計測自動制御学会東北支部 第175回研究集会 (1998.6.19)
資料番号 175-5

クラシファイアシステムを用いたロボット学習に関する研究

A study on Robot Learning using a Classifier System

○釜谷博行*, 阿部健一**

○Hiroyuki Kamaya*, Kenichi Abe**

*八戸工業高等専門学校, **東北大学大学院工学研究科

*Hachinohe National College of Technology, **Faculty of Engineering, Tohoku University

キーワード： 機械学習 (Machine Learning), クラシファイアシステム (Classifier System), 遺伝的アルゴリズム (Genetic Algorithm), バケツリレーアルゴリズム (Bucket Brigade Algorithm), ロボット学習 (Robot Learning)

連絡先： 〒039-1192 八戸市田面木字上野平16-1 八戸工業高等専門学校 電気工学科
釜谷博行, Tel.: (0178)27-7283, Fax.: (0178)27-9379, E-mail: kamaya-e@hachinohe-ct.ac.jp

1. はじめに

環境変化やロボット構成要素の故障など, あらかじめ予期されない出来事に対処するため, ロボットが自らそれらに適応できるように設計することが, 実世界で作業するロボットを開発する上で重要となる。適応能力を実現する手段のひとつとして遺伝を基盤とした機械による学習, GBML(Genetic Based Machine Learning)が提案されており, その代表的なものとしてクラシファイアシステム (Classifier System, 以下CSと称す) がある[1]。CSには, Hollandらのルールセットを集団と考えるミシガンアプローチとDe Jengらのルールセットを個体と考えるピッツアプローチの二つがある[2]。本研究ではオンライン処理に向くミシガンアプローチを用いる。

CSは, これまでブール関数の学習問題やロボットのナビゲーション問題など, さまざまな問題に適用されている。しかし, それらの多くは学習システムの出力に対してすぐに評価を与える形となっている。例えば, ロボットのナビゲーション問題では, ゴールに近づくにつれて大きくなるような評価値がロボットの行動毎に与えられる[3]。

このため, ルールを順次起動していき, 長いルールシーケンス経過後, タスクが達成されたときに初めて正の報酬 (reward) が得られるというような報酬遅れのある問題に直接適用できない。この場合, 報酬獲得までに順次起動しなければならないルールシーケンスを, 報酬を頼りに自動的に生成するメカニズムが必要となる。

そこで, 本研究では, このようなメカニズムを実現するため, 従来提案されている報酬割り当て方法に改良を加えた後, ロボットの行動学習へ適用し, 本学習システムの有効性を実験的に検証する。

2. クラシファイアシステム

クラシファイアシステムでは, プロダクションルールをビット列で表現した分類子 (classifier) に

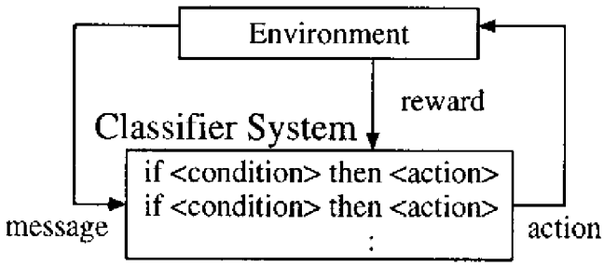


Fig. 1　クラシファイアシステムと学習環境

遺伝的な操作を加え, 所望の知識を獲得させるメカニズムを有している。また, 環境の動的な変化に対しても, 獲得された知識に遺伝的操作を加えることで, 比較的容易に適応することが可能である。

2.1　分類子

分類子は, つぎのように定義される。

$$< classifier > = < condition > : < action >$$
$$if \ < condition > \ then \ < action >$$

環境情報$< message >$と分類子の$< condition >$部分を比較し, 適合したルールが存在すれば, 対応する$< action >$を出力する。環境は, この$< action >$に対する評価値を CS に出力する。それぞれの分類子は, この評価値を用いてルールの良さを表す数値である強度 (strength) を更新していく (Fig.1)。

メッセージ, 条件部, 行動部はそれぞれつぎのように表される。

$$< message > = \{0,1\}^n$$
$$< condition > = \{0,1,\#\}^n$$
$$< action > = \{0,1\}^n$$

条件部の#は don't care で, メッセージの対応するビットが 0, 1 のいずれの場合にも適合することを表す。例えば,

$$< condition > = \{\#01\#1\}$$

をもつ分類子は,

$$\{00101\}, \{00111\}, \{10101\}, \{10111\}$$

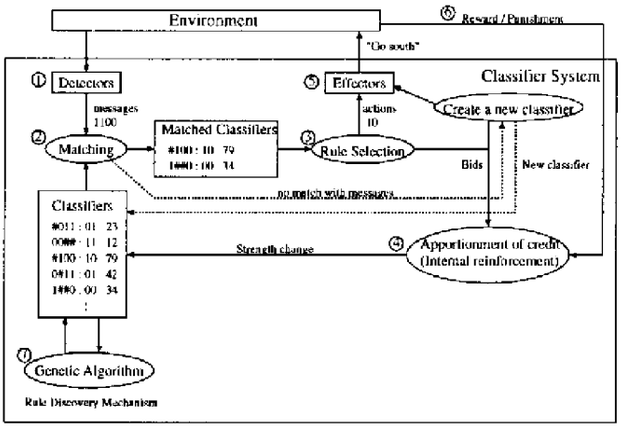


Fig. 2　クラシファイアシステムの処理の流れ

の 4 つのメッセージに適合する。

分類子どうしで互いに協調させながら問題解決を図るため, ゴール近くで受け取った報酬をゴールから遠いところまで伝搬させる必要がある。これには, Bucket Brigade Algorithm (BBA) と Profit Sharing Plan (PSP) が提案されている[4]。本研究では, 報酬獲得まで選択された分類子の全シーケンスを記憶せず, 前回選択された分類子のみを記憶するという点において, わずかなメモリで済む BBA を利用する。

2.2　CS の処理手順

CS の大まかな処理手順は以下のとおりである (Fig.2)。

1) 環境情報を 0, 1 系列にコード化し, CS にメッセージを入力する。

2) 分類子集団の中からメッセージに適合する分類子を探す。

3) それぞれの分類子の強度に応じて, 競合するルールの中からボルツマン分布により確率的に 1 つの分類子を選択する。具体的には分類子 C_a を選択する確率は次のように表される。

$$Prob(C_a) = \frac{\exp^{\frac{S_a}{T}}}{\sum_{k \in match} \exp^{\frac{S_k}{T}}} \quad (1)$$

ここで, C_aはメッセージと適合する分類子を, S_aはその強度を表す。Tは温度係数 (temperature) と呼ばれ, 分類子選択においてランダムさの程度を調整するパラメータである。

4) 選択された分類子 C_iの強度を

$$S_i' = S_i - B_i \quad (2)$$

とする。ただし,

$$B_i = C_{bid} S_i \quad (3)$$

C_{bid}は強度に対する割合を表す。

報酬伝搬を行なうため, 1 時刻前に選択した分類子 C_jの強度を

$$S_j' = S_j + \gamma B_i \quad (4)$$

とする。ここで, γを割引率 $(0 \leq \gamma \leq 1)$ と呼ぶことにする。

5) 選択された分類子のアクションを実行する。

6) 環境から報酬 Rが得られたならば, 選択された分類子 C_iの強度を

$$S_i'' = S_i' + R \quad (5)$$

に更新する。

7) あらかじめ設定された強度 $minStrength$ 以下になった分類子を削除し, GA を用いて新たな分類子を生成する。

8) 手順 1) へ戻る。

なお, 学習開始時は分類子の初期集団をランダムに生成する。

手順 2) においてメッセージと適合するルールが存在しない場合には, 最も強度の小さな分類子を削除し, メッセージを包含する新たな分類子を生成する。このとき, 条件部はメッセージのそれぞれのビット位置において, ある確率で#へ置き換え, 行動部はランダムに生成する。

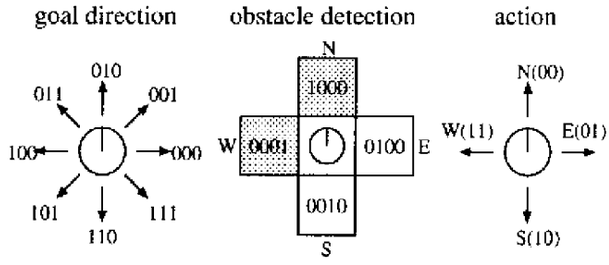


Fig. 3　センサ入力と行動

手順 7) の GA では, 交叉 (crossover) と突然変異 (mutation) を確率的に選択して実行する。交叉は, 分類子の強度に基づいて親を選択し, 交叉ポイントをランダムに決定し行なう。また, 突然変異は分類子の強度に基づいて選択された親のビット列をある確率で誤ってコピーさせて行なう。

なお, 新たに生成された分類子の強度は, 初期値に設定する。

3. ロボットの行動学習

3.1 問題設定

グリッドワールドにおいて, センサを装備したロボットがスタート地点から出発し, 未知障害物を回避しながら最短でゴールへ向う行動の学習問題について考える。環境情報 (メッセージ) はセンサにより取得し, ゴール方向を 3bits, ロボット近傍の障害物を 4bits の合計 7bits で表す。行動は 4 方向の 2bits で表す (Fig.3)。

メッセージの例を Table 1に示す。障害物情報ビットは, 1 で障害物あり, 0 で障害物なしを表す。これは, Fig.4のように, ゴールが N 方向にあり, N, E 方向に障害物が存在し, S, W 方向には障害物が存在しない場合に対応する。

分類子の例を Table 2に示す。これは,

Table 1　メッセージの例

	ゴール方向 (3bits)	障害物情報 (4bits)
例	010	1 1 0 0 N E S W

Table 2　分類子の例

	条　　件	:	行　動	強　度
例	010 (N 方向)	#1#0 (NESW)	: 11 (W 方向)	234.0

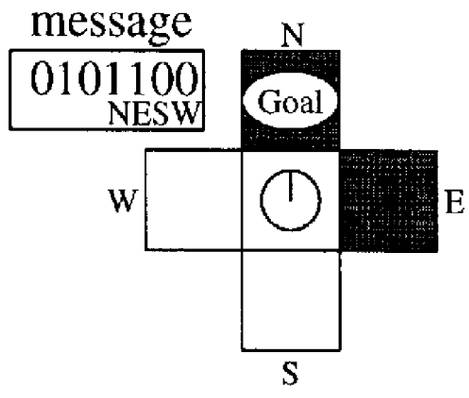


Fig. 4　メッセージの例

「ゴールが N 方向にあり, E 方向に障害物があり, W 方向に障害物がない」

ならば,

「W 方向へ進め」

このルールの「強度は *234.0*」である

を表している。これを図示すると, Fig.5のようになる。

報酬として, ゴール到達時のみに+1000, 障害物の方向へ移動しようとすると−20, それ以外は移動毎に−2が与えられる。1000 ステップ実行後もゴールへ到達できなかった場合は, 報酬は 0 となり, つぎの試行へ移る。ここでは, ロボットの x, y座標といった情報やゴールまでの距離に応じて逐一報酬を与えるようなメカニズムを利用していない。

分類子を生成する場合, 一般的に CS では, #(don't care) の比率は全ビットのほぼ 20〜30%であることから, 本実験でも確率 20%で#(don't care) を含むようにした。

また, 分類子の強度の初期値は 100, その他のパラメータは $C_{bid} = 0.2$, $\gamma = 0.9$, $minStrength = 20$, $T = 0.5$ とした。

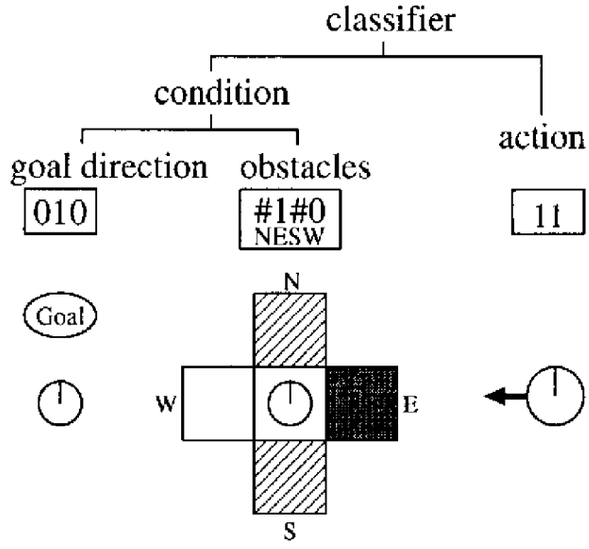


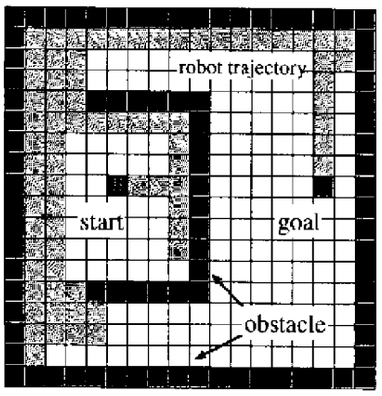
Fig. 5　分類子の例

Fig.6は学習初期と学習後のロボットの行動軌跡である。学習初期ではゴール到達までに無駄な行動を実行しているが, 学習後は少ない step 数でゴールへ到達していることがわかる。

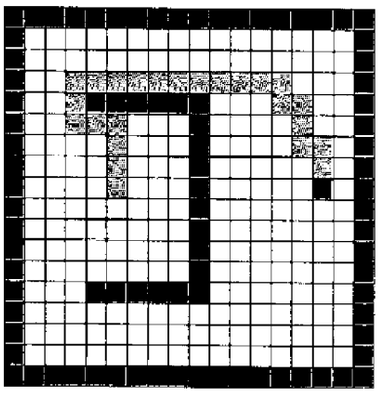
3.2　GA の効果について

GA の効果を調べるため, ここでは, 手順 7) を削除した GA 無しの場合と, GA 有りの場合について比較した。分類子の数は, GA 無しが 500 で, GA 有りが 20 である。

Fig.7は, 20 種の乱数パターンの結果を平均した学習曲線で, 横軸が trial 数を縦軸がゴール到達までに要した step 数を表す。optimal は最適値を表わしており, ゴールまでの最短ステップ数は 24 である。学習の進行とともに step 数が減少しているが, GA を用いると小集団でも良好な結果が得られることがわかった。なお, GA 無しの場合には, 集



(a) 学習初期



(b) 学習後

Fig. 6　ロボットの軌跡

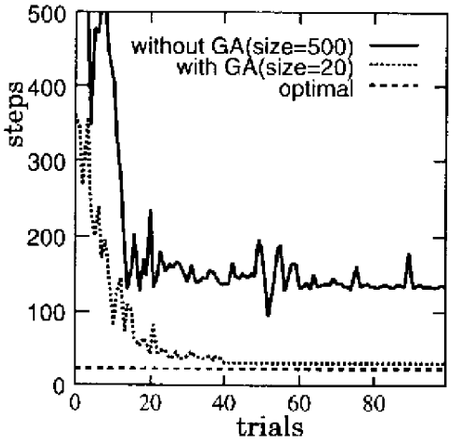


Fig. 7　学習曲線 (GA の効果)

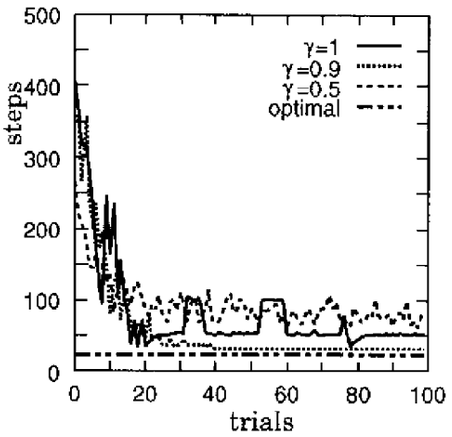


Fig. 8　学習曲線 (割引率γ の効果)

団が小さいとゴールへ到達できなかった。

今回の実験では, don't case を考慮に入れないと, 2^7個の条件が必要である。また, それぞれの条件に対して 2^2個の行動が考えられる。そのため, $2^7 \times 2^2 = 2^9 = 512$ 個の分類子が必要となる。実験結果から, GA を利用することによって, かなり小さな分類子集団で学習可能であるといえる。

3.3　割引率γ の効果について

Fig.8は割引率γ の値のみを変えて実行した結果である。$\gamma = 0.9$ で良好な結果となった。$\gamma = 1.0$, $\gamma = 0.5$では, あまりよくない。これは, $\gamma = 1.0$では, 2つの分類子のあいだで, ゴール到達とは無関係のループが形成されるためであると考えられる。また, $\gamma = 0.5$では, ゴールから遠いルールまで報酬がうまく伝搬されないためであると考えられる。

Table 3に獲得された分類子の例を示す。初期強度の 100 を超えるものについて, 強度の大きい順に列挙した。また, それぞれの分類子の対応する状況を Fig.9に示した。このことから, 良好なルールが獲得されていることがわかる。

4.　おわりに

本研究では, CS の報酬割り当て方法に改良を加え, グリッドワールドにおけるロボットの行動学習で報酬が遅れる場合に適用した結果, 小さな分類子集団でも良好な学習性能を確認できた。

今回の実験は, センサ情報, 行動ともにノイズのない理想的な状況で行なっている。今後は, 本学習システムをノイズのある複雑な環境へ適用したい。

Table 3　獲得された分類子

condition	action	strength
1100#00	10	3744.7
#1100#0	01	3094.5
00#0#00	00	1829.4
00010#0	11	1210.4
11#0001	01	340.5
#11#0#0	11	175.5

また, 乱数のシード値を変えると, うまく最適値へ収束しない場合がある。誤収束を避け, 最適値へ収束させるための方法について検討したい。

参考文献

[1] Goldberg D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley (1989)

[2] 坂和, 田中: 遺伝的アルゴリズム, 朝倉書店 (1995)

[3] Venturini G.: Adaptation in dynamic environments through a minimal probability of exploration, From animals to animats 3, *SAB94*, MIT Press, 371/379 (1994)

[4] 堀内, 藤野, 片井, 椹木: 経験強化を考慮した *Q*-Learning に関する一考察, SICE システム情報関連合同シンポジウム講演論文集, 81/86 (1995)

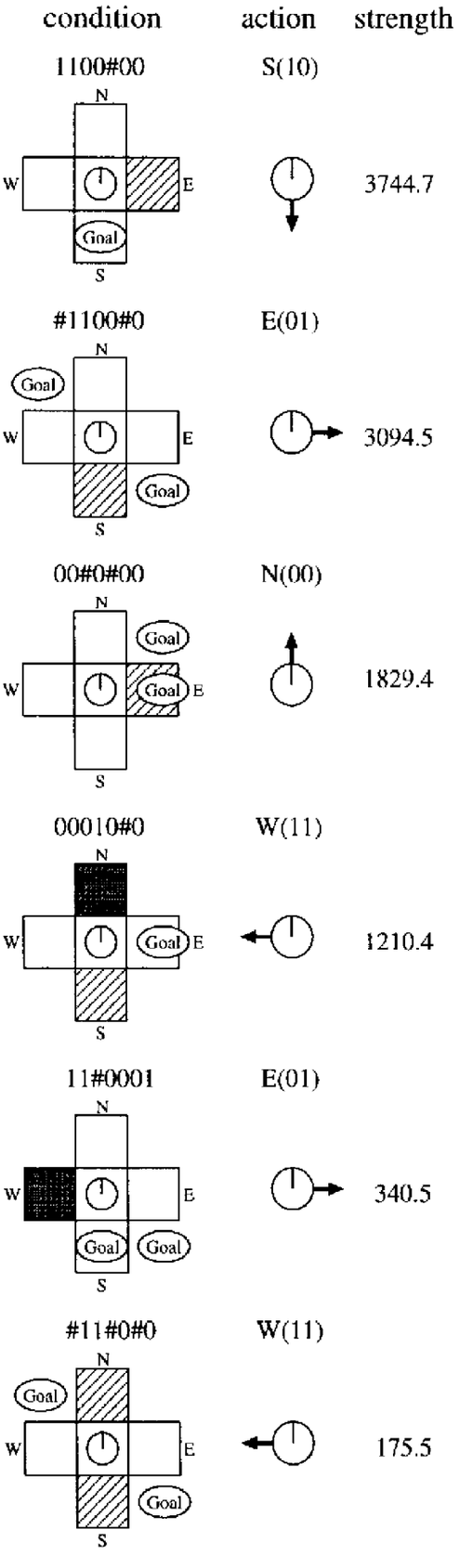


Fig. 9　獲得された分類子