．平 1 名工大大学院博士課程了．同年名工大・電気情報・助手．平 5 名工大・電気情報・講師，平 8 名工大・電気情報・助教授，現在に至る．分散システム，画像認識，分散協調処理に関する研究に従事．工博．情報処理学会，人工知能学会，IEEE 各会員．

State Generalization Method with Support Vector Machines in Reinforcement Learning

Ryo Goto Hiroshi Matsuo

Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan

abstract

The conventional reinforcement learning assumes discrete state space. Therefore, it is necessary to make states discrete in order to handle continuous state environments. However, if a simple discretization is applied, the number of states increases exponentially with the dimension of the state space, and the learning time increases. In this paper, we propose a state generalization that is able to quickly adapt to environments by using Support Vector Machines. We conducted an experiment on the simulation task that navigates a robot to a goal. As a result of the experiment, the proposed method adapted to environment by a few trials.

keywords

reinforcement learning, Support Vector Machine, state generalization