

特徴構成法を用いた Q 学習の効率改善

宮 本 行 庸[†] 上 原 邦 昭^{†,††}

本稿では、特徴構成法を用いた強化学習システム FCQL について述べる。従来の強化学習では、対象とする環境の各状態を識別する適切な属性が、学習の前段階であらかじめ準備されていることを仮定している。現実には、学習システムが状態を識別するのに十分な入力系を持っているとは限らず、領域に固有の特徴を適宜構成していく機能が必要とされる。本稿では、構成的帰納学習に用いられる特徴構成法を、強化学習の一手法である Q 学習と統合し、有限離散時間環境における適切な内部表現と評価関数を学習する手法を提案する。結果として、単位時間における期待報酬値を最大化するのみでなく、収束までに費やす状態数の大幅な削減が実現できた。

Improving the Effectiveness of Q -Learning by Using Feature Construction

YUKINOBU MIYAMOTO[†] and KUNIAKI UEHARA^{†,††}

In this paper, we describe a new reinforcement learning system called FCQL (Feature Constructive Q -Learning). Usually, reinforcement learning methods assume that they can identify each state before learning. In a real-world domain, the learner only has limited sensors, so is required the ability to construct new features. This paper describes an approach integrating feature construction with Q -learning to learn efficient internal state representation and a decision policy simultaneously in a finite, deterministic environment. The result shows that FCQL can not only maximize the long-term discounted reward per unit time, but also reduce the number of status to converge.

1. はじめに

近年、実環境における移動ロボットのように、未知なる環境下での学習機能を持ったシステムに関心が高まりつつある。このような環境下では、学習の対象やタスク、あるいは環境との相互作用によって環境に変化が生じ、それらの変化に対するシステムの即応性が求められる。このようなシステムの設計において、動物の学習形態を模倣した強化学習¹²⁾が注目されている。強化学習は、教師つき学習とは異なり、学習者自らが試行錯誤を繰り返しながら、次第に行動を洗練していく学習手法である。強化学習の中でも、特に注目されている手法の一つに Q 学習¹³⁾¹⁴⁾があげられる。

Q 学習は、有限離散 Markov 環境下において、十分な試行の後では最適解への収束が保証されている一

方、学習完了までに必要とする状態が膨大になるという欠点を持っている。これは、 Q 学習が過去の状態を参照する際に完全一致を要求するため、環境の全状態を探索するまで学習が収束しないことが原因である。また、 Q 学習が収束するための条件として、状態を識別するのに十分な属性が与えられなければならないという仮定がある。このため、最初に決定する属性を慎重に選ぶ必要がある。逆に、状態を記述する属性を詳細にとりすぎると、状態の一致条件が厳しくなり、さらに収束が遅れるという問題が生じる。このような問題の解決策として、得られた状態から新たな特徴を作り出し、それらの特徴を用いて状態を評価する機能を Q 学習に持たせることが考えられる。

得られた状態を分割し、状態空間を構成する手法に、以下のような研究がある。G algorithm³⁾では、状態を記述する属性をノードとして、木構造による Q 関数の実現で状態空間を分割しているが、与えられた属性をそのまま用いて分割を行うため、この手法では DNF 問題を解くことができない。浅田²⁾らによる状態空間の自律的構成法では、ゴールまでの行動数に基づいて状態の集合を構成しているが、時間的な抽象化

[†] 神戸大学工学部情報知能工学科

Department of Computer and Systems Engineering,
Kobe University

^{††} 神戸大学都市安全研究センター

Research Center for Urban Safety and Security, Kobe
University

を行っており、状態に基づく空間的な分類は行われていない。前者は与えられた属性による状態空間の分割、後者は得られた状態を時間別に分類して状態空間の再構成を試みる手法であるが、いずれの手法も得られた状態から対象領域の特徴を構成するものではない。また、EOPs⁵⁾では、クラスを識別する関数を定義して分類を行っているが、この手法もまた得られた状態を分類するにとどまり、同じクラスに属する状態から空間的な特徴を構成するものではない。

構成的帰納学習¹⁾¹⁰⁾で提案された特徴構成法は、対象領域に適切な特徴を新たに作り出し、状態の記述を更新する手法である。特徴構成法は、分類などの概念学習において蓄積される状態数の削減と学習精度の向上などの成果が報告されている。本稿では、特徴構成の機能を持つ強化学習システム FCQL (Feature Constructive Q-Learning) を提案する。FCQL は Q 学習の枠組に特徴構成法を統合したシステムで、報酬を用いて状態を分類し、各クラスごとに共通の空間的な特徴を構成して、最適解への収束に必要な状態数の削減と、評価関数の早期収束を達成している。

2. 特徴構成法

2.1 構成的帰納学習における特徴構成法

対象とする領域の状態集合を分類し、同じクラスに属する状態に共通の概念を獲得する手法に、帰納学習がある。一般的な帰納学習は、概念記述があらかじめ与えられた属性から選ばれるために、選択的帰納学習とも呼ばれる。選択的帰納学習で学習できない問題の例に DNF 問題がある。DNF 問題とは、学習したい概念が DNF (選言標準形: Disjunctive Normal Form) で記述される問題領域である。目標概念となる DNF は、属性の連言の選言形式で記述される。

DNF 問題の代表例に三目並べ問題 (図 1) がある。この問題は、「三目並べの終了時における X の勝ち」という概念を学習する問題である。以下では、学習したい概念に含まれる例を正例、それ以外を負例と呼ぶ。

ここで、正例を記述する概念が $(x_1=X)$ であると仮定して、選択的帰納学習を用いて正例と負例への分割

を考える。このような記述は正例中にも負例中にも一様に存在しているため、学習は失敗する。これは、三目並べ問題で学習したい概念が $(x_1=X) \wedge (x_2=X) \wedge (x_3=X) \vee (x_4=X) \wedge (x_5=X) \wedge (x_6=X) \vee \dots$ のような DNF で記述され、単項の属性のみでは分類できないことが原因である。このように、DNF 問題のような複雑な問題を学習するには多くの例を保持しておかなければならず、例からの学習を行う Q 学習において非効率な問題の代表例であると言える⁶⁾。

構成的帰納学習では、得られた状態から新たに特徴を構成し、その特徴が一定の基準を満たせば選択される。つまり、構成的帰納学習には、選択的帰納学習の機能に加えて、「特徴の構成」と「特徴の選択」の機能が追加されている。一般に、構成的帰納学習における特徴とは、対象領域の概念を表現するための要素を指す。特徴の構成は、学習した概念に矛盾が発生した場合☆、選択できる概念の候補がなくなった場合などに行われる。特徴を構成するためには操作子が必要となるが、与えられた属性から特徴を構成する操作子として、数値属性に対しては比較、統合、細分化など、非数値属性に対しては分割、統合、論理演算などがあげられ、三目並べの例では論理演算子の一つである論理積を用いている。

特徴構成法において議論される点は、大きく分けて二つある。まず一つ目は、特徴構成の手続きが組み込み型か前処理型かという点である。組み込み型は、学習アルゴリズム本体に特徴構成法が組み込まれており、学習中に特徴構成を逐次行うことができるが、各アルゴリズムごとに特化した実装をする必要がある。一方、前処理型は学習を行う前に特徴構成を行うため、他の学習アルゴリズムのフィルタとして用いることができるが、学習中には特徴を構成できないことが欠点である。

もう一つは、特徴構成が状態に基づくか知識に基づくかという点である。状態に基づく手法は、特徴構成の際にあらかじめ領域依存な知識を必要としないため、汎用性の高い手法であるが、構成された特徴の精度は低い。一方、知識に基づく手法は、領域に固有の特徴が適切な形で構成されやすいが、知識が必要になるため、汎用性に欠ける。

代表的な構成的帰納学習システムに IB3-CI¹⁾、GALA⁴⁾ などがある。IB3-CI では、帰納学習システム IB3 に特徴構成法 STAGGER を組み込み、分類精度の向上を達成している。STAGGER は組み込

x_1	x_2	x_3	X	X	X	O	X
x_4	x_5	x_6	O			O	X
x_7	x_8	x_9		O		O	X
Each attributes			Positive instance			Negative instance	

図 1 三目並べ問題

Fig. 1 Tic-Tac-Toe problem.

☆ 正例ばかりで過剰一般化となる場合など。

み型であるが、評価は静的な状態集合を対象としているので、逐次的に得られる状態に基づいて構成された特徴には、冗長なものが多く含まれる。また、特徴構成のための知識を与えることも可能であるが、IB3-CIは知識に基づく特徴構成のみでしかよい学習結果は得られていない。

一方、GALAは状態に基づく前処理型の特徴構成法で、多くの帰納学習アルゴリズムのフィルタとして適用できる。GALAは特徴選択の評価関数として利得比基準⁹⁾を用いており、精度の高い選択ができる。しかしながら、前処理型であるため、対象領域の状態がすべて既知である必要があり、学習中には特徴を更新できないことが問題となっている。

Q学習では逐次的に状態に遭遇するため、特徴は学習中に適宜更新されることが望ましい。このため、FCQLでは特徴構成法をアルゴリズム中に組み込んでいる。さらに、特徴に基づいた行動決定と、特徴の評価値の更新のために、FCQLでは各特徴と行動の対に評価値を与える関数を新たに定義している。本稿では、Q学習では非効率な問題の例としてDNF問題を取り上げる。また、特徴を構成する操作子に論理積を採用し、以下では属性一値対の連言のことを特徴と呼ぶ。

2.2 FCQLにおける特徴構成法

2.2.1 特徴の構成

帰納学習では、与えられる状態は複数の属性一値対と、その状態の属するクラスで記述されている。FCQLでは、このクラスの相当する部分を、状態が得られた直後の報酬を用いて定義している。具体的には、時刻 t で得られた状態を s_t 、報酬を r_t とすると、 s_t は複数の属性一値対で記述されている部分で、 r_t がクラスに相当する部分となる。各クラスごとの報酬値が既知であるとする、 s_t は報酬値が r_t に最も近いクラスに分類される。

また、FCQLで扱う特徴は、属性一値対の連言で記述され、2値をとる。たとえば、2.1節の三目並べの例では、状態 s_t に対し、特徴 $f_1 = (x_1=X) \wedge (x_2=X)$ = Trueとなることは、 s_t に $(x_1=X)$ と $(x_2=X)$ という属性一値対が含まれていることを意味している。

FCQLの特徴構成では、まず、得られた状態の属性一値対について、すべての2項連言が構成される。このとき、すでに構成された特徴が存在すれば、それらの特徴と属性との連言についても、可能な組合せがすべて構成される。三目並べの例では、すでに特徴 f_1 が存在しているとき、特徴 f_2

$$\begin{aligned} f_2 &= (f_1 = \text{True}) \wedge (x_3 = X) \\ &= (x_1 = X) \wedge (x_2 = X) \wedge (x_3 = X) \end{aligned}$$

という組み合わせも可能となり、属性の3項以上の連言も特徴の候補として構成される。状態を記述する属性数を n とすると、特徴を構成する項数は最大で n となる。また、一つの状態から構成される特徴数は、最大で $\sum_{i=1}^n n C_i$ となり、全特徴数はこの状態数倍が上限となる。

2.2.2 特徴の選択

構成された特徴の候補は、何の評価も受けていないため、その中から適切な特徴を選択する必要がある。たとえば、2.2.1項の三目並べの特徴 f_1 を用いて、特徴 f_3

$$\begin{aligned} f_3 &= (f_1 = \text{True}) \wedge (x_4 = O) \\ &= (x_1 = X) \wedge (x_2 = X) \wedge (x_4 = O) \end{aligned}$$

といった特徴も構成できるが、この特徴は三目並べにおける正例の条件 $\star$ を満たしていないので、学習には不適切な特徴である。このような候補すべてを採用すると特徴が増えすぎてしまうため、その中から適切な特徴を選択しなければならない。FCQLでは、特徴の採用に評価指標を設け、特徴の候補の中から最も適切な特徴を選択して解決している。特徴選択のための評価指標として、FCQLではGALAに導入された利得比基準を採用する。

利得比基準は、ある状態集合を分割するときに、利得比 (Gain Ratio) が最大の属性を選択する基準である。たとえば、状態集合 T を部分集合 $T_1, T_2, \dots, T_n$ に分割する n 値の属性 X と、いくつかのクラス C_i を考える。 T 中の状態数を $|T|$ 、 T 中で C_i に属する状態数を $\text{freq}(C_i, T)$ と表すと、 k 個のクラスに関する平均情報量 (Info) は、 T 内のある状態が属するクラスを同定するのに必要な情報量の平均値を表し、

$$\begin{aligned} \text{Info}(T) &= - \sum_{i=1}^k \frac{\text{freq}(C_i, T)}{|T|} \\ &\quad \times \log_2 \left(\frac{\text{freq}(C_i, T)}{|T|} \right) \end{aligned} \quad (1)$$

となる。 X によって T が n 通りに分割された後、クラスを同定するのに必要な情報量の期待値 (Info_X) は、

$$\text{Info}_X(T) = \sum_{i=1}^n \frac{|T_i|}{|T|} \times \text{Info}(T_i) \quad (2)$$

となる。これらの差である利得 (Gain)

$$\text{Gain}(X) = \text{Info}(T) - \text{Info}_X(T) \quad (3)$$

$\star X$ が一列に並んでいる状態。

は, X で T を分割したとき, クラス分けに役立つ部分の情報量を表している. 一方, 分割情報量 (*Split Info*) は, X による分割自体の情報量を表し,

$$\text{Split Info}(X) = - \sum_{i=1}^n \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right) \quad (4)$$

となる. したがって, 利得比

$$\text{Gain Ratio}(X) = \frac{\text{Gain}(X)}{\text{Split Info}(X)} \quad (5)$$

は, 分割によって得られる情報量のうち, クラス分類に役立つ部分の割合を表している.

利得比基準は, 分類すべきクラス数が状態数と比べて十分に小さいときにも, 分類に役立つ特徴を分割された部分集合の情報量に基づいて定量的な評価ができるという利点がある. また, 離散属性では属性ごとにとり得る属性値の数の隔たりが生じやすいが, この場合でも利得比基準は特定の属性を不利に扱うことがない. さらに, 分割後の部分集合の大きさが不均等な場合にも安定した結果が得られるという性質がある. したがって, FCQL が対象とする問題領域の状態は離散属性で記述されており, 特徴のとり値が 2 値であるために不均等な分割になりやすいが, 利得比基準はこのような問題領域に頑健であるため, FCQL への導入に適していると考えられる.

FCQL では, 構成された特徴の候補の中で利得比が最大の特徴を 1 つ選択し, 新しい特徴としている. これは, 利得比が大きな特徴は, その特徴を用いて状態集合を分割したときに, 同じクラスの状態が集まりやすいことを意味している. 利得比が同じ特徴が複数個存在する場合は, 連言を構成する項数が少ない特徴を選択している. 最後に, この特徴がすでに採用された特徴と重複しなければ, 新たな特徴として採用する. 採用された特徴は特徴を記憶しておく領域に追加され, この領域を特徴集合と呼ぶ. 以上の手続きで採用された, 特徴集合中にある j 番目の特徴 f_j に対し, 状態 s_t において $f_j = \text{True}$ となるとき, 特徴 f_j は状態 s_t に含まれるといい,

$$f_j \in s_t \quad (6)$$

と表記される. このときに採用された特徴を f_j , 特徴構成が行われる直前にとった行動を a_t , 得られた報酬を r_t とすると, f_j と a_t の対に対して評価値を与える関数 F を特徴関数と呼び, $F(f_j, a_t)$ の初期値を

$$F(f_j, a_t) = \begin{cases} r_t & (i = t) \\ 0 & (i \neq t) \end{cases} \quad (7)$$

と定義する. 特徴関数は, 特徴の評価値の更新, および特徴の淘汰に用いられる.

2.2.3 特徴関数の更新

2.2.2 項の手法で得られた特徴を追加していくのみでは, 特徴が増えすぎてしまう恐れがある. また, 構成の時点ではよいと判断された特徴も, 学習が進むにつれ不適切になっていく可能性もある. 選択された特徴が不適切なとき, それらの特徴に基づいた行動決定は最適であるとは言えないばかりか, 不適切な行動を招く恐れもある. この現象を解消するために, 選択された特徴の評価値を更新し, 不要な特徴を淘汰する手続きが必要となる. FCQL では, この手続きを行う評価基準に特徴関数を用いている. たとえば, ある特徴 f_j の特徴関数の値 $F(f_j, a_t)$ は, 以下の式に基づいて更新される.

$$\begin{aligned} F(f_j, a_t) \leftarrow & F(f_j, a_t) + \alpha(r_t \\ & + \gamma \max_{g \in s_{t+1}, b \in A} F(g, b) \\ & - F(f_j, a_t)) \end{aligned} \quad (8)$$

α は学習率で, $0 < \alpha \leq 1$ なる定数である. α の値が大きいほど, 直前の報酬を重要視するように関数が更新され, $\alpha = 1$ のときには過去の特徴値を一切利用しない. γ は割引率で, $0 \leq \gamma \leq 1$ なる定数である. $\gamma = 0$ の場合は, 現在の報酬のみに着目し, 将来得られる報酬を無視することになる. 逆に, γ の値が大きいほど将来の報酬を重視し, $\gamma = 1$ の場合は時刻 t と $t+1$ で得られる報酬を等価に扱うことになる. つまり, 式 (8) は, 次の時刻で最適と思われる行動を選択したときに得られる報酬の見積もりを一段階だけ割り引いた値と, 直前に得られた報酬の和に, 学習率に従って $F(f_j, a_t)$ の値を近づけることを示している.

構成されたすべての特徴は, 0 でない報酬値が得られた状態から構成されているので, 試行が進むにしたがって F 値は実際に得られる報酬値に近付いていく. 誤って構成された特徴は, 以降の試行で報酬を得られることが少なく, F 値は 0 に近づいていく. この更新手続きの結果, すべての行動について評価値のクラスが 0 になった特徴は特徴集合から削除される.

2.2.4 強化学習から特徴構成へのフィードバック

2.2.3 項の操作で更新された特徴関数を用いて, 特徴構成の際に採択されなかった特徴の評価値の更新に反映させることを考える. 2.2.1 項で構成された特徴のうち, すでに特徴集合中に存在するために棄却された特徴については, 式 (8) に従ってその評価値が更新される. 以上の操作を強化学習側から特徴構成へのフィードバックとし, 強化された特徴関数を更新に用

いた結果、既存の特徴の評価値をより早く正確な値に近づけることができる。

2.2.5 状態集合の大きさに関する考察

利得比基準による特徴の評価は、対象とする状態数の大きさに依存する。充分な大きさの状態集合が得られたとき、利得比基準によって対象領域を表現するのに充分な数の特徴を得ることができる。しかしながら、学習初期においては、経験した状態数が少なく、 Q 表の規模も小さい。利得比基準による評価は小さな状態集合に対しては安定せず⁸⁾、必ずしも正当な評価がなされるとは限らない。不適切に選択された特徴が学習初期において不適切な行動を招き、適切な状態の蓄積による Q 関数拡張の妨げになっていると考えられる。この現象は、充分な量の状態を蓄積しないうちに特徴を構成し始めた点に問題があり、蓄積された状態の量に基づいて特徴の構成を始めるタイミングを考慮する必要がある。

以上のような問題点の解決策として、特徴を構成するのに充分な量の状態を得てから特徴構成を始めることが考えられる。しかしながら、「充分な」量を定量的に示す指標を作ることは困難である。また、充分な量の状態を得るまで待つと、試行全体としての収束速度に影響が出る可能性もある。

FCQL では、最初に特徴構成をしない通常の Q 学習を行い、一定数の状態を蓄積し、その後一度だけ特徴構成を行う手続きに切り替える。このとき、学習アルゴリズムを Q 学習から FCQL に切替えるまでの状態を蓄積する過程をディレイと呼ぶ。

3. 特徴構成法を用いた Q 学習

3.1 対象とする学習領域

FCQL が対象とする学習領域は、外界となる環境が有限離散 Markov 決定過程によってモデル化される。時刻 t におけるシステムの入出力は、以下のように定式化される。

◦ 与えられる入力：

- 状態 s_t
- 報酬 r_t

◦ 決定すべき結果：

- 行動 a_t

状態 s_t は離散属性で記述される。報酬 r_t はあらかじめ定義されたクラスのうち、いずれかのクラスに属するものとする。また、行動 a_t は、あらかじめ定義された行動集合 A のうち、 s_t において実行可能な行動の集合から選択される。選択された a_t を実行し、このとき生じた環境の変化に応じて報酬 r_t が環境から

得られる。学習の目的は、長期にわたる割引報酬和の最大化にあり、単位行動あたりの報酬値で評価される。

また、FCQL が対象とする問題領域は、行動に伴う報酬が即時に得られることを仮定している。このため、即時的な報酬に無関係な状態は学習の対象とされず、即時に報酬が得られる行動からの学習が行われることとなる。このような行動の獲得は反射的行動獲得と呼ばれ、FCQL では反射的行動獲得を行う。

3.2 FCQL アルゴリズム

Q 学習では、状態と行動の組に対して評価値を設定し、この評価値を手がかりに学習が進行する。この評価値を Q 値と呼び、 Q 値を導く関数を Q 関数と呼ぶ。最も単純な Q 関数の実現方法は、すべての状態と行動の組について表を作り、内容として各組の Q 値を記録しておき、 Q 値を直接更新しながら学習を進めていく手法である。この実現方法は Table Lookup 法と呼ばれる。

FCQL では、現在の状態の入力、行動選択、行動の実行と報酬の獲得、学習と特徴構成による内部状態の更新までの一連の手続きを一周期とする。また、 Q 関数の実現には、離散的な環境での実装が容易な Table Lookup 法を用いている。図 2 に FCQL アルゴリズムを示す。

3.2.1 行動選択

行動選択手続き Policy では、まず s_t に含まれる特徴を判定する。特徴集合中にある j 番目の特徴について、 $f_j = \text{True}$ となる特徴を検出し、この操作を特徴集合中のすべての特徴について行う。

s_t に含まれる特徴が存在する場合は、過去に経験した状態と同様の特徴をもつ状態に遭遇したと判断され、各行動の評価値によって Boltzmann 分布に基づいた確率選択が行われる。 s_t に含まれる特徴が複数存在するときは、各特徴における報酬の和によって、Boltzmann 分布に基づいた確率選択が行われる。すなわち、状態 s_t に含まれる特徴 f_j に基づく行動 a_i を選択する確率 $p(a_i|f_j \cap f_j|s_t)$ は、

$$p(a_i|f_j \cap f_j|s_t) = \frac{e^{F(f_j, a_i)/T}}{\sum_{f_l \in s_t} \sum_{a_k \in A} e^{F(f_l, a_k)/T}} \quad (9)$$

となる。ただし、 T は温度定数で、値が大きいほど行動はよりランダムになり、それぞれの行動を選択する確率の差が小さくなる。逆に、 T を 0 に近づけると、わずかな Q 値の差が行動選択に大きく影響し、極限では Q 値を最大にする行動が選ばれる。つまり、

$$a_t = \arg \max_b Q(s_t, b) \quad (10)$$

MAIN LOOP:

- (1) 現在の状態 s_t を入力する.
- (2) 行動 a_t を選択する.
 $a_t \leftarrow \text{Policy}(s_t)$.
- (3) a_t を実行し, 次の状態 s_{t+1} に遷移し, 報酬 r_t を獲得する.
- (4) 内部状態を更新する.
 $\text{Learn}(s_t, a_t, r_t, s_{t+1})$.
- (5) (1) に戻る.

Policy(s_t):

以下に示す優先順位で, いずれか一つの処理が選択される.

- (1) s_t に含まれる特徴が存在すれば, 特徴に基づく行動選択を行う.
- (2) s_t と同一の状態が Q 表に存在すれば, 状態に基づく行動選択を行う.
- (3) ランダムに行動を選択する.

Learn(s_t, a_t, r_t, s_{t+1}):

以下に示す優先順位で, いずれか一つの処理が選択される.

- (1) 特徴に基づく行動選択を行った場合, 特徴関数を更新する.
- (2) 状態に基づく行動選択を行った場合, Q 関数を更新する.
- (3) $r_t \neq 0$ の場合, 新たに特徴を構成する.
- (4) 内部状態の更新を行わない.

図2 FCQL アルゴリズム
Fig. 2 The FCQL algorithm.

となる. 学習当初より報酬が最大の行動を選択すると, 環境の探索が不十分となるため, 確率的な行動選択を採用する⁹⁾. Boltzmann 分布による行動選択は, 各行動ごとの報酬を重みとした確率選択であり, 正の報酬が得られる行動を重視し, 負の報酬が得られる行動を小さな正の値で評価している. 報酬の値を選択確率に直接用いるのではなく, すべて正の値に変換するために, 式 (9) の分母が 0 になることを回避できる.

以上のような行動選択方式を特徴に基づく行動選択と呼ぶ. 特徴に基づく行動選択は, 以下に示す状態に基づく行動選択よりも優先的に処理される.

s_t に含まれる特徴が存在しない場合は, 学習者は s_t を Q 表と照合し, 最適と判断される行動 a_t を決定する. このとき, s_t と一致する状態が Q 表内に存

在すれば, Boltzmann 分布に基づく確率選択を行い, 行動選択手続きを終了する. すなわち, 状態 s_t で行動 a_i を選択する確率 $p(a_i|s_t)$ は,

$$p(a_i|s_t) = \frac{e^{Q(s_t, a_i)/T}}{\sum_{a_k \in A} e^{Q(s_t, a_k)/T}} \quad (11)$$

となる. この行動選択方式を状態に基づく行動選択と呼ぶ.

上記のいずれにも該当しない場合, s_t において可能な行動の中からランダムに行動が選択される.

3.2.2 学習と特徴構成

学習手続き Learn では, 行動 a_t の実行により遷移した状態 s_{t+1} と, 得られた報酬 r_t を用いて, システムの内部状態を更新する. このとき, 行動選択手続きでとった戦略により, 以下のように処理が分かれる.

行動選択手続きで特徴に基づく行動選択が行われた場合は, 同じ特徴をもつ状態を過去に経験したと判断できるので, 式 (8) に基づいて特徴関数を更新し, 学習手続きを終了する.

状態に基づく行動選択が行われた場合は, すでに経験済の状態であると判断できるので, 以下の式に基づいて Q 関数を更新し, 学習手続きを終了する (α は学習率, γ は割引率).

$$\begin{aligned} Q(s_t, a_t) &\leftarrow Q(s_t, a_t) + \alpha(r_t \\ &\quad + \gamma \max_{b \in A} Q(s_{t+1}, b) \\ &\quad - Q(s_t, a_t)) \end{aligned} \quad (12)$$

上記のいずれにも該当しない場合は, r_t によってさらに処理が分かれる. $r_t \neq 0$ のときは, 報酬が得られる特徴が存在すると判断できるため, 2.2 節で述べた手法で特徴の構成と選択を行い, 特徴集合に追加する. また, s_t を新たに Q 表に追加し, それぞれの行動に対する Q 値の初期値を

$$Q(s_t, a_i) = \begin{cases} r_t & (i = t) \\ 0 & (i \neq t) \end{cases} \quad (13)$$

と定義する. $r_t = 0$ のときには, Q 関数の更新, 特徴関数の更新, および特徴構成のいずれの処理も行わない.

4. FCQL による学習例

4.1 問題設定

本章では, 人工的な迷路問題⁷⁾ に対し FCQL を用いてシミュレーションを行い, その結果について述べる. 対象となる問題領域を採用した理由として,

- (1) 有限離散 Markov 環境である.

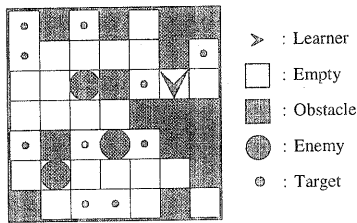


図3 シミュレーション環境

Fig. 3 Environments.

(2) 構成したい特徴が DNF で記述される。

といった点があげられる。なお、対象とする問題領域は、図3に示すような2次元の迷路を想定する。迷路は広さが7×7ブロックの格子状の環境で、この環境内には5種類の物体が存在する。

図3で矢頭型に描かれている物体が学習者の位置を表している。小さな点で示されているのが餌で、学習者が餌を捕らえると+0.5ポイントの報酬を得る。大きな円で描かれているのが敵で、出現地で静止しており、学習者が重なると-1.0ポイントの報酬を得て、その試行は終了する。矩形状に黒く塗りつぶされている物体は障害物で、障害物の向うにさらに障害物がある場合は、学習者は障害物のある方向へ進むことができない。それ以外の場合は、障害物の向うにある物体を押し潰しながら障害物とともに1ブロック進む。このとき、敵を押し潰すと+1.0ポイントの報酬を得る。また、白い矩形で表される領域は空白になっており、学習者はこの領域に進むことができる。学習者が入力として得られる属性は、学習者自身の位置、および学習者の周囲4方向×距離2ブロック、合計で9属性である。各試行開始時における学習者とすべての敵、障害物、および餌の出現地点は、それぞれ順に空白の中からランダムに選ばれるものとする。また、各物体の数をそれぞれ敵が3、壁が15、餌が10と定める。

学習者が行う状態の入力、行動選択、行動の実行に伴う報酬の獲得、および内部状態の更新までの一連の手続きを1ステップと定義する。また、学習者が敵に重なるか、あるいは100ステップが経過するまでの一連のステップ群を1試行とする。シミュレーションは最大100ステップの試行を2000回繰り返し、以上の作業5回の平均を取り、単位試行あたりの報酬値、および蓄積されている状態数を評価する。この問題領域では反射的行動獲得を行うため、常に最新の関数値を重視するように α と γ の値を高く、 T の値を中央よりやや低くして、アルゴリズム中の各パラメータを $\alpha = 0.9$, $\gamma = 0.9$, $T = 0.4$ と設定する⁹⁾¹¹⁾。

この問題領域が持つ特徴は、学習者の前後左右い

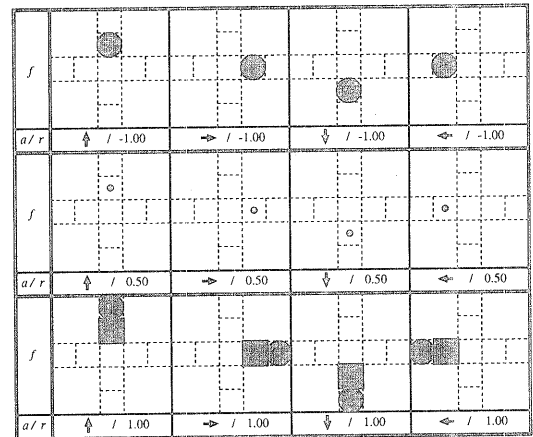


図4 問題領域に固有の特徴

Fig. 4 Specific features of this domain.

れかの1ブロックの距離に餌が存在するという単項の属性で記述される特徴が4種類、同様に1ブロック先に敵が存在する特徴が4種類、学習者の前後左右いずれかが壁で、その1ブロック先に敵が存在するという2項連言の特徴が4種類、計12種類である。図4に、この問題領域における特徴の正解を示す。

上段の f で示す図が対象領域の特徴で、学習者の周囲4方向の空間にある物体を示している。下段の a/r で示している部分が、それぞれ特徴 f に基づく行動とその評価値で、行動は矢印の方向へ進むことを表している。得られる3種類の報酬ごとに一つのクラスとすると、これらの特徴は各クラスごとのDNFで記述できる。

4.2 シミュレーション結果

本節ではFCQLを用いたシミュレーション結果を示す。比較対象として、構成したい特徴が最初から与えたFCQL、および従来のTable-Lookup型Q学習を取り上げ、理論的な収束値との比較、および改良による学習効率の変化を評価する。本シミュレーションでは、構成したい特徴を最初から与えたFCQLが収束する値を理論値と見なしている。評価対象は、学習中に蓄積された状態数、及び単位試行あたりの平均報酬値の推移とする。本シミュレーションでは特徴構成を始める状態数を400個と定める。なお、ディレイをかける状態数については、6章の問題点で再び検討する。図5、図6は、それぞれ学習中に蓄積された状態数の推移、および平均報酬値の推移を示している。

図5より、状態数400のディレイをかけたにもかかわらず、蓄積状態数の大幅な増加はなく、従来のQ学習よりかなり少ない状態数で安定している。これは、一括して最初に蓄積した400個の状態が、対象とする

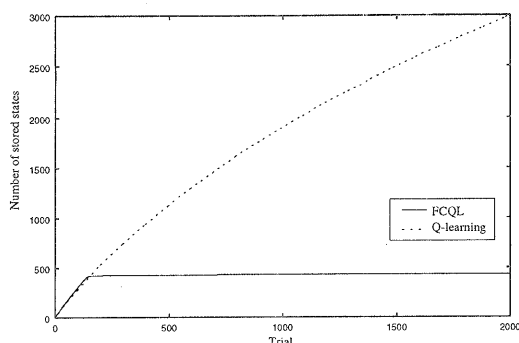


図 5 蓄積状態数の推移

Fig. 5 Number of stored status.

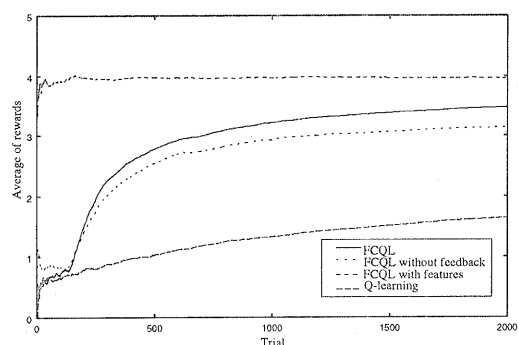


図 6 平均報酬値の推移

Fig. 6 Average of rewards.

領域の性質をよく反映したものであると考えられる。よって、ディレイをかけている間に蓄積された状態をもとに特徴を構成すると、以降の学習効率の向上に多大な貢献があったものと推測される。

図 6 より、ディレイを考慮した結果、FCQL は学習初期に急激な傾斜を描いて最適値へと収束し、その後も安定状態を続けている。改良前の状態と比較すると、学習初期での学習速度、および収束ポイントの双方で改善が見られる。また、ディレイによる遅れは、結果として学習全体に影響を及ぼすことはなかったと考えられ、学習効率の向上が見られた。また、強化学習からのフィードバックを除いた場合、特徴関数の収束が遅れるため、収束値が若干劣る結果が表れている。

次に、学習中に構成された特徴の例を図 7 に示す。この例では、実際に構成された特徴は 28 個で、そのうち 12 個を構成された順に示している。

正解となる特徴のうち、2 項連言で記述される特徴については、4 つすべてが構成されていた。単項で記述される特徴は、他の不要な属性が混在し、冗長な記述となっているが、報酬と無関係な特徴は一つしか構

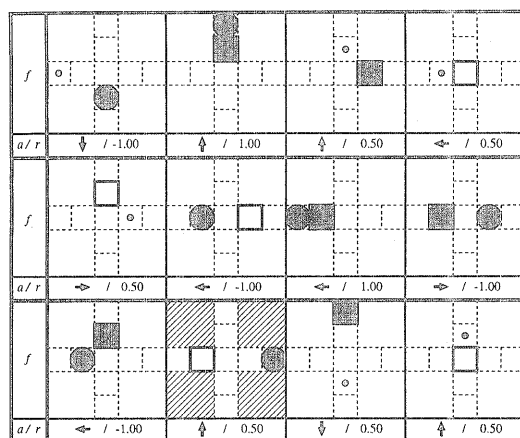


図 7 構成された特徴の一部

Fig. 7 Part of constructed features.

成されておらず（図 7 の斜線部）、蓄積された状態を分類するには充分であったと考えられる。また、構成された特徴の連言の項数はほとんどが 2 項までで、最長で 3 項のものがごくまれに見られる程度であった。

最後に、構成された特徴が実際に有効に利用されていることを検証する。図 8 は、各試行において、それぞれ特徴、および状態に基づいて行動が決定された回数の平均を表している。

FCQL では、状態に含まれる特徴があれば、優先してその特徴に基づく行動決定を行うが、図 8 より、特徴に基づいて行動が決定された回数の方が試行全体として多いと判断できる。これは、学習初期で構成された特徴が、同じクラスに属する状態の性質をよく表しているため、未経験の状態に対しても同じ特徴を持つ経験済みの状態と同じ行動を選択し、成功したためであると考えられる。したがって、正の報酬が得られる特徴を持つ状態に対しては過去の行動を積極的に追従し、負の報酬が得られる特徴を持つ状態に対しては過

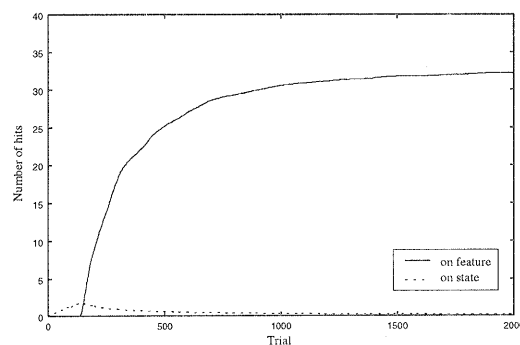


図 8 特徴および状態に基づく行動決定回数

Fig. 8 Number of decisions based on features or status.

去の行動を避ける様に判断され、単位時間あたりの報酬の向上に大きく貢献している。

5. 関連研究

5.1 G algorithm

G algorithm³⁾は、 Q 学習の枠組みにおいて行動空間を固定し、入力となる状態を一般化して状態空間を再構成する手法である。状態の分類は報酬に基づいて行われる。また、対象とする領域はすでに抽象化された離散的な空間であり、これらの点において FCQL と類似している。

G algorithm 独自の実装として、報酬を予測する D 関数を定義している。 D 関数は、状態、行動、報酬の組から将来への報酬の重み和を見積もる関数である。このため、長期にわたる報酬獲得のための行動選択である合目的行動獲得に優れており、この点において FCQL における反射的行動獲得とは逆の立場にある。状態空間の再構成では、与えられた属性によって状態空間を分割しており、得られた状態から対象領域の特徴を構成するものではない。

5.2 ロボットの行動獲得のための状態空間の自律的構成

浅田²⁾による手法は、実世界の連続空間を対象とし、行動に基づいて状態空間を再構成するロボット学習システムである。状態空間の再構成では、ゴールまでの行動数に基づいて状態の集合を構成しており、合目的行動獲得を行う。この手法は実世界を念頭に置いた手法であるため、連続空間に強いという利点がある。

この分割手法では、行動に基づく時間的な抽象化を行っており、状態に基づく空間的な分割は行われていない。示されている実験結果からは、各状態集合に対応する行動が、空間的にゴールへの距離がより遠い行動が選択されている場合があり、必ずしも空間的に最適な行動になっていない。これは、各状態集合ごとに属する状態の空間的な類似性が低く、結局ロボットの視点を中心とした行動要素に基づく分割しか達成できていないことを示している。

5.3 EOPs

EOPs⁵⁾では、 Q 学習をベースに、状態入力をもとにクラスを識別する関数 EOP を定義し、この関数に基づいて状態空間を分割、再構成する手法である。EOP は入力属性の重み和で計算された値の大きさによって 2 値を返す関数で、この値によって 2 クラスを判別している。

EOPs では、データとなる状態を獲得した後に空間の分割が行われ、状態を蓄積しながら同時に空間を

再構成するという手続きはとられていない。また、獲得戦略もランダムで、状態の蓄積と空間の構成は全く別々に扱われており、構成された空間を再評価する手続きが欠けている。結果として、少なくとも一度は対象領域を下調べする必要性に迫られる。一方、FCQL では Q 学習の戦略に基づいて一定量の状態を蓄積しながら特徴を構成し、それらの特徴を再評価する手続きが学習中に組み込まれているため、未知の環境に対して適応性が高いと言える。

6. おわりに

本稿では、FCQL を人工的な迷路問題に適用し、報酬の推移についてシミュレーションを行った。環境の例として取り上げた迷路問題は、ロボット学習の範疇で用いられる代表的な問題の一つである。しかしながら、実際のロボットを実験に用いる場合には、センサからの情報も連続値であり、FCQL の枠組に組み込むには適切な方法で離散化する必要がある。本稿での迷路問題はすでに離散化された状態にあり、一般的な実数値の連続空間を対象とする問題に対しては、適切な離散化を行う操作子の適用によって対応が可能であると考えられる。また FCQL では、特徴を構成する操作子として論理積を採用したが、他にも特徴を構成する操作子に発見的な手法の採用が考えられ、この点についても検討の余地がある。対象領域とした DNF 問題は、 Q 学習のみでは非効率な問題の例として取り上げたが、最終的には入力属性の組合わせて特徴を構成できるという前提に立っている。状態を識別する要素が、センサからの情報に全く現れてこない、いわゆる隠れ状態を扱う問題では、得られた状態からの特徴構成ができないため、他の状態再構成の手法が有利となる場合がある。

FCQL の問題点として、ディレイをかける状態数を試行中に決定できない点があげられる。図 9 は、ディレイをかける状態数をそれぞれ 0, 100, 200, 300, 400, 500, 700 に設定した場合の報酬の推移を示している。

状態数が少ない場合は、構成される特徴の精度が低いため、不安定な振る舞いを示している。逆に、状態数を増やすと、学習初期において特徴構成を行わないため、報酬の向上が遅くなる。2000 試行終了時において、ディレイをかける状態数を 400 に設定した場合の収束値が最も高いので、本シミュレーションでは特徴構成を始める状態数を 400 個と決定した。今後は、特徴構成を始めるために十分な状態数を決定するための理論的な指標を検討する必要がある。

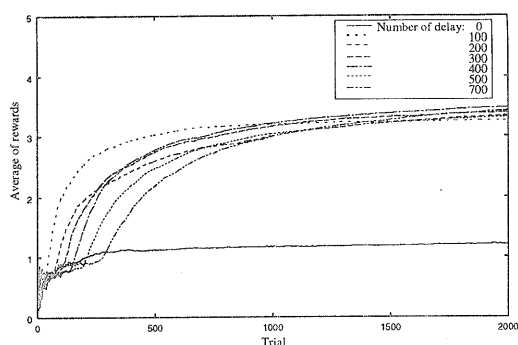


図9 ディレイをかける状態数を変化させた場合の報酬の推移
Fig. 9 Average of rewards testing several number of the delay.

参考文献

- 1) Aha, D. W.: Incremental constructive induction: an instance-based approach, *Proc. of the Eighth International Workshop on Machine Learning*, pp. 117-121 (1991).
- 2) Asada, M., Noda, S., and Hosoda, K.: Action-based sensor space categorization for robot learning, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1502-1509 (1996).
- 3) Chapman, D. and Kaelbling, L. P.: Input generalization in delayed reinforcement learning: an algorithm and performance comparisons, *Proc. of the Twelfth International Joint Conference on Artificial Intelligence*, pp. 726-731 (1991).
- 4) Hu, Y. J. and Kibler, D.: Generation of attribute for learning algorithms, *Proc. of AAAI-1996*, pp. 806-811 (1996).
- 5) Ishiguro, H., Sato, R., and Ishida, T.: Robot oriented state space construction, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1496-1501 (1996).
- 6) Kaelbling, L. P.: Learning functions in k -DNF from reinforcement, *Proc. of the Seventh International Conference on Machine Learning*, pp. 162-169 (1990).
- 7) Maclin, R. and Shavlik, J. W.: Incorporating advice into agents that learn from reinforcements, *Proc. of AAAI-1994*, pp. 694-699 (1994).
- 8) Quinlan, J. R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993).
- 9) Sutton, R. S.: Integrated architectures for learning, planning, and reacting based on approximating dynamic programming, *Proc. of the Seventh International Conference on Machine Learning*, pp. 216-224 (1990).
- 10) 滝 寛和: 構成的帰納学習とバイアス, 人工知能学会誌, Vol. 9, No. 6, pp. 818-822 (1994).
- 11) Tan, M.: Multi-agent reinforcement learning: independent vs. cooperative agents, *Proc. of the Tenth International Conference on Machine Learning*, pp. 330-337 (1993).
- 12) 畠見達夫: 強化学習, 人工知能学会誌, Vol. 9, No. 4, pp. 830-836 (1994).
- 13) Watkins, C. J. C. H.: *Learning from Delayed Rewards*, Ph.D thesis, Cambridge University Psychology Department (1989).
- 14) Watkins, C. J. C. H. and Dayan, P.: Technical note on Q-learning, *Machine Learning*, Vol. 8, pp. 279-292 (1992).

(平成 10 年 10 月 30 日受付)

(平成 10 年 12 月 21 日採録)



宮本 行庸 (学生会員)

昭和 45 年生。平成 6 年神戸大学工学部システム工学科卒業。平成 8 年同大学院自然科学研究科情報知能工学専攻博士前期課程修了。現在、同大学院自然科学研究科知能科学専攻博士後期課程在学中。主に人工知能、機械学習の研究に従事。人工知能学会学生会員。



上原 邦昭 (正会員)

昭和 29 年生。昭和 53 年大阪大学基礎工学部情報工学科卒業。昭和 58 年同大学院博士後期課程単位取得退学。大阪大学産業科学研究所助手、講師、神戸大学工学部情報知能工学科助教を経て、同大学都市安全研究センター教授。情報知能工学科を兼任。平成元年より 2 年まで Oregon State University, Visiting Assistant Professor。平成 6 年より 8 年まで神戸大学総合情報処理センター副センター長。工学博士。人工知能、特に機械学習、マルチメディアデータベース、自然言語によるヒューマンインタフェースの研究に従事。1990 年度人工知能学会研究奨励賞受賞。人工知能学会、電子情報通信学会、計量国語学会、日本ソフトウェア科学会、システム制御情報学会各会員。