計測自動制御学会論文集
Vol.42, No.12, 1/9 (2006)

ランダムタイリングとGibbs-samplingを用いた多次元状態-行動空間における強化学習†

木村　元*

Reinforcement Learning in multi-dimensional state-action space using random tiling and Gibbs sampling

Hajime KIMURA*

In real-robot applications, learning controllers are often required to obtain control rules over high-dimensional continuous state-action space. Random tile-coding is a promising method to deal with high-dimensional state space for representing the state value function. However, there is no standard reinforcement learning scheme to deal with action selection in high-dimensional action space, especially the probability of action variables are mutually dependent. This paper introduces a new action selection scheme using random tile-coding and Gibbs sampling, and shows the Q-learning algorithm applying the proposed scheme. We demonstrate it through a Rod in maze problem and a redundant arm reaching task.

Key Words: reinforcement learning, Q-learning, random tile coding, Gibbs sampling, action selector

1. はじめに

強化学習は，ロボットの制御規則を自ら発見し改善していくための学習制御方法として有望である[4) 12)]．脚型移動ロボットやヒューマノイド型ロボットなどの多数のアクチュエータを有するロボットにおいて動作規則を学習する場合、状態空間だけでなく行動空間についても高次元で膨大な空間であり、状態の評価方法だけでなく行動空間での行動選択方法および学習方法にも工夫が必要である．

代表的な強化学習法である Q-learning[11)] や SARSA[10)] は離散的な状態・行動を対象としている．そのため連続な状態空間や行動空間における状態評価関数 (state value function) や状態-行動評価関数 (state-action value function) を表現するために関数近似として CMAC を用いる方法[10)]，ファジィを用いる方法[2)]，過去の経験を用いて補間する方法[1) 8)] などが提案されている．

特に状態空間の次元数が高い場合，次元の呪いを回避するための有望な手段としてタイルコーディングの一種であるハッシュを使うべきという主張がされている[10)]．これは，ランダムに選ばれたいくつかの状態変数で作られる空間中に大きさや位置がランダムなタイルを配置し，そのタイルを，状態価値関数を表現するための特徴ベクトル要素1つに対応させる．状態入力がそのタイル領域内に入ると対応する特徴ベクトル要素の値が1になり，それ以外の場合はゼロになる．このランダムタイルによる状態特徴ベクトル生成は，タイルの初期配置が学習性能に大きな影響を与える問題点があるが，依存関係の考慮が必要な状態変数集合とそうでない状態変数を区別したり，重要な状態変数値の組合せをエキスパートの知識から予め与えておくことが容易など多くの利点を有する．このように，高次元空間において状態評価値や状態-行動評価値の汎化については多くの手法が提案されているが，高次元の行動空間における行動選択方法および学習方法については，あまり注目されてこなかった．連続な行動空間を扱うための最も単純な接近法としては，行動空間を全てメッシュに区切り，状態入力に応じて各行動メッシュに対して Q 値に応じた確率を割当てることが考えられるが，行動空間の次元が高くなると「次元の呪い」によってたちまち空間爆発を起こし，記憶容量的にも計算量的にも実行不能になってしまう．また，せっかく膨大な空間を「汎化」によって少ないパラメータで扱おうとしているのに，細かい離散化によってパラメータ数を爆発させては意味がない．しかしながら離散化を粗くしすぎてもきめ細かな行動選択ができないため制御の質が落ちるジレンマがあった．

本論文では，まずランダムタイリングを用いて状態-行動

† 計測自動制御学会 システム・情報部門学術講演会 (SSI2005) にて発表

* 九州大学 大学院工学研究院 海洋システム工学部門
* Dept. of Marine Engineering, Graduate School of Engineering, Kyushu University

(Received January 6, 2006)
(Revised October 1, 2006)

空間を汎化し，Q 値や行動選択確率分布を表現する方法を提案する．次に，多次元で連続な行動空間を細かく離散化するが，Gibbs サンプリングによって空間爆発を回避しつつ行動選択を行う方法を提案する．Q-learning や SARSA など，これまで代表的強化学習手法だったにもかかわらず，行動選択の困難さから膨大な行動空間を持つ強化学習問題への実装が困難だったが，本手法により容易に実装できることを示す．

2. 問題の定式化

状態空間を $\mathcal{S}$，行動空間を $\mathcal{A}$，上下界を持つ実数の集合を $\mathcal{R}$ と表す．各時刻 t でエージェントは状態観測 $s_t \in \mathcal{S}$ に基づいて行動 $a_t \in \mathcal{A}$ を実行し，状態遷移に伴う報酬 $r_t \in \mathcal{R}$ を得る．本論文が環境のモデルとして仮定するマルコフ決定過程 (MDP) では，一般に次の状態や報酬は確率的で，その分布は s_t と a_t にのみ依存する．MDP では次の状態 s_{t+1} は遷移確率 $T(s_t, a, s_{t+1})$ に従って決まり，報酬 r_t も期待値 $r(s_t, a)$ によって与えられる．エージェントは予め $T(s_t, a, s_{t+1})$ や $r(s_t, a)$ についての知識を持っていない．強化学習の目的はエージェントのパフォーマンスを最適化する政策を得ることである．無限期間のタスクにおける評価規範として，以下の割引報酬の合計を考える．

$$V_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \ , \tag{1}$$

割引率 $0 \leq \gamma \leq 1$ は未来に得るであろう報酬の現時点での重要度を表し，V_t は時刻 t の評価値 (value) を表す．MDP における評価関数は以下に定義される．

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, \pi \right] \ , \tag{2}$$

ただし $E\{\cdot\}$ は期待値を表す．MDP における学習の目的は，各状態 s において式 (2) で定義される評価値を最大化するような最適政策を見つけることである．本論文で扱う環境では，状態空間 $\mathcal{S}$ および行動空間 $\mathcal{A}$ は多次元空間で表されるものとする．

3. ランダムタイリングによる高次元状態-行動空間の汎化方法の提案

高次元空間において関数を汎化する方法は多種多様であるが，本論文では多数のランダムタイルを用いる方法を提案する．これは状態空間における汎化のためのタイルコーディングの一種であるハッシュに基づいたアイデア [10] である．ある 1 つのランダムタイル $f(x)$ は，入力空間 x を構成する次元のうち，任意の 1 次元以上の複数次元で定義される部分空間中のある矩形領域を表し，入力された座標 x がその矩形領域である場合 $f(x) = 1$ を出力し，それ以外の場合 $f(x) = 0$ である．矩形領域を構成する次元や矩形の範囲・大きさなどは，タイル毎にランダムに与える．このように，ある 1 つのタイルは空間を構成する次元の一部分で構成される空間の，とある領域として処理するが，このタイルを空間全体からみると，定義される部分空間中では範囲の限定された矩形だが，それ以外の空間では全領域をカバーするタイルと同義である．このようなランダムタイルを多数用いて重み付け線形和することにより，空間全体に対して関数近似（汎化）を行う．本論文では，状態空間に対するランダムタイル $f_j(s)$（ただし $j \in \{1, 2, \ldots T_s\}$）および行動空間に対するランダムタイル $g_k(a)$（ただし $k \in \{1, 2, \ldots T_a\}$）の 2 種類のタイル集合を用いて特徴量ベクトルを生成する．特徴量ベクトルを用いて関数近似を行う場合，絶対和ノルムが一定値（理想的には 1）であることが望ましい．そこで $2T_s$ 個の要素を持つ状態特徴量ベクトル $F(s) = (F_1(s), F_2(s), \cdots F_{2T_s}(s))$ および $2T_a$ 個の要素を持つ行動特徴量ベクトル $G(a) = (G_1(a), G_2(a), \cdots G_{2T_a}(a))$ を用いる．このとき，特徴量ベクトルの前半の要素はタイルのベクトルそのままで、後半の要素は 1 から各タイルの値を引いた値とする．

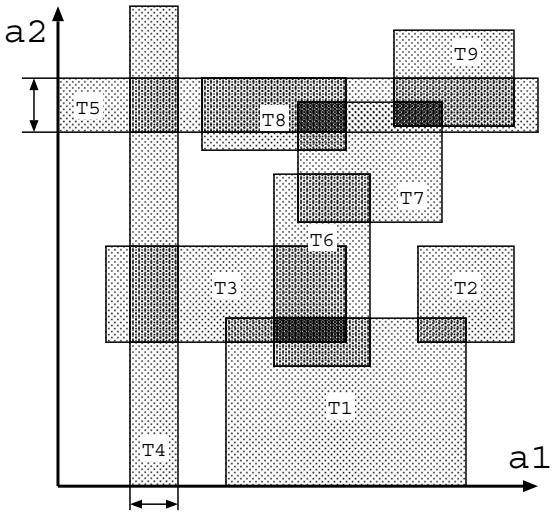


Fig. 1 An example of the random-tiling using 9 tiles in two-dimensional action space. The tile T4 is defined by the arrow width in the sub-space a1, and then it covers the other all sub-space a2. The tile T5 is defined similarly.

$$F_i(s) = \begin{cases} f_i(s) & \text{, where } \ i \leq T_s \\ 1 - f_{(i-T_s)}(s) & \text{otherwise} \end{cases} \tag{3}$$

$$G_i(a) = \begin{cases} g_i(a) & \text{, where } \ i \leq T_a \\ 1 - g_{(i-T_a)}(a) & \text{otherwise} \end{cases} \tag{4}$$

状態 s，行動 a に対する状態-行動評価値（Q 値）は，特徴量ベクトル $F(s), G(a)$ の各要素と $2T_s \times 2T_a$ コの重み変数 w_{jk}（ただし $j = 1, \cdots, 2T_s$, $k = 1, \cdots, 2T_a$ ）を用いて以下のように計算する：

$$Q(s, a) = \frac{1}{T_s T_a} \sum_{j=1}^{2T_s} \sum_{k=1}^{2T_a} F_j(s)\, G_k(a)\, w_{jk} \tag{5}$$

式 (5) の $\frac{1}{T_s T_a}$ は， $F_j(s)\, G_k(a)$ の組合せで表される特徴量

ベクトルのノルムを 1 にするための正規化定数である．この Q 値を温度パラメータ T で除したボルツマン分布に従い，行動を確率的に選択する．各重み変数の値を調節することにより，Q 値および行動選択確率分布が変化する．本手法には以下の特徴がある：

- ランダムタイルが定義される行動空間の部分空間によって行動変数間の依存関係が表現できる．
- 定義されている行動部分空間が互いに干渉していないランダムタイルが存在することにより，互いの空間において独立に行動を出力することが必要なタスクを学習できる．例えば，ある行動変数の集合は固定された特定の値だけ，別の行動変数の集合では 0〜1 の値を一様に出力することが求められるような場合にも対処できる．

このように興味深い特徴が期待されるが，高次元の行動空間全体に対して定義された重み関数のボルツマン分布に従う行動選択は，そのまま実行しようとすると組合せ爆発を起こしてしまうため工夫が必要である．

4. 高次元空間での複雑な確率分布によるサンプリングと学習

4.1 従来手法：フラットな行動選択

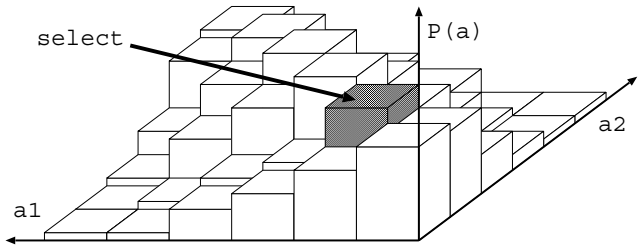


Fig. 2 An action-selection with a finite-grid approach in two-dimensional action space. The probability function of the action selection $P(a)$ is represented by quantizing the action space $a1 - a2$ into a finite number of cells.

ランダムタイリングによって高次元空間において複雑な状態-行動評価関数（Q 関数）や行動選択確率分布関数が表現できるが，この分布に従って行動を選択するための最も原理的に単純な方法は，フラットな行動選択法である．これは，行動空間を全て細かい格子状に離散化し，各超矩形領域に対して確率を割当て、ルーレット選択によっていずれかの超矩形領域に相当する行動を選択するものである．本論文では，連続な行動空間を各次元毎に等しく D 分割して格子状に離散化する．よって行動空間を N 次元と仮定すると D^N コの行動 a_i $(i = \{1, \cdots, D^N\})$ へ離散化される．行動は，各行動 a_i 毎に定義された確率 $P(a_i)$ に従って，どれか 1 つが選ばれる．Fig.2 は 2 次元の行動空間において 2 次元行動空間を格子状に分割し，各メッシュの持つ確率に応じて行動選択する様子を示す．フラットな行動選択では，2 次元空間 $a1 - a2$ で離散化された全ての格子について確率を計算し，その比率に基づいて格子を一つ選択して行動出力とする．

本論文では，行動の確率 $P(a_i)$ は，一般に状態 s において行動 a_i に割り当てられた Q 値 $Q(s, a_i)$ に応じた以下のボルツマン分布とする：

$$P(a_i) = \frac{\exp(Q(s, a_i)/T)}{\sum_{j=1}^{D^N} \exp(Q(s, a_j)/T)} \qquad (6)$$

ただし T は正の値をとる温度パラメータであり，値が大きいと全行動の確率が均一に近づき，ゼロに近づくと大きな Q 値を持つ行動の確率が大きくなる．このとき，各行動の $Q(s, a_i)$ はランダムタイルの値と重み変数の線形和から計算される．

このフラットな行動選択方法は，式 (6) の示すとおり全行動空間の Q 値を調べる必要があるため，行動次元数が 2〜3 次元程度ならば実行可能だが，次元数が高くなると離散化した行動空間が指数的に増大するため，記憶容量的にも速度的にも実行不能になる．

4.2 Gibbs サンプリングによる行動選択

本論文では高次元空間における確率分布に従って効率良くサンプリングするための方法の 1 つである Gibbs サンプリングを用いる行動選択法と適正度の計算法を提案する．Gibbs サンプリングとは，高次元の確率変数において，注目している次元以外の次元の変数の値を固定し，そのときの条件付確率分布を用いて 1 次元ずつ順番にサンプルを行っていく処理を全ての次元に対して十分な回数繰返し，最終的に得た値をサンプルとする Markov chain Monte-Carlo 法（MCMC 法）の一種である [3]．Fig.3 は 2 次元の行動空間における Gibbs サンプリングの様子を示す．2 次元空間 $a1 - a2$ で離散化された全ての格子における確率は Fig.2 のフラット選択の場合と同じだが，1 次元ずつサンプルを行っていくので，Fig.2 のような全行動空間における確率を全て計算するような処理が不要であることが分かる．Fig.3 の右上 (1) は $a2$ を固定した条件付確率により $a1$ をサンプルし，左下 (2) では (1) で選んだ $a1$ を固定した条件付確率により $a2$ をサンプルしている．これで反復 1 回分であり，さらに右下 (3) では (2) で選んだ $a2$ を固定した条件付確率により再び $a1$ を選んでいる．このような反復を十分な回数繰返した結果得られた $a1, a2$ を最終的な行動出力とする．Gibbs サンプリングでは，反復回数についての理論的な下限については明らかにはされておらず，実験的に決められているのが実情である．

Fig.3 で示した例題のように 2 次元程度の空間を 6×6 に粗く分割した程度では，フラットな行動選択との計算量的な差はあまりないが，行動次元数や分割数が増加すると，その差は顕著に現れる．例えば行動空間が 8 次元で各次元を 10 分割する場合，フラットな行動選択においては各メッシュにおける重みの計算を 10^8 =1 千万回行わなければならないが，Gibbs サンプリングでは $10 \times 8\times$（反復回数）程度であり，反復回数を数十回程度に抑えればフラットな行動選択法に比べて 1 万分の 1 程度の計算量で済むことになる．

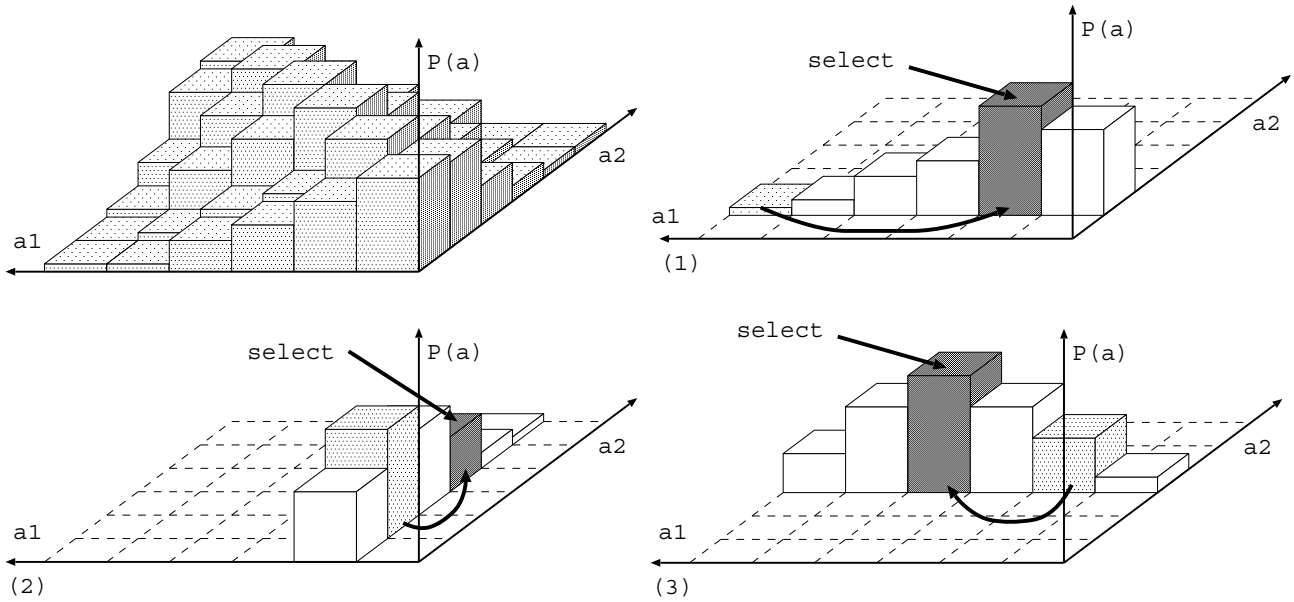


Fig. 3 An example of the Gibbs-sampling scheme with the finite-grid approach in two-dimensional action space. The top left shows a joint distribution of the action selection probability. The top right (1) represents selecting action $a1$ following the conditional probability given $a2$. The bottom left (2) represents selecting action $a2$ following the conditional probability given $a1$ at the top right (1). The bottom right (3) represents selecting action $a1$ again, following the conditional probability given $a2$ at the bottom left (2).

Gibbs サンプリングでは行動選択確率分布自体は式 (6) と同じだが，各行動座標軸毎に処理を行うため，記法を以下のように対応させる．フラットな行動選択の場合同様，行動の次元数 N で，行動空間は各次元毎に D 分割により離散化されているものとする．ある多次元行動 a について，各行動次元毎に分解して次のように表す：$a = (a^1, a^2, \ldots a^N)$ ただし各次元の要素 a^n (ただし $n \in \{1, 2, \cdots N\}$) は $a^n \in a^n_d$ ただし $d \in \{1, 2, \cdots D\}$ である．ある状態 s 行動 a に対する Q 値は，$Q(s, a) = Q(a^1, a^2, \ldots a^N|s)$ と表す．この Q 値は，フラット選択における値と同一である．Gibbs サンプリングの t 回目の反復においてサンプルされた行動要素を $a^1(t), a^2(t), \ldots a^N(t)$ と記する．このとき，$t+1$ 回目の反復における行動要素は以下の確率分布に従ってサンプルされる：

$$
\begin{aligned}
a^1(t+1) &\sim P(a^1|a^2(t), a^3(t), \ldots a^N(t)) \\
a^2(t+1) &\sim P(a^2|a^1(t), a^3(t), \ldots a^N(t)) \\
&\vdots \\
a^N(t+1) &\sim P(a^N|a^1(t), a^2(t), \ldots a^{N-1}(t))
\end{aligned}
$$

ここで，条件付き確率 $P(a^n_i|a^1, a^2, \ldots a^N)$ は，以下のボルツマン分布で与えられる：

$$
\begin{aligned}
&P(a^n_i|a^1, a^2, \ldots a^N) \\
&= \frac{\exp(Q(a^1, a^2, \ldots a^n_i, \ldots a^N|s)/T)}{\sum_{d=1}^{D} \exp(Q(a^1_d, a^2_d, \ldots a^n_d, \ldots a^N_d|s)/T)} \qquad (7)
\end{aligned}
$$

フラットな行動選択と比べると，考慮すべき行動が 1 次元ずつになり，式 (6) と同じような処理を繰り返し行うようになっただけである．このとき，全行動空間について Q 値を調べる必要がないのが大きな利点である．

4.3 強化学習アルゴリズム

ランダムタイルによる空間汎化と Gibbs サンプリングによる行動選択を代表的な強化学習アルゴリズムである Q-learning 法と組み合わせる．第 3 章で説明したように，状態空間のランダムタイル $f_j(s)$ (ただし $j \in \{1, 2, \ldots T_s\}$) および行動空間のランダムタイル $g_k(a)$ (ただし $k \in \{1, 2, \ldots T_a\}$) の 2 種類のタイル群を用いて，$2T_s$ 個の要素を持つ状態特徴量ベクトル $F(s) = (F_1(s), \cdots F_{2T_s})$ および $2T_a$ 個の要素を持つ行動特徴量ベクトル $G(s) = (G_1(s), \cdots G_{2T_a})$ を式 (3), (4) より生成する．状態 s，行動 a に対する状態-行動評価値（Q 値）は，特徴量ベクトル $F(s)$, $G(s)$ の各要素と $2T_s \times 2T_a$ コの重み変数 w_{jk}（ただし $j = 1, \cdots 2T_s$, $k = 1, \cdots 2T_a$ ）を用いて以下のように計算する：

$$
Q(s, a) = \frac{1}{T_s\, T_a} \sum_{j=1}^{2T_s} \sum_{k=1}^{2T_a} F_j(s)\, G_k(a)\, w_{jk} \qquad (8)
$$

エージェントは状態 s を観測し，以下のボルツマン分布に従って行動 a_i を選択し実行する：

$$
P(a_i) = \frac{\exp(Q(s, a_i)/T)}{\sum_{j=1}^{D^N} \exp(Q(s, a_j)/T)} \qquad (9)
$$

ただし行動は式 (7) の Gibbs サンプリングにより選択するため，全行動について Q 値を計算する必要は無い．行動選択後，報酬 r と遷移先の状態 s' を観測する．遷移先での状

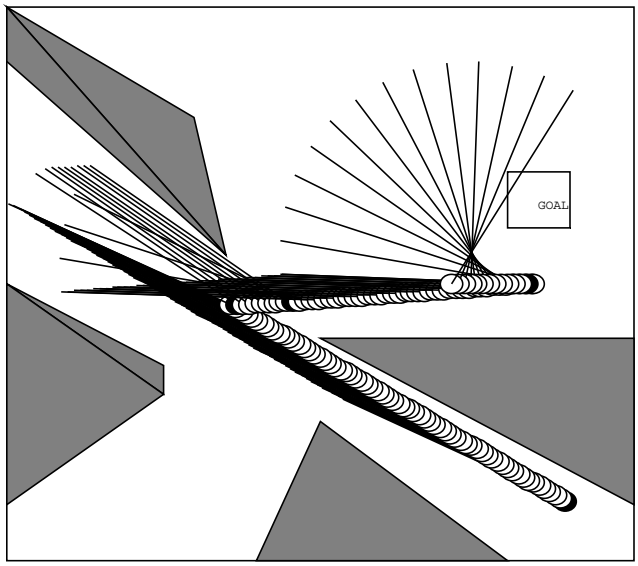


Fig. 4 A learned example behavior of a rod in maze problem. The black circles of the bottom of the rod represent the positions at which the learner makes decisions.

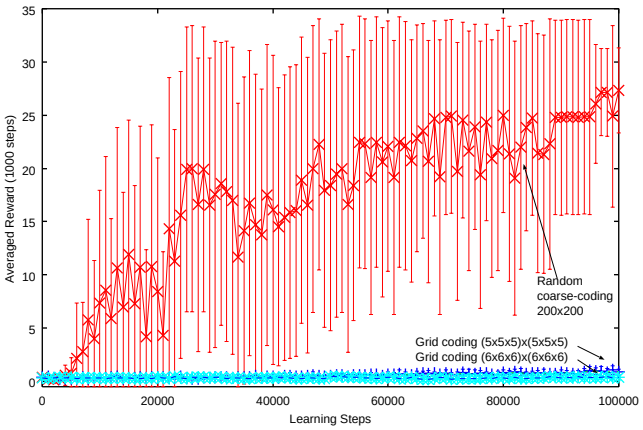


Fig. 5 Learning results averaged over 10 trials in the rod-in-maze problem. The vertical axis gives the averaged reward over making 1000 decisions (steps). The number of iteration in Gibbs sampling is 15.

態 s' における Q 値を使い，もとの状態 s で実行した行動 a_i の Q 値を更新する：

$$\text{TD\_error} = r + \gamma \max_a Q(s', a) - Q(s, a_i) \qquad (10)$$

$$Q(s, a_i) \leftarrow Q(s, a_i) + \alpha\, \text{TD\_error} \qquad (11)$$

ただし γ は割引率，α は学習率（$0 \leq \alpha \leq 1$）である．ここで Q 値は式 (8) で表されているので，式 (11) で示される Q 値の更新は，重みパラメータ w_{jk} を式 (10) を用いて以下のように更新することで実現する：

$$w_{jk} \leftarrow w_{jk} + \frac{F_j(s)\, G_k(a_i)}{T_s\, T_a}\, \alpha\, \text{TD\_error} \qquad (12)$$

ここで $\max_a Q(s', a)$ を探す必要があるが，これはボルツマン分布の温度パラメータ T をゼロに近づけて Gibbs サンプリングによって得た行動の Q 値で代用したり，あるいは Gibbs サンプリングを行う過程で見つかった最大の Q 値を用いるなど，大規模問題では近似的に与えざるを得ないという問題点はあるが，真に最大の Q 値でなくても，選んだ行動の Q 値を充てれば SARSA アルゴリズムになるので，on-policy 学習をすることは保障される．

5. 実験

Fig.4 に示すように，Rod in maze 問題[7] に修正を加えた問題へ本手法を適用する．状態空間 $s = (x, y, \theta)$ は 3 次元で，(x, y) は棒の下端の円の中心座標を表し，$0 \leq x \leq 1, 0 \leq y \leq 1$ の矩形範囲になければならない．θ は棒と水平面のなす角度を表し，$0 \leq \theta \leq \pi$ である．行動 $a = (x_d, y_d, \theta_d)$ も 3 次元で，目標とする位置座標を表す．行動により目標座標が与えられると，その方向へ $s = (x, y, \theta)$ 空間中を直線的に移動し，途中で障害物あるいはゴール領域に触れるか，目標座標に到達すると，イベントが発生して次の意思決定を行う．ゴール領域は $0.8 \leq x \leq 0.9$, $0.6 \leq y \leq 0.7$ で，棒の長さは 0.4 である．障害物の座標は，障害物 1: $(1.0, 0.1) - (0.5, 0.4) - (1.0, 0.4)$, 障害物 2: $(0.4, 0.0), (0.5, 0.25), (0.8, 0.0)$, 障害物 3: $(0.0, 0.5), (0.25, 0.35), (0.25, 0.3), (0.0, 0.1)$, 障害物 4: $(0.0, 1.0), (0.3, 0.8), (0.35, 0.55), (0.0, 0.9)$ の領域で定義される．棒がゴール領域に到達すると 100 の報酬が与えられて初期状態 $(0.9, 0.1, \frac{5}{6}\pi)$ へ戻り，それ以外では報酬 0 である．

強化学習エージェントは，3 次元の状態入力の各次元の変数を 100 分割して離散化する．3 次元の行動 (x_d, y_d, θ_d) についても同様に各次元を 100 分割して離散化する．よって離散空間の強化学習問題としては状態数 $100^3 = 10^6$，行動数 $100^3 = 10^6$，状態-行動空間は 10^{12} という膨大な空間になる．この状態空間に対し，ランダムタイルを 200 個生成して状態特徴ベクトル $f_j(s)$ とする．このタイル群は，各状態次元についてタイル矩形領域の部分空間を構成する要素として選択する確率を 0.6 としてランダムに作成する．ただし各次元の要素を 1 つも選択しなかったタイルは除外し再生成する．ランダムタイルの矩形領域については，選択された部分空間において，離散化された空間の境界に合わせてランダムに生成する。ただしタイルの大きさについては各次元の離散化された定義域の 0〜25 %までの範囲を一様な乱数で生成し，タイルの中心位置は，各次元の離散化された定義域の全範囲を一様な乱数で生成する．行動のランダムタイル $g_k(a)$ も同様に 200 個生成する．Q-learning アルゴリズムにおける $\max Q$ を探す部分は，Gibbs サンプリングの過程で記録された最大の Q 値で代用した．Gibbs サンプリングの反復回数は 15 回，割引率 $\gamma = 0.9$，ボルツマン分布の温度パラメータ $T = 0.4$ で一定，学習率 $\alpha = 0.5$ に設定する．比較対象として，状態と行動空間を $5 \times 5 \times 5 = 125$ 個のタイルへ均一にグリッド分割した場合と，$6 \times 6 \times 6 = 216$ 個のタイルに均一に分割した場合について調べる．

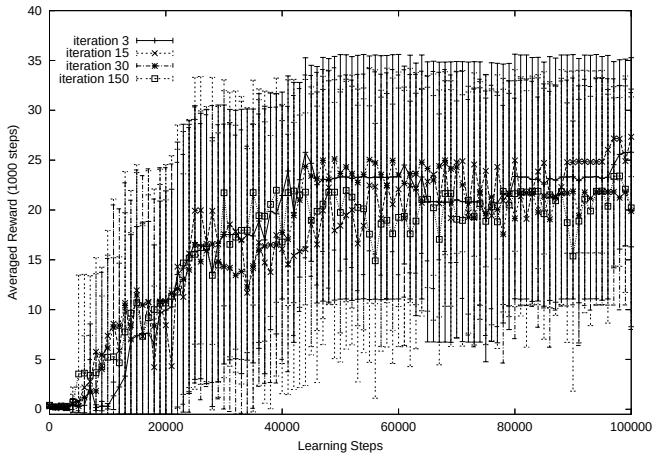


Fig. 6 Learning results averaged over 10 trials in the rod-in-maze problem. The vertical axis gives the averaged reward over making 1000 decisions (steps). The iteration numbers in Gibbs sampling are 3, 15, 30 and 150.

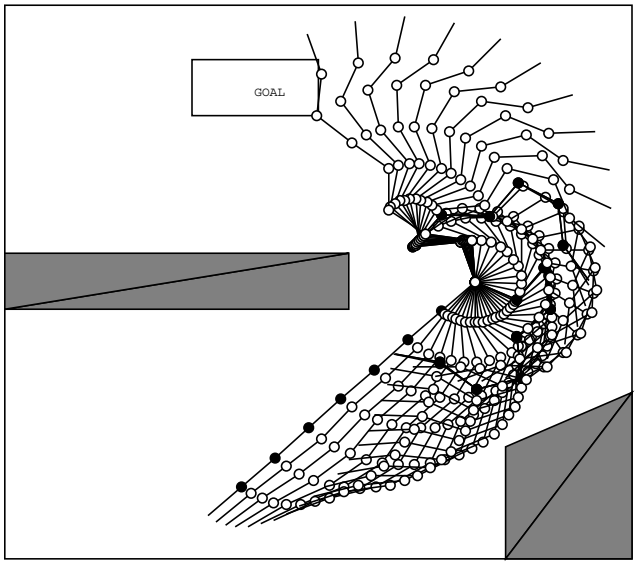


Fig. 7 A learned example behavior of a reaching task using redundant-arm that have eight-joints. The conditions that the joints are shown by black circles are the positions at which the learner makes decisions.

Fig.5 に学習結果（10 試行平均）を示す．この問題はスタートから 3 ステップでゴールする解が最良解だが，試行によっては提案手法は 4〜5 ステップを要する解しか見つからない場合がある．これは試行毎にランダムにタイルを生成しているため，タイルの配置によっては 3 ステップでゴールする解が見つからないためではないかと考えられる．これに対して，空間を均一にグリッド分割した場合，高々$6^3 = 216$ 個程度のタイルに分割した程度では全く解を見つけられないことが分かる．この問題は行動の次元数が 3 であるため，フラット行動選択も不可能ではないが，Gibbs-sampling と比較すると Q 値の計算回数は 600〜700 倍になり，実験に用いた計算機（CPU: Pentium4 1GHz, OS: WindowXP, 開発言語 Java JDK 1.4.2）では行動を 1 回選択するのに 60sec 程度を要するため，比較実験は困難である．また，フラット行動選択ではボルツマン分布の分母の値が大きくなり過ぎて計算に支障をきたすなど実装上の問題も発生した．

Fig.6 に Gibbs サンプリングの反復回数を変えた場合についての学習結果（10 試行平均）を示す．反復回数は 3 回から 150 回までの大きな範囲で変えてみたが，学習結果にはほとんど差は見られなかった．

さらに Fig.7 に示す冗長アームのリーチングタスク[7] に修正を加えた問題へも適用した．アームは 8 本のリンクが関節を介して直列に接続されており，2 次元平面上を動く．状態空間 $s = (\theta_1, \theta_2, \cdots \theta_8)$ は 8 次元で，θ_1 は根元のリンクが水平面となす角度で，$i > 1$ のとき θ_i は関節 i と $i-1$ との間の角度である．アームを構成するリンクは，全て $0 \leq x \leq 1, 0 \leq y \leq 1$ の矩形範囲になければならない．行動 $a = (\theta_{1d}, \theta_{2d}, \cdots \theta_{8d})$ も 8 次元で，目標とする関節角度を表す．行動により目標角度が与えられると，その方向へ $s = (\theta_1, \cdots \theta_8)$ 空間中を直線的に移動し，途中で障害物あるいはゴール領域に触れるか，目標座標に到達すると，イベントが発生して次の意思決定を行う．アームの台座の位置は $(0.75, 0.5)$，アーム全長 0.6，ゴール領域 $0.3 < x < 0.5,\ 0.8 < y < 0.9$，スタート状態 $-180 + 45$ 度，障害物 1: $(0.8, 0.0), (0.8, 0.2), (1.0, 0.3), (1.0, 0.0)$，障害物 2: $(0.0, 0.45), (0.55, 0.45), (0.55, 0.55), (0.0, 0.55)$ である．強化学習エージェントは，8 次元の状態入力の各次元を 10 分割して離散化する．8 次元の行動 $(\theta_{1d}, \cdots \theta_{8d})$ についても同様に各次元を 10 分割して離散化する．よって離散空間の強化学習問題としては状態数 10^8，行動数 10^8，状態-行動空間は 10^{16} という膨大な空間になる．この空間に対し，ランダムタイルを 200 個生成して特徴ベクトル $f_j(s), g_k(a)$ とする．ランダムタイルは，各次元についてタイル矩形領域の部分空間を構成する要素として選択する確率を 0.3 としてランダムに作成する．Gibbs サンプリングの反復回数は 40 回で，他の設定は Rod in maze 問題と同じである．比較対象として，状態と行動空間を $2^8 = 256$ 個のタイルへ均一にグリッド分割した場合について調べる．

Fig.8 に学習の様子（10 試行平均）を示す．この問題はスタートから 3 ステップでゴールする解が最良解であるが，提案手法では試行毎にランダムにタイルを生成しているにもかかわらず，全ての試行で最良解が見つかっている．しかし，均一なグリッド分割ではタイル数が多いにもかかわらず全く学習できなかった．

Fig.9 に Gibbs サンプリングの反復回数を変えた場合についての学習結果（10 試行平均）を示す．反復回数は 40 回から 400 回までの大きな範囲で変えてみたが，学習結果にはほとんど差は見られなかった．

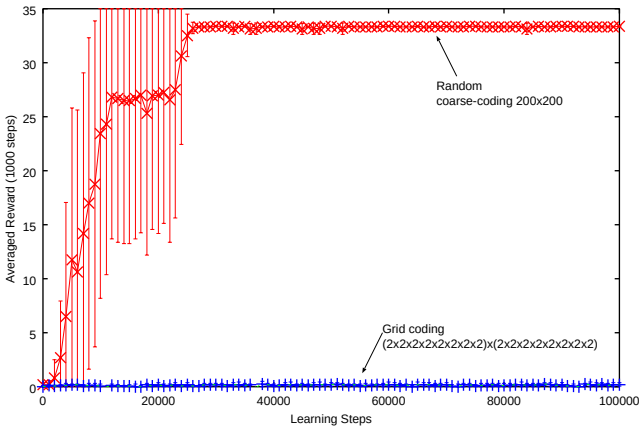


Fig. 8 Learning results averaged over 10 trials in the redundant-arm reaching task. The vertical axis gives the averaged reward over making 1000 decisions (steps).

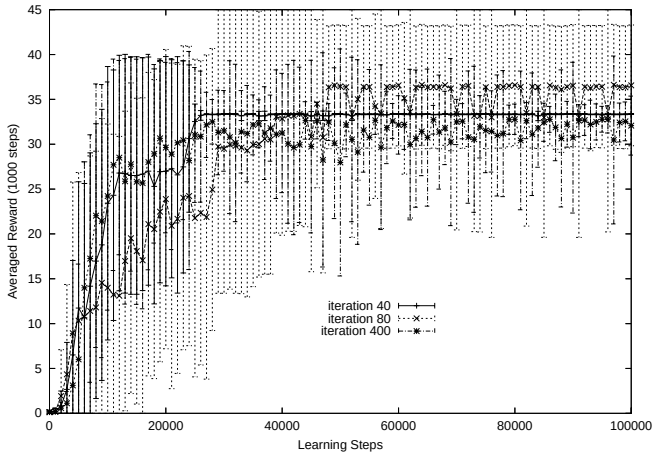


Fig. 9 Learning results averaged over 10 trials in the redundant-arm reaching task. The vertical axis gives the averaged reward over making 1000 decisions (steps). The iteration numbers in Gibbs sampling are 40, 80 and 400.

6. 考察

6.1 ランダムタイルの生成パターンと学習性能

実験ではいくつかの種類のランダムタイリングを用いたが，ランダムタイルの生成パターンによって，学習性能が大きく変わることがあった．特に Rod in maze 問題でその傾向が顕著に見られた．また，タスクの学習に必要なタイル数については，高い次元の問題ほど多くのタイルを必要とするとは限らないことが冗長アームの実験結果から考察される．冗長アームの問題では，アームの移動コストを考えず，意思決定ステップ数だけで評価しているため，最適なパスが無数に存在するために，200 個という比較的少ないタイルでも最適解を容易に発見できたと考えられる．しかし，同じような分割数でも均一にグリッド分割した場合には，どちらの問題においても全く学習できない．ランダムタイリングによって生じる細かい分割と粗い分割や，全ての次元の変数を考慮するのではなく一部の次元の変数に注目した汎化が重要と考えられる．線形関数近似を用いた Value 関数の強化学習では，学習が収束するための条件の一つとして特徴ベクトルが状態間で線形独立であることが示されている [10]．一般的な格子分割による離散化や，ラジアル基底関数による関数近似は，ほぼ全ての領域で単位ベクトルに近い線形独立な特徴ベクトルが生成されることが保障されるが，特徴ベクトルが単位ベクトルであることは収束の条件としては必要ではない．また，例題として取り上げているタスクをみれば明らかなように，連続な状態-行動空間におけるタスクは，本質的には複雑さが小さく，区別が必要な状態や行動の領域はあまり多くないと考えられる．本論文で提案しているランダムな矩形タイルを多数用いる方法は，上記のような本質的に複雑さの小さい連続な状態-行動空間におけるタスクにおいて，区別が必要な状態や行動の領域において線形独立な特徴量ベクトルを生成しやすいと考えられる．つまり，提案手法では特徴量ベクトルが単位ベクトルに制限されていないため，タスクに無関係にランダムに配置されたタイルでも，線形独立な特徴量ベクトルになりやすく，適切な関数近似が行えるのではないかと考えられる．

Fig.10 は冗長アームの問題において得られた初期状態における Q 関数の形状を示す．1 番目と 5 番目の行動要素の値だけを変化させ，それ以外の行動要素の値を一定にしている．ランダムタイルを用いた場合は，同一の特徴量を生成する均一なグリッド分割よりもよりなめらかな形状をしており，適切な関数近似が行えていることが観察できる．

実験において，状態空間に対するランダムタイルの中には，全く使用されていないものが 2〜3 割存在した．これは，定義された状態領域の中には到達不可能な領域があるなどの事情による．本論文では固定されたランダムタイリングを用いているため上記のような問題が生じているが，新しい未知の状態入力があるたびに，それに対応して適応的に新しいランダムタイルを生成するなどの工夫が考えられる．また，学習初期は大きな領域を持つランダムタイルによって学習し，学習進行に応じて重要と思われる領域へ適応的に小さめのタイルを追加していくことにより，関数近似の精度を上げていくことも考えられる．また，タイルを自由に追加できる点は，エキスパートが動作を教える場合において特定の領域に新しくタイルを追加してより細やかな動作をさせることが容易であるという利点がある．

6.2 Gibbs サンプリングにおける反復回数の決め方について

反復回数の理論的な下限については明らかにはされていないため，実験的に決めるしかないのが実情であり，扱う問題によってまちまちであるが，反復回数が多ければ問題は生じない．実験では極端に少ない反復回数の場合についても試したが，性能に対する影響は観察できなかった．これはランダ

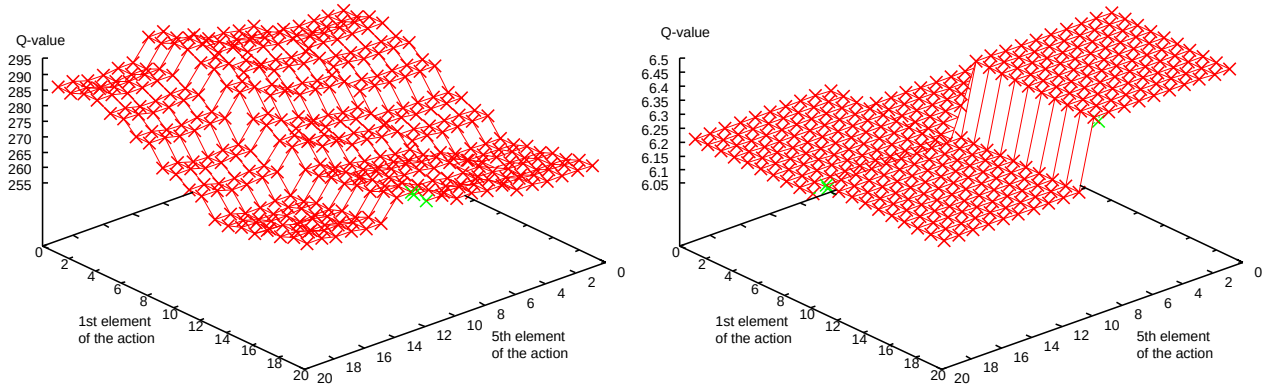


Fig. 10 Lefthand side: Landscape of the learned Q-function using Random-tile coding in the redundant-arm reaching task. The axises are the 1st and 5th element of the action in the initial state. Righthand side: Landscape of the learned Q-function using the grid-coding in the same condition.

ムタイルが矩形で，個数も 200 個程度と少ないため分布関数の形状がかなり単純なためではないかと考えられる．Gibbs サンプリングの反復回数は，エージェントの 1 回の行動選択に要する時間に影響するものであり，時間が許容する範囲内で反復回数をなるべく大きく設定するのが最も単純である．少ない反復回数で近似精度の良いサンプルを得るのは確率サンプリングにおける一般的問題であり，反復過程でアニーリングを行う方法や Neal の Ordered OverRelaxation 法[3] などがある．

6.3 ボルツマン選択における温度パラメータの設定

温度パラメータ T の設定は，扱う問題の行動次元が変わると，好ましい設定値が大幅に変わる傾向がある．これは扱う問題毎に設定が必要であり，かつ調節が困難な問題である．また，Gibbs サンプリングをそのまま実行する場合，$T \to 0$ の極限に近づくと，理論的なサンプルと Gibbs サンンプリングによって得る値とのギャップが大きくなったり，計算機での実装において問題が生じたりする．これは，提案手法において Q-learning における更新処理で式 (10) の MaxQ を得る部分が近似的にしか得られない問題にも通じる．

7. おわりに

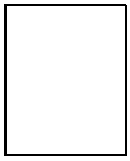
本論文では，高次元の状態-行動空間における有望な強化学習方法として，ランダムタイリングによる状態-行動空間の汎化と Gibbs サンプリングによる行動選択法を組合せて Q-learning へ適用する方法を提案した．ランダムタイリングによる汎化および Gibbs サンプリングによる行動選択方法は，どちらも計算処理が軽く，高次元空間における強化学習においてリアルタイムの意思決定に向いた方法である．これらはそれぞれ単独でも実装可能であるが，両方を組み合わせたときに最も効果が出ることを Rod in maze 問題および冗長自由度アームのリーチング問題へ適用して示した．今後の課題として，状況に応じ適応的にタイルを追加・削除する方法の検討や，Gibbs サンプリングを効率良く行う方法の検討，および温度パラメータの設定方法の確立がある．

参 考 文 献

1) 深尾 隆則，稲山 典克，足立 紀彦：正則化理論を用いた連続的状態と行動を扱う強化学習，システム制御情報学会論文誌, Vol.11, No.11, pp.593–599 (1998).
2) 堀内 匡，藤野 昭典，片井 修，椹木 哲夫：連続値入出力を扱うファジィ内挿型 Q-learning の提案，計測自動制御学会論文集, Vol.35, No.2, pp.271–279 (1999).
3) Jordan, M. I.: Learning in Graphical Models, The MIT Press, (1999).
4) 木村 元，宮崎 和光，小林 重信：強化学習システムの設計指針，計測と制御, Vol.38, No.10, pp.618–623 (1999).
5) 木村 元：強化学習における高次元数の行動空間の扱いについて—ハッシュと Gibbs-Sampling を用いた行動選択方法の提案-，計測自動制御学会 第 32 回知能システムシンポジウム, pp.399–404 (2005).
6) 木村 元：ランダムタイリングを用いた多次元状態-行動の強化学習，計測自動制御学会 システム・情報部門学術講演会 2005 講演論文集, pp.37–42 (2005).
7) Moore A.W. and Atkeson, C.G.: The Parti-game Algorithm for Variable Resolution Reinforcement Learning in Multidimensional State-spaces, *Machine Learning 21*, pp.199–233 (1995).
8) Santamaria, J. C., Sutton, R. S. & Ram, A.: Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces, *Adaptive Behavior 6 (2)*, pp.163–218 (1998).
9) Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding, *Advances in Neural Information Processing Systems 8 (NIPS8)*, pp.1038–1044 (1996).
10) Sutton, R.S. & Barto, A.: Reinforcement learning: An introduction, *A Bradford Book*, The MIT Press (1998).
11) Watkins, C.J.C.H. & Dayan, P.: Technical Note: *Q*-Learning, *Machine Learning 8*, pp.279–292 (1992).
12) 吉本 潤一郎，銅谷 賢治，石井 信：強化学習の基礎理論と応用，計測と制御, Vol.44, No.5, pp.313–318 (2005).

［著 者 紹 介］

木　村　　元（正会員）



1992 年東京工業大学工学部制御工学科卒業．1997 年同大学大学院総合理工学研究科知能科学専攻博士後期課程修了．1998 年 4 月，同大学大学院総合理工学研究科助手．2004 年 4 月，九州大学大学院工学研究院海洋システム工学部門助教授，現在に至る．人工知能，特に強化学習に関する研究に従事．人工知能学会，日本ロボット学会，日本船舶海洋工学会会員．