多次元状態-行動空間での強化学習
－ランダム矩形タイルによる汎化方法の検討－

Reinforcement Learning in High-dimensional state-action space:
Development of a random rectangular coarse coding

九州大学 大学院工学研究院 海洋システム工学部門　○木村 元
○ Hajime Kimura, Graduate School of Engineering, Kyushu University

Abstract: This paper presents an improvement method for rectangular coarse coding in reinforcement learning. The rectangular coarse coding is very simple and quite promising in high-dimensional continuous domains. However, a number of features are useless because the rectangular features are simply generated at random. In this paper, a criterion for the feature is proposed, and a feature inprovement algorithm is given. The algorithm is demonstrated through a crawling robot learing task, Rod in maze task, and a redundant-arm reaching task.

1　はじめに

強化学習は，未知なる環境でロボットの制御規則を自ら発見していくための枠組みとして有望である．単純かつ理論的に洗練された代表的な強化学習法としてテーブル形式のQ-learning[5]があるが，離散的な状態・行動を対象としており，これをそのまま実問題へ適用することは難しい．強化学習を実問題に適用するには，高次元の状態-行動空間を扱う必要があり，そのためには高次元の空間においてQ値を汎化する問題（関数）だけでなく，高次元の行動空間においてQ値などで計算される確率分布に従って行動を選択する問題も解決しなければならない．著者らは高次元の状態-行動における強化学習法として，ランダムな矩形タイルを多数用いた汎化方法にQ-learningを組合せ，Gibbsサンプリングによる行動選択を行う方法を提案している[2]．この方法は，特徴量をランダムな矩形タイルで与えるというad hocな方法である割には格子状に空間を分割するよりもはるかに高い学習性能を示すが，一方でランダムに生成することから全く使用されない無駄な特徴量が存在するなどの問題点があった．本論文では，上記の方法においてランダムに生成していた矩形特徴量を評価するための方法と，この評価に従ってより良い矩形特徴量を生成する新しいアルゴリズムを提案し，シミュレーション実験により提案手法の評価を示す．

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自然な評価規範として，割引報酬の合計がある．$E\{\cdot\}$は期待値，γは割引率を表すものとすると，MDPでの評価関数は以下のように定義される：

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, s_0 = s, \pi\right] , \qquad (1)$$

MDPにおける学習の目的は，各状態sにおいて式(1)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．

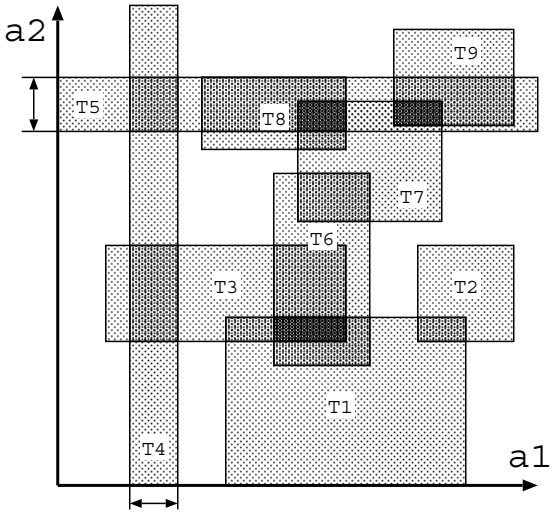

Figure 1: 2次元行動空間での9個のタイルによるランダムタイリングの例．タイルT4は部分空間a1の矢印で示される区間で定義されるタイルのため，それ以外の空間a2では全領域をカバーするタイルになる．タイルT5も同様．

3　ランダム矩形特徴量による空間汎化

多数の矩形ランダムタイルを用いて高次元空間を汎化する方法[3]を説明する．ある1つの矩形ランダムタイル$f(x)$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 $F(s) = (F_1(s), F_2(s), \cdots F_{2T_s})$ および $2T_a$ 個の要素を持つ行動特徴量ベクトル $G(s) = (G_1(s), G_2(s), \cdots G_{2T_a})$ 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。

$$F_i(s) = \begin{cases} f_i(s) & , \text{where } \ i \le T_s \\ 1 - f_{(i-T_s)}(s) & \text{otherwise} \end{cases} \tag{2}$$

$$G_i(s) = \begin{cases} g_i(s) & , \text{where } \ i \le T_a \\ 1 - g_{(i-T_a)}(s) & \text{otherwise} \end{cases} \tag{3}$$

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：

$$Q(s,a) = \frac{1}{T_s\, T_a} \sum_{j=1}^{2T_s} \sum_{k=1}^{2T_a} F_j(s)\, G_k(a)\, w_{jk} \tag{4}$$

式 (4) 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．しかしこの行動選択では，高次元行動空間でそのまま実行しようとすると組合せ爆発を起こすため，Gibbs サンプリングによる行動選択を行うなど工夫が必要である．

4　ランダム矩形特徴量改善法の提案

本章では，ランダムに生成した特徴量を評価する方法と，その評価に基づいてより好ましい特徴量を生成するアルゴリズムを提案する.「好ましい特徴量」について，まず経験したデータに全く反応しない特徴量や，逆に全データに反応してしまう特徴量は，状態判別という目的から考えると無意味なので，評価値として最低値を与えるべきである．よって，特徴量によってデータを判別したときによって得られる「情報量」（またはエントロピー）がその特徴量の評価として妥当である．一方で特徴量は，ある狭い局所的な状況を表して区別することで，その状況特有の行動を学習することを期待されており，注目している特徴量が全状態空間中で占める割合が少ないほうが好ましいとも考えられる．そこで本研究では個々の特徴量 f_i の評価関数 $H(f_i)$ として次の計算式を提案する：

$$H(f_i) = -p(f_i, D)\, \log p(f_i, D)\, \frac{1}{\text{矩形 } f_i \text{ の体積}} \tag{5}$$

1. 一定期間動作したときの状態-行動-報酬のデータを取得し，エピソードデータ D とする．
2. 矩形特徴量をランダムに，ただし重複を避けて生成する．
3. エピソードデータ D を用いて，ランダムに生成した矩形特徴量を個別に評価し，評価値がゼロの矩形特徴量をランダムに，ただし重複を避けて作り直す．評価値がゼロの矩形特徴量が無くなるまで繰り返す．
4. エピソードデータ D を用いて，評価値が最も低い矩形特徴量をランダムに，ただし重複を避けて作り直す．全ての矩形特徴量のうちで最低と評価される評価値が所定回数だけ改善されるまで繰り返す．
5. エピソードデータ D と生成した矩形特徴量集合を用いてオフラインで Q-learning を行う．

Figure 2: ランダム矩形特徴量改善アルゴリズム

ただし $p(f_i, D)$ は入力データの集合 D のもとで，特徴量 f_i が反応して 1 を出力する確率を示す.「矩形 f_i の体積」は，全空間の大きさを 1 としたとき特徴量 f_i が反応して 1 を出力する空間の大きさを体積で表したものである．

個別の特徴量の評価だけでなく，特徴量集合全体を評価することも必要である．このとき，個別の特徴量評価値を単に合計すれば良いわけではない点に注意が必要である．例えば，特徴量集合中にデータに対して全く同じ反応を出力する特徴量が複数存在する場合，状態を区別する目的から考えて無意味になるため，同じ反応を出力する特徴量群に対する評価値としては高々特徴量１つ分とすべきである．そこで本研究では特徴量集合に対する評価関数として次の計算式を提案する：

$$\sum_D \left(\begin{array}{l} D \text{ の各データに反応する全ての} \\ \text{特徴量の評価値のうち，最大の値} \end{array} \right) \tag{6}$$

Fig. 2 に式 (5) の個別特徴量の評価計算値を利用したランダム矩形特徴量改善アルゴリズムを示す．しかしこのアルゴリズムでは，特徴量集合全体に対する評価式 (6) を用いていない．以下の実験において本評価式の妥当性を検証するが，このような特徴量集合に対する評価関数を用いたアルゴリズム構築は今後の課題である．

5　Gibbs サンプリングによる行動選択

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．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

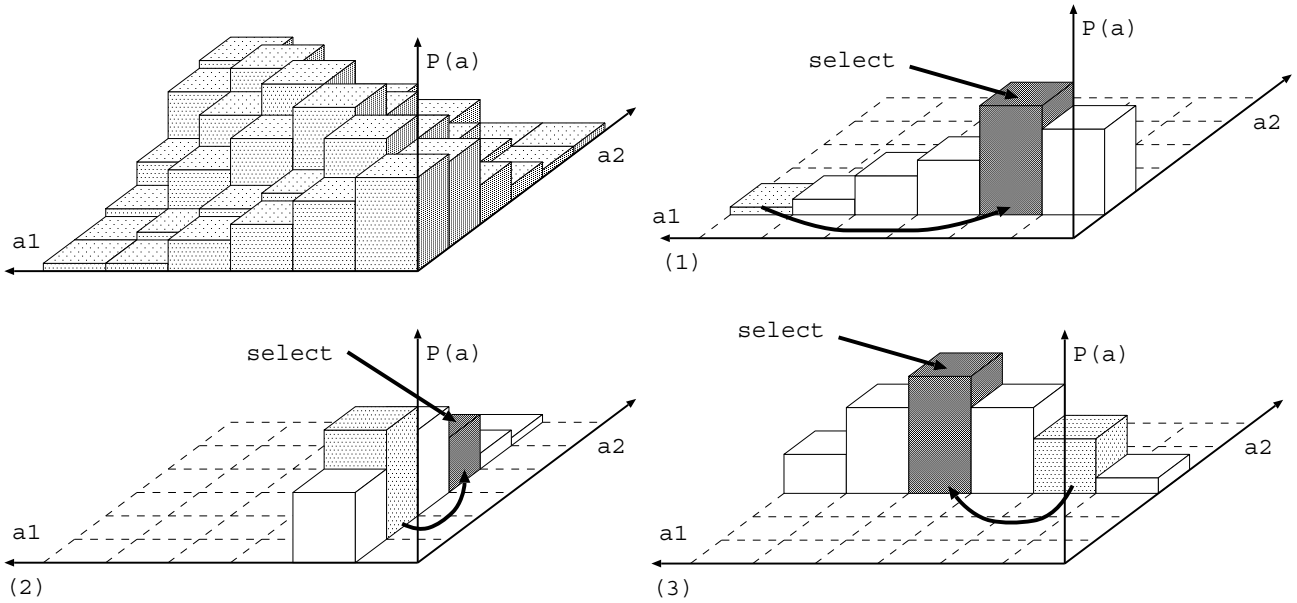

Figure 3: 格子状に分割された 2 次元行動空間を Gibbs サンプリングする様子．左上の図は行動選択確率分布，右上 (1) は $a2$ を固定したときの条件付確率により $a1$ を選ぶ様子，左下 (2) は (1) で選んだ $a1$ を固定した条件付確率により $a2$ を選ぶ様子，右下 (3) は (2) で選んだ $a2$ を固定した条件付確率により再び $a1$ を選ぶ様子．$a1-a2$ は行動空間を表し，たて軸は確率を表す．

$Q(s,a_i)$ に応じた以下のボルツマン分布とする：

$$P(a_i) = \frac{\exp(Q(s,a_i)/T)}{\sum_{j=1}^{D^N}\exp(Q(s,a_j)/T)} \tag{7}$$

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．この行動選択をそのまま実行すると，式 (7) 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 次元程度ならばある程度実用になるが，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，このままでは実行不能に陥る．

著者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した [2]．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 [1]．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式 (7) 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，式 (7) 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．

Gibbs サンプリングでは行動選択確率分布自体は式 7 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 $a^n\in a_d^n$ 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 $P(a_i^n|a^1,a^2,\ldots a^N)$ は，以下のボルツマン分布で与えられる：

$$P(a_i^n|a^1,a^2,\ldots a^N) = \frac{\exp(Q(a^1,a^2,\ldots a_i^n,\ldots a^N|s)/T)}{\sum_{d=1}^{D}\exp(Q(a^1,a^2,\ldots a_d^n,\ldots a^N|s)/T)} \tag{8}$$

フラットな行動選択と比べると，考慮すべき行動が 1 次元ずつになっただけで式 (7) 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．

6　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 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 (2), (3) 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：

$$Q(s,a) = \frac{1}{T_s\, T_a} \sum_{j=1}^{2T_s} \sum_{k=1}^{2T_a} F_j(s)\, G_k(a)\, w_{jk} \qquad (9)$$

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：

$$P(a_i) = \frac{\exp(Q(s,a_i)/T)}{\sum_{j=1}^{D^N} \exp(Q(s,a_j)/T)} \qquad (10)$$

ただし行動は式 (8) 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：

$$Q(s,a_i) \leftarrow Q(s,a_i) + \alpha \left(r + \gamma \max_a Q(s',a) - Q(s,a_i) \right) \qquad (11)$$

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 (9) 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：

$$w_{jk} \leftarrow w_{jk} + \frac{F_j(s)\, F_k(a_i)}{T_s\, T_a} \alpha \left(r + \gamma \max_a Q(s',a) - Q(s,a_i) \right) \qquad (12)$$

ここで $\max_a Q(s',a)$ を探す必要があるが，これは 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．

7　実験

Fig.4 に示す仮想的なほふくロボットに本手法を適用する．学習目標は，ロボットを前進させるために，アームを足のように作用させる動作の獲得である．関節は位置制御のサーボモーターによって角度を制御される．足の本数 K は設定により変えることが可能で，Fig.4 は $K = 4$ の場合を示す．各時間ステップにおいて，エージェントは $2K$ 個の関節モータの角度および K 個の足先のタッチセンサの状態という $3K$ 個の状態量を要素とした $3K$ 次元ベクトルを観測する：関節角度 $\phi_1, \ldots \phi_{2K}$ は $2K$ 次元連続空間で定義され，タッチセンサ $\phi_{2K+1}, \ldots \phi_{3K}$ は 0 または 1 の 2 値である．行動は，関節角度の目標値を指示し，$2K$ 次元ベクトル $(a^1, \ldots a^{2K})$ の各要素がそれぞれ関節角度を表す．

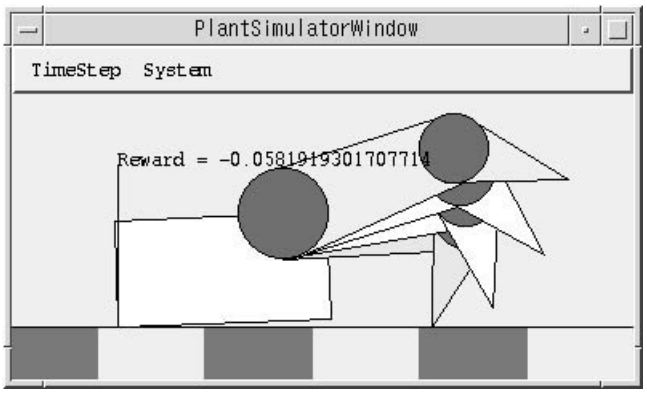

Figure 4: 多自由度ほふくロボットシミュレータ．右側が前方．足は K 本で，各足にはモータ 2 個ずつ計 $2K$ 個取り付けられている．各足の先端にはタッチセンサが付いている．状態は $2K$ 個の関節の角度と K 個のタッチセンサの値の計 $3K$ 次元で，行動は $2K$ 個の関節の目標値で計 $2K$ 次元である．

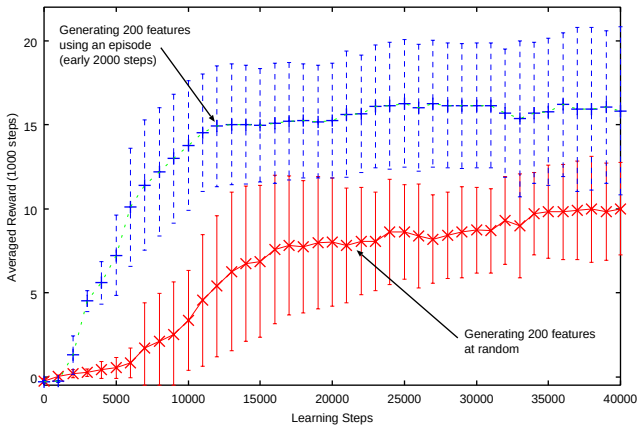

Figure 5: 2 本足ほふくロボット（状態 6 次元，行動 4 次元）での学習の様子．横軸は学習ステップ，縦軸は 1000 ステップ期間の平均報酬を表す．

行動ベクトルの各要素は $[0, 1]$ の範囲に限定される．行動が選ばれると，モータは指示された目標位置へ動きはじめる．関節角度が指示された位置まで動くか，あるいはタッチセンサの値が変化すると，状態遷移の結果として報酬が与えられ，次の時刻へ進む．関節のモータが目標位置まで動く途中でセンサの値が変化すると，そこで意思決定イベントが発生して動きが打ち切られるため，次のステップでの関節角度は行動として出力された目標角度には一致しない．よって状態遷移には不確実性が存在する．報酬は，ボディが前進した距離与えられる．ロボットが後退した場合，報酬の値は負になる．

強化学習エージェントは，関節の角度入力変数は 10 分割，タッチセンサの入力変数は 2 分割して離散化する．行動については各次元を 10 分割して離散化する．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，離散化された空間

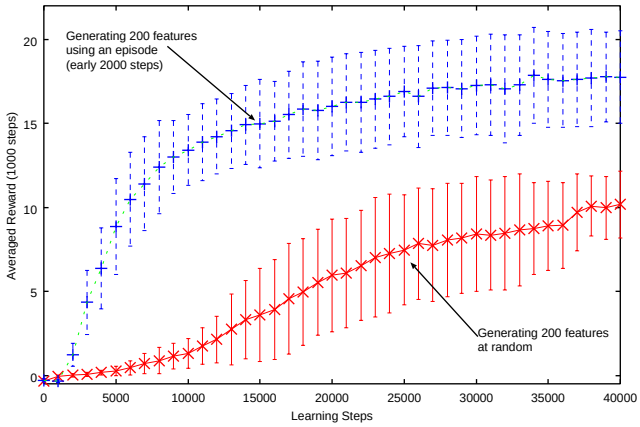

Figure 6: 3 本足ほふくロボット（状態 9 次元，行動 6 次元）での学習の様子．横軸は学習ステップ，縦軸は 1000 ステップ期間の平均報酬を表す．

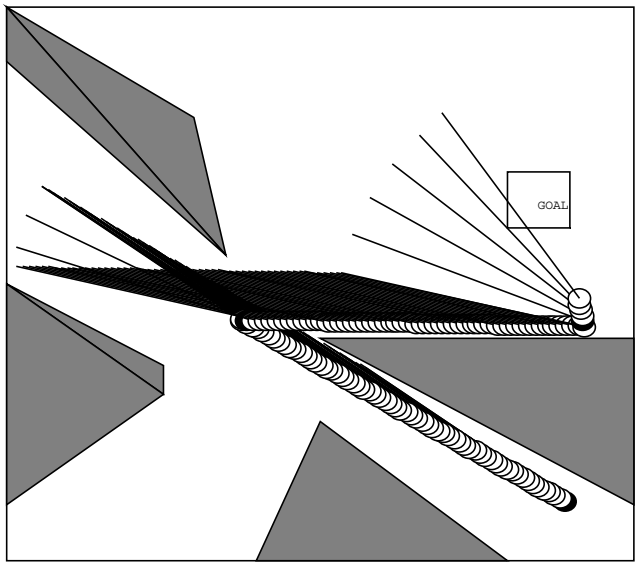

Figure 7: Rod in maze 問題．学習主体は棒下部の円が黒く表示された時点毎に意思決定を行う．

の境界に合わせて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．Gibbs サンプリングの反復回数は 30 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 $\alpha = 0.4$ に設定する．特徴量を生成する提案手法については，最初の 2000 ステップ分について学習せずにランダムに動作した経験をデータとして蓄え，Fig.2 のアルゴリズムによって特徴量を生成した後，2000 ステップ後より環境中での試行錯誤により Q-learning を行う．

Fig.5 に 2 本足，Fig.6 に 3 本足ほふくロボットの学習結果（10 試行平均）を示す．この問題は 2 本足の場合，足を交互に動かすのが最適な動作で，3 本足の場合は各足の動作の位相を 120 度ずつずらすのが最適な動作となる．ランダムな特徴量生成では最適な動作を見つけることは困難だが，提案手法では高い割合で最適な動作を学習できた．

Fig.7 に示す Rod in maze 問題 [4] に修正を加えた問題へ本手法を適用する．状態空間は，棒の下端の円の中心座標および棒が水平面となす角の 3 次元空間で構成される．

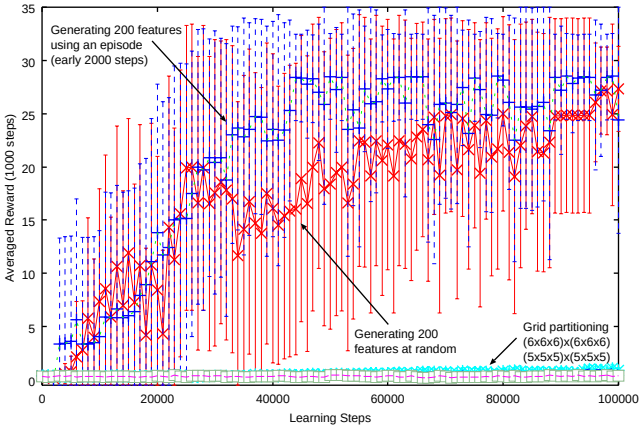

Figure 8: Rod in maze 問題（状態 3 次元，行動 3 次元）における学習の 10 試行の平均と標準偏差を示す．横軸は学習ステップ，縦軸は 1000 ステップ期間の平均報酬．

棒の端は常に円の部分より上にしか存在できない．行動も 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．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へ戻り，それ以外では報酬 0 である．問題設定の詳細は文献 [3] を参照のこと．

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 次元の行動についても同様に各次元を 100 分割して離散化する．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．行動のランダムタイル $g_k(a)$ も同様に 200 個生成する．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．参考までに，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も表示した．その他の設定についてはほふくロボットの実験と同じである．

Fig.8 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，どの手法でも 4〜5 ステップを要する解しか見つからない場合がある．この問題の場合，提案手法とランダムな特徴量生成とでは有意な差は見られない．これに対して，空間を均一にグリッド分割した場合，全く解を見つけられない．

さらに Fig.9 に示す冗長アームのリーチングタスク [4] 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．行動も 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．問題設定の詳細は文献 [3] を参照のこと．

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 次元の行動についても同様に各次元を 10 分割して離散化する．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

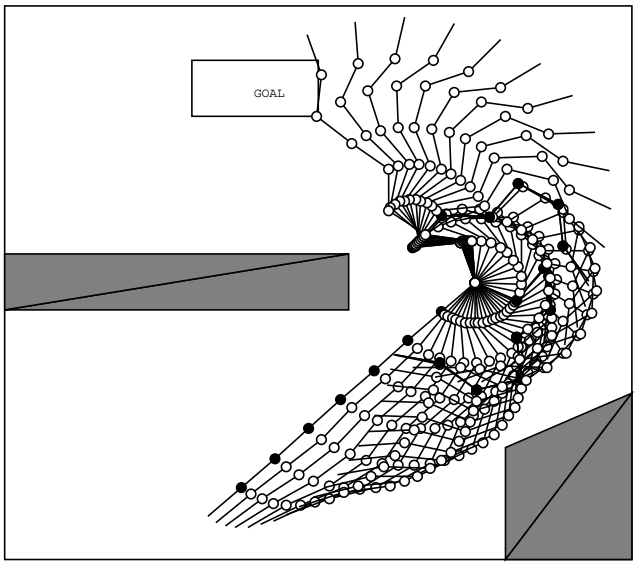

Figure 9: 8 関節冗長アームのリーチングタスク．学習主体は関節が黒丸で示された時点毎に意思決定を行う．

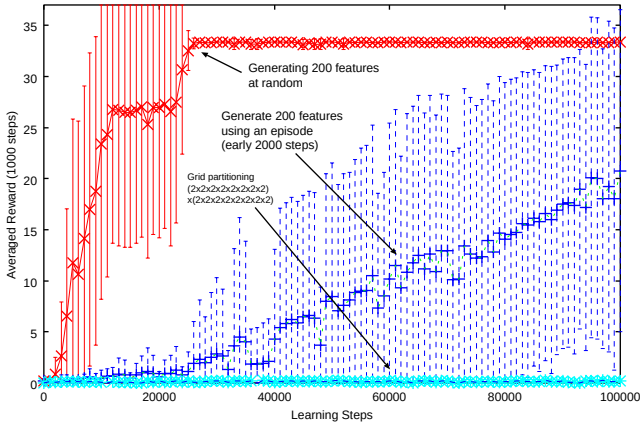

Figure 10: 8 関節冗長アームのリーチングタスク（状態 8 次元，行動 8 次元）での学習の様子．横軸は学習ステップ，縦軸は 1000 ステップ期間の平均報酬を表す．

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，他の設定はほふくロボットや Rod in maze 問題と同じである．参考までに，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挙げた．

Fig.10 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，ランダムにタイルを生成する手法では全ての試行で最良解が見つかっているのに対し，提案手法では大幅に学習性能が悪化している．

8 考察

提案手法の効果は，多足ほふくロボットの問題で顕著に見られたことから，個別の特徴量を評価する方法（式 (5)）とそれに基づいて改善するアルゴリズムは有効であると考えられる．しかし，Rod in Maze 問題では性能向上はあまり見られず，冗長アームの到達問題では大幅に悪化している．この原因は，特徴量の生成を行うときに使用するエピソードデータが学習初期のランダム動作だけであることが考えられる．つまり，多足ほふくロボットでは学習初期のランダム動作によって最適政策やそれに近い政策を表現するのに十分な状態を訪問することが可能なのに対し，Rod in Maze 問題や冗長アームの問題では，ランダム動作によりスタート付近の状態を訪問できても，なかなかその先のゴールまで進めず，ゴール付近での制御規則の表現に必要な特徴量を生成できなかったと考えられる．このような問題では，強化学習の進行と共に特徴量を追加・削除していくことが必要だろう．

また，個別特徴量評価式 (5) はある程度有効と考えられるものの，データの量などによって値がかなり異なってくるため，MDL などデータ量を考慮に入れた規範を参考にして適切な修正を行うことが必要と思われる．

さらに，特徴量集合の評価式 (6) は，この式による評価値と強化学習後に得た政策の性能との相関が負になった．例えば 2 足ほふくロボットの場合，評価値と 40000 step 学習後の平均報酬の相関係数 $r = -0.71$ で，3 足ほふくロボットの場合でも相関係数 $r = -0.53$ であり，有効な評価式ではないと考えられる．Fig.2 に示した提案手法は，式 (6) を用いていないため，本手法の有用性を否定するものではないが，今後アルゴリズムを改良する上で特徴量集合の評価式を適切に定式化することが必要である．

9 おわりに

高次元の状態-行動空間における強化学習として有望な矩形ランダム特徴量による汎化と Gibbs サンプリングによる行動選択法を用いた Q-learning を取り上げ，矩形ランダム特徴量の評価方法および改良アルゴリズムを提案した．本手法を多足ほふくロボット学習問題，Rod in maze 問題および冗長アームのリーチング問題へ適用し，有用性や限界について考察した．

References

[1] Jordan, M. I.: Learning in Graphical Models, The MIT Press, (1999).

[2] 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).

[3] 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).

[4] 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).

[5] Sutton, R.S. & Barto, A.: Reinforcement learning: An introduction, *A Bradford Book*, The MIT Press (1998).