個別化による学習分類子システムの一般化促進

中 田 雅 也*・原 田 智 広*・佐 藤 圭 二*
松 島 裕 康*・高 玉 圭 樹*

Promoting Generalization in Identification Based Learning Classifier System

Masaya NAKATA*, Tomohiro HARADA*, Keiji SATO*,
Hiroyasu MATSUSHIMA* and Keiki TAKADAMA*

This paper proposes a novel Learning Classifier System (LCS) called Identification-based LCS (IXCS) to promote a generalization of classifiers (i.e., rules) by selecting effective ones and deleting ineffective ones. Through the intensive simulation of the 20-Multiplexer problem, this paper has revealed the following implications which cannot be achieved by the conventional LCS, XCSTS: (1) IXCS can not only generalize the classifiers earlier but also generate the classifiers which are robust to the noisy environment; and (2) IXCS can derive a higher performance with a lower number of micro-classifiers.

Key Words: LCS, XCS, generalization, identification

1. はじめに

学習分類子システム (Learning Classifier System: LCS¹⁾) は生物の環境への適応戦略である学習と進化の双方の概念を取り入れた適応システムである。この LCS は環境との相互作用を通じて分類子 (classifier) と呼ばれる条件-行動ルールを Q 学習²⁾などの強化学習³⁾の枠組みで学習を行なうとともに遺伝的アルゴリズム (Genetic Algorithm: GA⁴⁾) を用いて分類子を進化させることで、複数の環境状態に適応可能である汎用的なルールをもつ分類子を生成する。これを分類子の一般化 (generalization) と呼ぶ。現在の主流である正確性に基づく学習分類子システム (Accuracy-based Learning Classifier System: XCS⁵⁾) は、正確かつ適切に一般化された分類子を学習することが可能であるがつぎの問題点をかかえている。まず、1) XCS は学習初段階において多種類の分類子を生成し環境に適応する分類子を探索するため、環境のもつ状態空間の大きさに応じて分類子を記憶する上限数 (max population size) を適切に設定する必要がある。さらに、2) XCS では分類子の多様性を制御する GA において正確性 (accuracy) の高い分類子が進化対象となるため、雑音環境下では誤った正確性が見積もられ学習性能が低下する。一般的に環境に適応する分類子は分類子の有効性に応じて、1) 報酬獲得に関与す

る分類子 (有効な分類子)、2) 報酬獲得に関与しない分類子 (無効な分類子) に識別できるが、XCS では分類子の有効性は考慮されず同等の基準で GA が行なわれる。この問題に対して、本論文では報酬の有無に基づき分類子の有効性を識別する個別化 (identification) を提案し、個別化を XCS に組み込んだ、個別化による学習分類子システム (Identification based XCS: IXCS) を提案する。また、LCS の一般的なベンチマーク問題である Multiplexer 問題⁵⁾を用いて、提案手法の学習性能を検証する。

本論文の構成はつぎのとおりである。以下、第 2 章では学習分類子システム (XCS) について述べ、第 3, 4 章では個別化および個別化による学習分類子システム (IXCS) について述べる。第 5 章では Multiplexer 問題における実験結果を示し、第 6 章で考察する。最後に第 7 章で本論文をまとめる。

2. 正確性に基づく学習分類子システム (XCS)

2.1 概要

XCS の学習過程を Fig. 1 に示す。XCS では環境からの入力である状態に対して適切な行動を実行することで環境から得られる報酬 (reward) を最大化するように学習する。なお、Fig. 1 の検出器 (detector) と効果器 (effector) は環境の状態を XCS の表現に変化し、XCS の行動表現を環境への行動に変更する。以下、XCS における分類子について説明し、つぎに学習過程を構成するメカニズムである実行部 (performance component)、強化部 (reinforcement component)、そして発見部 (discovery component) を順に説明する⁶⁾。最後に、XCS における分類子がつぎ条件の一般化を行なうための包摂 (subsumption) について説明する。XCS⁷⁾ではこれらの組み

* 電気通信大学大学院情報理工学研究科 調布市調布ヶ丘 1-5-1

* Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu

(Received March 9, 2011)

(Revised September 1, 2011)

Fig. 1 Overview of XCS

合わせにより学習過程が実現される。

2.2 分類子

分類子 (classifier) は条件-行動ルールであり、環境状態と照合する条件部 (condition: C) と、その条件部と照合したときに実行する行動を示す行動部 (action: A) をもち、行動部は 0, 1 のビット列で表現される。一般的に条件部は 0, 1 のビット列で形成されるが、任意の値を示す # (don't care) を組み込むことによって、複数の環境状態に適応可能な汎用的な条件部が形成される。XCS における分類子は予測値 (prediction), 誤差値 (error) および適合度 (fitness) の評価値をもつ。XCS では、条件部および行動部が共に等しい分類子や、後述する包摂の操作により統合された分類子の集合をマクロ分類子 (macro-classifier) として集約し、集約した数を重合度 (numerosity) と呼ぶ。また記憶する分類子の上限数 (max population size) をパラメータ N と定める。 j 番目の分類子 cl_j の予測値, 誤差値, 適合度, 重合度を、それぞれ p_j , ϵ_j , F_j , num_j と記す。以下、分類子に対する操作は分類子を単位として記述するが、実際には同様の効果をもたらすマクロ分類子の操作として実現される。

2.3 メカニズム

2.3.1 実行部

実行部は分類子集団 (population: $[P]$) の中から環境状態と照合する分類子の行動を選択し出力する処理を担う。この処理を Fig. 1 を用いて説明する。環境より入力される状態 (state) は、一般的に 0, 1 からなるビット列で形成される。入力された状態は、分類子集団の各分類子の条件部と照合され、照合に成功した分類子は照合集合 (match set: $[M]$) を形成する。ここで入力状態に一致する分類子が存在しない場合は、入力状態に照合する分類子を作り出す被複 (covering) と呼ばれる操作が行なわれる。被複で生成される分類子の条件部は状態と同じビット列で形成されるが、確率 $P_{\#}$ で # に置き換えられ、行動はランダムに選ばれる。

つぎに、各行動に対する予測報酬として予測配列 (prediction array) が計算される。予測配列における各行動 a_i の予測値 $P(a_i)$ は (1) 式で定義される。ただし式中の $[M]_{|a_i}$ は、

照合集合中で行動部に行動 a_i をもつ分類子の集合を示す。ただし、照合集合に存在しない行動の予測値は、null とされ以下の行動選択の対象とならない。この予測配列 (各行動に対する予測報酬) に従って行動選択 (action selection) を行ない、照合集合から選択された行動を行動部にもつ分類子をまとめて行動集合 (action set: $[A]$) を形成する。そして、選択された行動が実行される。この処理単位をステップと呼ぶ。

$$P(a_i) = \frac{\sum_{cl_k \in [M]_{|a_i}} p_k \times F_k}{\sum_{cl_k \in [M]_{|a_i}} F_k} \quad (1)$$

2.3.2 強化部

強化部は各ステップにおける実行部の処理が完了した後、環境から得られた報酬に基づいて行動集合を形成する各分類子の予測値, 誤差値および適合度を (2)~(4) 式のように更新する。この処理は、Fig. 1 において $[A]_{-1}$ で表わされる 1 ステップ前の行動集合 (previous action set) に含まれる各分類子 cl_j に対して行なわれ、ただし r は獲得した報酬であり、左矢印は左辺への代入を意味する。

$$P \leftarrow r + \gamma \max_a P(a) \quad (2)$$

$$p_j \leftarrow p_j + \beta(P - p_j) \quad (3)$$

$$\epsilon_j \leftarrow \epsilon_j + \beta(|P - p_j| - \epsilon_j) \quad (4)$$

P は予測値 p_j を更新する際の目標値であり、前ステップの報酬 r と予測配列中の最大予測値 $\max_a P(a)$ を用いて計算される。パラメータ β ($0 \leq \beta \leq 1$), γ ($0 \leq \gamma \leq 1$) はそれぞれ学習率 (learning rate), 割引率 (discount factor) と呼ばれ、それぞれ学習の更新速度と将来の報酬を考慮する度合いを制御する。そして、更新後の誤差値 ϵ_j に基づき分類子の絶対的な正確さ κ_j と $[A]_{-1}$ における相対的な正確さ κ'_j が (5)~(7) 式のように計算され、最後に適応度 F_j が更新されることで強化部における一連の処理が完了する。ここで適応度 F_j の計算に正確さ κ_j を用いる点が XCS の特徴である。

$$\kappa_j = \begin{cases} 1 & \text{if } \epsilon_j \leq \epsilon_0 \\ \alpha(\epsilon_j/\epsilon_0)^{-\nu} & \text{otherwise.} \end{cases} \quad (5)$$

$$\kappa'_j = \frac{(\kappa_j \times num_j)}{\sum_{cl_k \in [A]_{-1}} (\kappa_k \times num_k)} \quad (6)$$

$$F_j \leftarrow F_j + \beta(\kappa'_j - F_j) \quad (7)$$

2.3.3 発見部

発見部では GA による分類子の生成と削除を通して、分類子集団 $[P]$ を進化させる。分類子は自分が対象となった前回 GA の発動時の時間を記憶する。XCS における GA は、前ステップ行動集合 $[A]_{-1}$ を形成する各分類子について、この時間と現在の時間を比較し前回 GA 発動時から経過した時間をそれぞれ求め、経過した時間の平均値がパラメータ θ_{GA} の値を上回る場合に発動される。GA が発動すると、 $[A]_{-1}$ の各分類子の適応度の比率を選択確率として 2 つの分類子が親個体として選択 (selection) される。つぎに、子個体として、それぞれの親個体の分類子と同様の条件部、行動部および各

評価値をもつ分類子が2つ生成され交叉 (crossover) および突然変異 (mutation) がそれぞれパラメータ χ , μ の確率で適用される。交叉は、同じ行動部をもつ親個体として選択された条件部を遺伝子と捉え、2つの親個体間の条件部を組み合わせることで新たな条件部を生成する。また突然変異は条件部を変異確率 μ で、ランダムな条件部に置き換える。最後に、子個体である2つの分類子は分類子集団に追加される。また、分類子集団中の分類子数がパラメータ N を超えた場合は N を超えた数だけ適合度の低い分類子が削除 (deletion) される。 N を超えないときは削除されない。

2.3.4 包摂

包摂は分類子集団の中で、ある分類子と同じ行動部をもち、かつ、より一般化された条件部をもつ分類子が存在した場合に前者を後者に統合することで、分類子集団における分類子条件部の一般化を促進する。ここで、2つの分類子間でどちらの条件部がより一般化されているかの判断は一般性比較演算 (is-more-general operator) として定義される。マクロ分類子が条件部と行動部が共に等しい分類子集合を束ねる表現であるのに対し、包摂は一方の条件部を他方の条件部へと変更する操作であり、結果として分類子集団内のマクロ分類子の数の減少に寄与する。

3. 個別化

3.1 概要

一般的に、環境状態における最適な行動は状態ごとに異なるが、XCS は最適な行動が同一である複数の状態を集約した汎用的な条件部と、最適な行動を行動部にもつ分類子を学習することができる。このような分類子を正確に一般化された分類子と呼ぶ。しかし、正確に一般化された分類子には、1) 報酬獲得に関与する分類子 (有効な分類子) だけでなく、2) 報酬獲得に関与しない分類子 (無効な分類子) も含まれているが、XCS ではこのような分類子の有効性を識別する機構は存在しない。このため、XCS では学習初段階において多種類の分類子を生成する必要がある、記憶する分類子の上限数 N を適切に設定する必要がある。また、行動選択や親個体選択では適合度に依存して選択されるために、雑音環境下では正確性が正しく見積もられないため学習性能が著しく低下する。

そこで、本論文では分類子の有効性を識別する個別化 (identification) を提案する。個別化により削除対象、親個体対象を適切に設定することにより効率の良い進化を促進でき、さらに適合度に依存せず識別することで雑音環境下においても適切に学習を行なうことができる。

なお本論文では、研究の第一段階として、特にシングルス
ステップ問題⁵⁾を対象とした個別化に焦点ををしぼる。シング
ルスステップ問題とは1ステップで報酬が与えられ問題が終了す
る問題で、データマイニングやクラスタリング問題に適用す
るために研究が行なわれている^{8), 9)}。

3.2 個別化手法

シングルステップ問題では、1ステップで報酬が与えられ

Fig. 2 Overview of IXCS

るため、報酬の有無により分類子の有効性を見積もることができる。具体的には報酬が与えられると有効な分類子であり、与えられなければ無効な分類子として識別する。従来の XCS のメカニズムに個別化を加えることで、正確に一般化された分類子の中でも、有効な分類子についてのみ学習することが可能となる。

4. 個別化による学習分類子システム

4.1 概要

Fig. 2 に個別化による学習分類子システム (Identification based XCS: IXCS) の学習過程を示す。IXCS は XCS に個別化を加えたシステムである。IXCS は行動実行時に環境から与えられる報酬により個別化を行ない、分類子の有効性は行動実行時に形成した行動集合に対して評価する。さらに、IXCS では照合集合を形成し行動集合を形成しなかった分類子で形成される非行動集合 (not action set: $\bar{A}$) を形成し、非行動集合についても分類子の有効性を見積もる。これにより、常に削除対象には無効な分類子を設定し親個体選択には有効な分類子が設定される。以下、IXCS のメカニズムについて述べる。

4.2 メカニズム

4.2.1 実行部

実行部において、行動集合の形成まで XCS と同様の過程を行なうが、行動集合形成時に行動集合に属さない分類子により非行動集合を形成する。行動実行後、報酬が与えられた場合は行動集合を形成する分類子を有効な分類子と識別し、非行動集合を形成する分類子を無効な分類子としてみなす。逆に、報酬が与えられなかった場合は無効な分類子と識別し、非行動集合内の分類子は有効な分類子としてみなす。

4.2.2 強化部

強化部においても分類子の各評価値は XCS と同様の更新を行なうが, IXCS での予測値の更新は (8) 式に示すように, 適合度に関する勾配を反映した更新式を用いる. (8) 式は正確に一般化された分類子に対して更新の影響を強め, 正確に一般化されてない分類子の過剰な更新を抑制できることが示されている¹⁰⁾.

```

1: state ← recognizing an input from environment
2: generate [M]
3: generate P(A)
4: action ← selecting the action from P(A)
5: generate [A]
6: generate [Ā]
7: reward ← execute action
8: parameter update
9: if(reward > 0)
10:   if(average elapsed time > θGA)
       run GA ← selecting the parents from [A]
11:   delete from [Ā]
12: else
13:   if(average elapsed time > θGA)
       run GA ← selecting the parents from [Ā]
14:   delete from [A]

```

Fig. 3 Algorithm of IXCS

$$p_j \leftarrow p_j + \beta(P - p_j) \times \frac{F_j}{\sum_{cl_k \in [A]} F_k} \quad (8)$$

4.2.3 発見部

発見部では、実行部の個別化の評価に基づいて GA における削除・進化対象を設定する。具体的には、有効な分類子を進化対象として親個体に設定し、無効な分類子を削除対象として設定する。また XCS と同様に、親対象に設定された有効な分類子である各分類子について、前回 GA 発動時から経過した時間の平均値を取り、その平均値がパラメータ θ_{GA} の値を上回る場合に発動される。XCS において削除は分類子集団から削除される分類子が選択されるのに対して、IXCS では選択対象を限定することで無効な分類子が優先的に削除される。また、XCS と同様に分類子集団中の分類子数がパラメータ N を超えた数だけ削除される。 N を超えないときは削除されない。

4.3 アルゴリズム

上記のようなメカニズムを備えた IXCS のアルゴリズムを Fig. 3 に示す。シングルスステップにおいては 1 ステップで問題が終了されるため、 $[A]_{-1}$ は形成されず、 $[A]$ に対してメカニズムが適用される。

5. 実験

5.1 実験内容

IXCS の学習性能を検証するため、LCS のシングルスステップ問題における評価として一般的なベンチマーク問題である Multiplexer 問題⁵⁾、中でも 20-Multiplexer 問題¹¹⁾を用いて従来手法(XCSTS)と比較し、つぎの2つの実験を行なう。具体的には、XCSTSに個別化を加えた提案手法であるIXCSTSとXCSTSについて比較を行なう。なお、XCSTSとは発見部のGAの親個体選択において、行動集合から確率 τ で分類子を選出し、適合度の大きい分類子をトーナメント形式で選択(tournament selection)する機構を備え、学習性能および

雑音耐性が向上することが示されている¹¹⁾。各実験で学習における行動選択は、環境のもつ状態空間を十分に探索するために予測配列を考慮しないランダム選択とする。

・実験1:IXCSTSの雑音環境下における学習性能を分析する。実験1で用いる雑音はalternating noise¹¹⁾およびgaussian noise¹¹⁾を用いる。alternating noiseとは確率 P_X で与える報酬が入れ替わる雑音である。具体的には、報酬 r を与えときに0を与え、同様に0を与えときに r を与える。実験1では雑音なし、 $P_X = 0.1, 0.2$ と変化させる。さらにgaussian noiseを用いた雑音は、本来与えられる報酬 r に対して、分散 σ のガウス分布に基づいて与えられる値が r に付加される雑音である。実験1では $\sigma = 300, 400, 500$ と変化させる。
・実験2:記憶する分類子の上限数 N を変化させたときの学習性能を分析する。実験2では、記憶する分類子の上限数を $N = 800, 600, 400$ と変化させる。

5.2 Multiplexer 問題

Multiplexer 問題は、入力と出力との間の高い非線形性と、入力の適切な一般化により入出力関係を縮約できる構造をもつため、学習分類子システムの一般化能力の評価に用いられる。Multiplexer 問題で与えられる入力であるビット長 l は $k + 2^k$ であり、 k は任意の整数である。また、出力はビット列 $b_0b_1 \dots b_{l-1}$ の最初から k ビット($b_0 \sim b_{k-1}$)を10進数に変換した値 d をアドレスとして、 b_{k+d} ビットの値となる。具体的には、 $k = 2$ としたときに与えられる入力111001のアドレスは $b_0b_1 \rightarrow 11$ を変換し $d = 3$ と定まり、 b_{2+3} のビット値である1が出力となる。また、本論文では $k = 4$ とした20-Multiplexer 問題を用いて、任意の20ビットの状態(入力)に対して正しい行動(出力)を実行した場合は報酬として $r = 1000$ を与え、間違った場合は0を与える。

5.3 評価方法とパラメータ設定

評価方法としては、1回の学習終了ごとに新たに与えられた異なる入力の20-Multiplexer 問題について、予測配列中の最大予測値を示す行動を選択(greedy 選択)し解答する。また、評価基準として、1) 正答率(performance)、2) マクロ分類子の総数(population size)の2点より評価する。正答率は高い値であるほど、正確に学習できていることを示し、マクロ分類子の総数は低い値であるほど少ない分類子数で学習できていることを示している。また、実験1では評価時に雑音により間違った行動が正答とならないように(間違った行動で正答率が上がることを防ぐために)、評価時には雑音を付加しない。学習回数は1試行につき300000回行ない、各基準は学習回数5000回ごとの移動平均で示し、試行数20回の平均をとる。

各実験で用いるXCSTSのパラメータの設定はつぎのとおりである¹²⁾。 $N = 2000$, $\nu = 5$, $\beta = 0.2$, $\epsilon_0 = 0.001$, $\theta_{GA} = 25$, $\chi = 1.0$, $\mu = 0.04$, $P_{\#} = 0.6$ 。IXCSTSは有効な分類子を早く識別することが重要であるため、正確性を反映されやすく設定し、 $\#(\text{don't care})$ の発生確率 $P_{\#}$ を高く設定する。また、条件部と行動部の組み合わせが正しい有

Fig. 4 Performance of IXCS and XCS when adding alternating noise

Fig. 6 Performance of IXCS and XCS when adding gaussian noise

Fig. 5 Population sizes of IXCS and XCS when adding alternating noise

Fig. 7 Population sizes of IXCS and XCS when adding gaussian noise

効な分類子は、突然変異で条件部を頻繁に変更するべきでない。以上より、IXCSTS では $\epsilon_0 = 1$, $P_{\#} = 0.8$, $\mu = 0.01$ とし、その他のパラメータは XCSTS と同値とする。

5.4 実験 1 の結果

Fig. 4 および Fig. 5 に alternating noise の実験結果、Fig. 6 および Fig. 7 に gaussian noise の実験結果を示す。Fig. 4, 6 の縦軸は正答率 (performance), Fig. 5, 7 の縦軸はマクロ分類子の総数 (population size), 各図の横軸は学習回数 (iterations) を表す。

Fig. 4 より、雑音なし ($P_X = 0.0$) の場合では記憶できる分類子数 ($N = 2000$) が十分に設定されているために XCSTS の学習性能が良いが、IXCSTS は XCSTS よりも正答率が若干低く、また学習速度も遅い。しかし、雑音付加時では XCSTS の正答率と学習速度が低下しているが、IXCSTS の正答率は XCSTS よりも高く、学習速度も速くなっている。特に、 $P_X = 0.2$ の場合では XCSTS は学習不可能であるが、IXCSTS では高い学習性能を実現している。また、Fig. 5 より、雑音付加時での XCSTS はマクロ分類子の総数が減少していないが、IXCSTS ではマクロ分類子の総数が減少しており、適切に一般化が行なわれていることがわかる。

Fig. 6 より、 $\sigma = 300$ では IXCSTS は XCSTS よりも正答

率が低く、また学習速度も遅い。しかし、 $\sigma = 400, 500$ では XCSTS の正答率は低く、または学習不可能となっているが、IXCSTS では高い学習性能を実現している。また、Fig. 7 より、雑音付加時での XCSTS はマクロ分類子の総数が減少せず、一般化が行なわれていないが、IXCSTS は雑音付加時においても、マクロ分類子の総数が減少し、適切に一般化されていることがわかる。以上より、IXCSTS は雑音影響下においても IXCSTS のメカニズムが適切に動作し、安定した学習性能を実現していることがわかる。

5.5 実験 2 の結果

Fig. 8 および Fig. 9 に実験結果を示す。Fig. 8 において縦軸は正答率 (performance), Fig. 9 において縦軸はマクロ分類子の総数 (population size), 各図の横軸は学習回数 (iterations) を表す。

Fig. 8 より、分類子の上限数を減少させた場合では IXCSTS は XCSTS よりも、学習速度が速く、正答率が高い。特に、 $N = 600, 400$ の場合では XCSTS が学習不可能であるのに対し、IXCSTS では高い正答率を実現している。Fig. 9 より、XCSTS では $N = 800$ においてもマクロ分類子の総数の減少には限界があるが、IXCSTS では学習初段階でマクロ分類子は集約され適切に包摂が実行されていることがわかる。特に、

Fig. 8 Performance of IXCS and XCS when setting lower max population size

Fig. 9 Population sizes of IXCS and XCS when setting lower max population size

XCSTS では N が 600 以下になると、分類子は集約されず包摂が実行されていないが、IXCSTS ではマクロ分類子の総数が減少し適切に一般化が行なわれている。以上より、IXCSTS は記憶する分類子の上限数を減少させた場合でも IXCSTS のメカニズムが適切に動作し、安定した学習性能を実現していることがわかる。

6. 考 察

6.1 IXCSTS の雑音耐性に関する考察

本論文では IXCSTS の雑音耐性を検証するために、alternating noise と gaussian noise の 2 種類の雑音について学習性能を分析した。alternating noise は与える報酬が入れ替わる雑音であり、雑音の発生確率だけ識別精度が低下するが、gaussian noise は常時雑音が付加され不定の報酬が与えられるため、報酬の有無に基づいて識別する個別化手法では識別精度が低下する。具体的には、本来の報酬が 0 であるときに雑音付加により誤って報酬が与えられる場合に、間違った識別が発生する。したがって、IXCSTS は alternating noise 付加時よりも gaussian noise 付加時の場合が学習速度が遅く、正答率も低下した。

しかし、XCSTS は両雑音の影響をうけ適合度が正しく見

積もられないために学習性能が IXCSTS よりも大幅に低下している。これは XCSTS が分類子の有効性を考慮せず進化対象に設定するため、雑音の影響をうけ適合度が低下した有効な分類子についても削除対象に設定されるが、IXCSTS は適合度に依存せず分類子の有効性に応じて削除対象と進化対象に設定するため、雑音の影響を軽減している。

6.2 IXCSTS の学習性能に関する考察

IXCSTS の学習性能を明確にするため、学習中に出現した 20-Multiplexer 問題における最も一般化された分類子 (optimal classifier) の出現率 (rate of acquired optimal classifiers) およびその重合度の平均 (average of numerosity in optimal classifiers) を用いて考察する。20-Multiplexer 問題 ($k = 4$) における最も一般化された分類子とは、分類子の条件部のビット列の $b_0 \sim b_{k-1}$ が 0 もしくは 1 であり、アドレス d が指し示す b_{k+d} ビットの出力が分類子の示す行動と一致し、かつ、それ以外のビット値はすべて # であるような分類子を指す。たとえば、条件部: 0001#1##### で行動部が 1 であると最も一般化された分類子であり、このような最も一般化された分類子の組み合わせは 32 通り存在する。検証内容としては実験 1 および実験 2 の実験設定で、最も一般化された分類子の出現率およびその重合度の平均について評価する。

実験 1 と同様の実験設定で得られた検証結果を Fig. 10, Fig. 11 および Fig. 12, 実験 2 と同様の実験設定で得られた検証結果を Fig. 13 に示す。また, Fig. 10 は雑音なしの場合であり, Fig. 11 は alternating noise ($P_X = 0.2$) 付加時の場合, Fig. 12 は gaussian noise ($\sigma = 400$) 付加時の場合, Fig. 13 は雑音なし時の $N = 600$ としたときの場合である。各図において左の縦軸は獲得した optimal classifier の出現率 (rate of acquired optimal classifiers: occurrence), 右の縦軸は出現した optimal classifier の重合度の平均 (average of numerosity in optimal classifiers: average) であり, 各図の横軸は学習回数 (iterations) を表わす。optimal classifier の出現率は全 32 通りに対して出現した数の割合を示す。また, 図中の occurrence は optimal classifier の出現率, average は出現した optimal classifier の重合度の平均を示している。occurrence が高い値ほど, optimal classifier の出現率が高く, 適切な一般化が行なわれていることを示し, average が高い値ほど, 雑音影響下で誤って optimal classifier が削除された場合でも, optimal classifier が消滅しにくい度合を示している。

Fig. 10 から, XCSTS と IXCSTS は同等の出現率を保っているが, IXCSTS の重合度の平均は XCSTS より倍以上増えていることがわかる。この要因としては, IXCSTS は有効な分類子を進化対象に設定することで, 効果的な進化を促し, 結果として有効な分類子は optimal classifier に包摂され重合度が増大した。また, Fig. 11 および Fig. 12 より, XCSTS では alternating noise および gaussian noise が付加されることにより, 正確な分類子であっても適合度が低下するために誤っ

Fig. 10 Rate of acquired optimal classifiers and average of numerosity in optimal classifiers

Fig. 12 Rate of acquired optimal classifiers and average of numerosity in optimal classifiers when adding gaussian noise ($\sigma = 400$)

Fig. 11 Rate of acquired optimal classifiers and average of numerosity in optimal classifiers when adding alternating noise ($P_X = 0.2$)

Fig. 13 Rate of acquired optimal classifiers and average of numerosity in optimal classifiers when setting lower max population size ($N = 600$)

て削除されるのに対して、IXCSTSは雑音付加時においても出現率と重合度の平均はXCSTSより高く、適切に有効な分類子を探し、これらの分類子を学習強化している。重合度が低ければ分類子は消滅しやすくなり、XCSTSでは分類子の有効性に依存することなく学習が行なわれるため、有効な分類子の重合度は上昇せずに雑音の影響を受け消滅した。一方で、IXCSTSでは有効な分類子を認識し進化対象に設定することで、有効な分類子の重合度は高く、雑音により誤って削除された場合でも消滅することがない。したがって、IXCSTSは雑音耐性に優れている。さらにFig. 13より、分類子上限数を低く設定した場合でも、IXCSTSはoptimal classifierの出現率および重合度の平均がXCSTSより高く、分類子の上限数が小さくても学習できることがわかる。これは、分類子の上限数が小さくても無効な分類子が削除され、有効な分類子の重合度が十分大きくなりoptimal classifierの出現率が上がったためである。

6.3 IXCSTSの位置づけ

Fig. 14に20-Multiplexer問題におけるIXCSTSの位置づけを示す。XCSは、20-Multiplexer問題において、適切な

上限数を設定した場合に高い正答率を実現することが示されている¹¹⁾が、XCSTSは雑音付加時においても適切に学習し、XCSよりも雑音耐性に優れていることが示されている¹¹⁾。しかしXCSTSは複雑な雑音に対しては学習不可能となり、問題に対して十分な記憶する分類子の上限数を定める必要があった。このようにXCSTSをはじめとするXCSは、適合度に依存して進化対象や削除対象を設定するため、雑音がない場合では適合度が正しく見積もられ高い学習性能を実現する。しかし、雑音を付加した場合のalternating noise ($P_X = 0.1$), gaussian noise ($\sigma = 300$)では、学習性能が保たれているが、 $P_X = 0.2$, $\sigma = 400$, 500設定時では、急激に学習性能が低下し雑音耐性が低い。さらに、記憶する分類子の上限数を減少させた場合では、正確に一般化された分類子の効率のよい探索が行なわれないために、学習性能が低下する。したがって、XCSやXCSTSは上記のような限定的な環境に対してのみ学習可能であるのに対し、IXCSTSは、雑音を付加した場合もしくは分類子上限数を変更した場合でも、安定した学習性能を実現し、XCSTSが学習できないような環境に対して

Fig. 14 Position of XCSs on noise robustness and max population size

も学習可能である。

6.4 IXCSTS のシングルステップ多行動問題への拡張

本論文ではシングルステップ問題として 2 種類の行動をもつ 20-Multiplexer 問題に適用し IXCSTS の学習性能を分析した。2 種類の行動である場合は、報酬の有無により一方が正しい行動であり、もう一方が間違った行動と一意に定まる。したがって、分類子の有効性の識別精度は高く、IXCSTS は有効な分類子が進化対象となり学習性能が向上した。一方で、多種類の行動をもつ問題では、報酬の有無による識別方法では、分類子の有効性が一意に定まらない。具体的には、行動集合を形成する分類子が有効な分類子である場合は、非行動集合を形成する分類子は無効な分類子のみで形成するため、正しい識別が行なわれるが、行動集合を形成する分類子が無効な分類子である場合は、非行動集合を形成する分類子は、有効な分類子と無効な分類子が混在するため、必ずしも有効な分類子が進化対象に設定されない。さらに、XCS で用いられる行動選択であるランダム選択では、分類子の有効性に依存せずに行動を選択するため、誤った識別が発生しやすい。したがって、多種類の行動をもつ問題では識別精度が低下し、IXCSTS の学習性能は低下する。

多種類の行動をもつ問題においても、識別精度を向上させるためには、行動選択時に正しい行動を選択し、行動集合が有効な分類子で形成される割合を増やすことが重要である。そのため、行動選択では、確率 ϵ でランダム選択を行ない、それ以外では予測配列中の最大予測値をもつ行動を選択する ϵ -greedy 選択を用いることで、正しい行動をもつ分類子を選択する割合が増え、識別精度を向上させることを考える。以上のような効果を検証するために、シングルステップ多行動問題に、 ϵ -greedy 選択を用いた IXCSTS を適用し、実験する。

用いる問題は Hidden Parity 問題¹³⁾を拡張した問題で、0, 1 で構成されるビット列の入力に対して先頭ビットから n ビットまでの 1 の出現数に依存して、最大 n の回答数が設定できる問題である。ここでは $n = 4$ として 4 種類の行動を設定する。また、実験 1 で用いたパラメータ設定と整合性をとるために、20-Multiplexer 問題と同じ状態空間の大きさをもつように、入力ビット列のビット長は 20 と設定する。与えられる出力としては、先頭ビットから n までに出現した 1 の出現数と行動数の剰余の値が出力となる。たとえば、入力：00110110101001110001 に対して、先頭ビットから 5 ビット

Fig. 15 Performance of IXCS and XCS when adding alternating noise (ϵ -greedy selection)

Fig. 16 Population sizes of IXCS and XCS when adding alternating noise (ϵ -greedy selection)

(00110) までの 1 の出現数は 2 であり、 $2\%4 = 2$ が出力となる。この入力に対し、行動選択において行動 2 を選択し実行した場合、報酬 $r = 1000$ が与えられ、それ以外は 0 となる。IXCSTS と XCS は行動選択を ϵ -greedy 選択とし、 $\epsilon = 0.5$ 、また $P_{\#} = 0.9$ とし、実験では、実験 1, 2 と同様の実験設定と評価方法を用いる。

実験 1 と同様の実験設定で得られた検証結果を Fig. 15, 16, 17, 18 に示し、実験 2 と同様の実験設定で得られた検証結果を Fig. 19, 20 に示す。また、Fig. 15, 16 は雑音なし ($P_X = 0.0$) と alternating noise 付加時の場合であり、Fig. 17, 18 は gaussian noise 付加時の場合、Fig. 19, 20 は記憶する分類子の上限数を変更した場合で雑音は付加していない。Fig. 15, 17, 19 において縦軸は正答率 (performance)、Fig. 16, 18, 20 において縦軸はマクロ分類子の総数 (population size)、各図の横軸は学習回数 (iterations) を表わす。

Fig. 15, 17 より、 ϵ -greedy 選択に変更した IXCSTS は XCS よりも学習速度が低下しているが、正答率が高く、Fig. 16, 18 より、マクロ分類子の総数が減少し、適切に一般化が行なわれている。さらに Fig. 19 では、XCS と同等の学習速度で高い正答率を示しており、Fig. 20 より、マクロ分類子の総数が減少し、適切に一般化が行なわれている。これは、 ϵ -greedy

Fig. 17 Performance of IXCS and XCS when adding gaussian noise (ϵ -greedy selection)

Fig. 18 Population sizes of IXCS and XCS when adding gaussian noise (ϵ -greedy selection)

選択により IXCSTS は、分類子の有効性の識別精度が向上することで有効な分類子を進化対象に設定し、XCSTS のメカニズムが適切に動作したためである。以上より、シングルステップ多行動問題においても、 ϵ -greedy 選択を用いて識別精度を向上させることで IXCSTS は XCSTS と比較し、雑音耐性に優れ、学習性能が高いことがわかる。

7. 結 論

本論文では、XCS が分類子の有効性を認識せず無効な分類子についても学習進化させることで記憶する分類子が増大し、雑音により学習性能が低下する問題について、分類子の有効性を識別し、その有効性に応じて適切な処理を行なう個別化を提案し、XCS に個別化を組み込んだ個別化による学習分類子システムを提案した。20-Multiplexer 問題を用いて提案手法の有効性を検証し、従来手法 (XCSTS) と比較することでつぎの知見を得た。1) 雑音付加時においても安定した学習性能を保ち、2) 記憶する分類子の上限数を減少させた場合でも、効果的な分類子を学習し、安定した学習性能を保つことを示した。

今後は以下の課題に取り組む予定である。1) 雑音がない場合において XCSTS が実現する学習性能を達成するため、個別化手法を改善する。2) マルチステップ問題に適応したシス

Fig. 19 Performance of IXCS and XCS when setting lower max population size (ϵ -greedy selection)

Fig. 20 Population sizes of IXCS and XCS when setting lower max population size (ϵ -greedy selection)

テムの構築を目指し、より複雑な環境下においても適切に有効な分類子を識別する個別化手法を拡張することである。

参 考 文 献

- 1) J.H. Holland: Escaping brittleness: the Possibilities of General-purpose Learning Algorithms Applied to Parallel rule-based systems, in R. Michalski, J. Carbonell and T. Mitchell, Eds., Machine Learning: An Artificial Intelligence Approach, **2**, 593/623, Morgan Kaufmann (1986)
- 2) J.C.J. Watkins: Learning from Delayed Rewards, PhD thesis, Cambridge University (1989)
- 3) R.S. Sutton: Learning to Predict by the Methods of Temporal Differences, Machine Learning, **3**-1, 9/44 (1988)
- 4) D.E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley (1989)
- 5) S.W. Wilson: Classifier fitness based on accuracy, Evolutionary Computation, **3**-2, 149/175 (1995)
- 6) 和田, 高玉, 下原, 片井: 実数値学習分類子システムの分析, 人工知能学会論文誌, **20**-1, 57/66 (2005)
- 7) M.V. Butz and S.W. Wilson: An Algorithmic Description of XCS, Soft Computing, **6**-3, 144/153 (2001)
- 8) H.H. Dam, H.A. Abbass and C. Lokan: DXCS: An XCS System for Distributed Data Mining, GECCO2005, 1883/1890 (2005)
- 9) K. Tamee, L. Bull and O. Pinngern: Towards clustering with XCS, In Proceeding of Genetic and Evolutionary Computation Conference, 1854/1860 (2007)

- 10) M.V. Butz, D.E. Goldberg and P.L. Lanzi: Gradient descent methods in learning classifier systems: improving XCS performance in multistep problems, *Evolutionary Computation*, **9**-5, 452/473 (2005)
- 11) M.V. Butz, K. Sastry and D.E. Goldberg: Tournament Selection in XCS, *GECCO2003*, LNCS 2724, 1857/1869 (2003)
- 12) M.V. Butz: *Rule-Based Evolutionary Online Learning Systems*, Springer (2006)
- 13) T. Kovacs: Deletion schemes for classifier systems, *GECCO1999*, 329/336 (1999)

=====

[著 者 紹 介]

中 田 雅 也 (学生会員)

1988 年生。2011 年電気通信大学電気通信学部人間コミュニケーション学科卒業。同年、同大学大学院情報理工学研究科総合情報学専攻修士課程入学、現在に至る。強化学習、進化計算、学習分類子システムの研究に従事。

原 田 智 広

1987 年生。2010 年電気通信大学電気通信学部人間コミュニケーション学科卒業。同年、同大学大学院情報理工学研究科総合情報学専攻修士課程入学、現在に至る。進化計算、宇宙小型衛星開発の研究に従事。

佐 藤 圭 二

1983 年生。2009 年電気通信大学大学院電気通信学研究科人間コミュニケーション学専攻修士課程修了。修士 (工学)。2010 年同大学大学院情報理工学研究科総合情報学専攻博士課程入学、現在に至る。学習分類子システム、多目的最適化、路線最適化の研究に従事。人工知能学会会員。

松 島 裕 康 (正会員)

1982 年生。2010 年電気通信大学大学院電気通信学研究科人間コミュニケーション学専攻修士課程を修了。修士 (工学)。同年、同大学情報理工学研究科総合情報学専攻博士後期課程に進学。学習分類子システム、マルチエージェントシステム、遺伝的アルゴリズム、宇宙小型衛星開発の研究に従事。人工知能学会会員。日本学術振興会特別研究員 DC。

高 玉 圭 樹 (正会員)

1970 年生。98 年東京大学大学院工学系研究科博士課程修了。博士 (工学)。同年、国際電気通信基礎技術研究所 (ATR) 入所。2002 年東京工業大学大学院総合理工学研究科講師を経て、2006 年電気通信大学電気通信学部助教授、2007 年准教授、2011 年教授、現在に至る。マルチエージェントシステム、強化学習、創発的計算手法の研究に従事。著書に「マルチエージェント学習—相互作用の謎に迫る—」など。IEEE, 人工知能学会, 情報処理学会, 日本ロボット学会, 電子情報通信学会, 各学会会員。

=====