

適応概念学習アルゴリズム：RF4

齊藤和巳[†] 中野良平[†]

本論文では、問題を解くにつれて、自ら適応して探索効率を改善する概念学習法 RF4 について報告する。RF4 は、望ましくない論理式を枝刈りする学習バイアス、および、存在限量子や集約関数を含む論理式を生成する複合化ルールを用いて、事例の識別概念を深き優先で探索する。論理式の探索順序は、過去に解いた問題を基に、それが識別概念の構成要素となる確率を推定することにより、動的に決定される。KRK チェス終盤戦問題では、ランダムに選んだ事例群の学習を数回繰り返せば、RF4 の探索効率が改善されるだけでなく、未知の事例に対する正答率も向上することを確認した。図形の多彩な識別概念を求めるボンガルド問題では、図形オブジェクトとそれらの関係についての基本的な知識ベースを与えたところ、100 のボンガルド問題に対して RF4 は 41 問を正答できた。一方、代表的な概念学習アルゴリズム、GOLEM, INDUCE, FOIL では、いずれも 20 数問程度しか正答できなかった。RF4 の適応機能の評価では、問題を解くにつれて、推定確率の信頼性が高くなるので、ボンガルド問題を解くための平均時間が次第に短縮されることを確認した。特に、41 問すべてを解答した後、再び同じ問題を解くための平均時間は約 1/3 に短縮された。また、統計検定を用いて、RF4 が用いた確率集合から、問題解決を高速化する知識の一部を検出できた。

Adaptive Concept Learning Algorithm: RF4

KAZUMI SAITO[†] and RYOHEI NAKANO[†]

This paper describes a concept learning algorithm called RF4 which adaptively improves its concept learning efficiency. RF4 performs a depth-first search on the basis of five criteria for pruning undesirable formulae and five transformation rules for combining formulae. The search order of formulae is dynamically determined by estimating the probability that the formula is a component of the concept. In a KRK chess end-game problem, after learning a few sets of training examples, RF4 quickly improved its learning efficiency as well as predictive accuracy. In a set of visual pattern recognition problems called Bongard problems, using primitive knowledge of graphical objects, RF4 solved 41 out of 100 problems efficiently; other learning algorithms, e. g. GOLEM, INDUCE, or FOIL solved much less. After solving the 41 training problems, the average time for RF4 to solve the same set of problems was reduced to about one-third. Through statistical tests, a piece of useful heuristic knowledge to improve the problem-solving efficiency was extracted from a set of probabilities used in RF4.

1. はじめに

知能の基本的な特徴の 1 つは、与えられた事例（問題）集合から、実質的に新たな知識を学習（発見）できることである。その知識には、新たな事例を適切に識別するための概念や、問題を高速に解決するための手続きなどがある。

概念学習は、与えられた論理空間において、適切な論理（識別）式を探索する問題として定式化できる⁹⁾。代表的な概念学習アルゴリズム^{5),12),13)}では、正答率や簡潔性などに基づいた 1 つのヒューリスティック評価関数に基づいて、評価値の高い論理式を選択し、それらの組合せにより識別概念を探索する。しかし、この

戦略では、探索効率は良いが、評価値が十分に高くない論理式の組合せからなる識別概念を見逃す危険がある。一般に、このような状況は少なからず起こるので、ある論理式を枝刈る積極的な根拠がなければ、その論理式を概念の探索に用いるべきである。よって、多数の枝刈り尺度を採用した深き優先探索は、概念学習において有望なアプローチであると考えられる。

多くの概念学習アルゴリズムでは、同じ問題が繰り返し与えられても、常に同じ処理ステップを実行し、過去の問題解決の経験に基づいて、概念学習効率を改善することはない。問題解決（概念学習）の高速化には、問題解決に有効なルールの組合せであるマクロオペレータ（チャンク）を学習する EBL^{4),10)} アプローチが広く研究されているが、1 つの事例から作成したマクロオペレータの一般的な有用性に関する utility 関

[†] NTT コミュニケーション科学研究所
NTT Communication Science Laboratories

題⁹⁾が課題として指摘されている。ここでは、問題集合からヒューリスティック評価関数を学習するアプローチ¹⁴⁾を採用する。この技術は、主として、ゲーム探索の分野だけで利用されているので、より広い応用においての検討が望まれる。

本論文では、過去に解いた問題を用いて、自ら適応して探索効率を改善する概念学習法 RF 4 について報告する。2章では、概念学習の枠組みについて説明する。3章では、RF 4 アルゴリズムの詳細について述べる。4章では、RF 4 のチェスの終盤戦問題への適用結果、5章では、ボンガルド問題への適用結果について報告する。

2. フレームワーク

事例を識別する概念は、論理式で表現されるとする。また、各事例は、未知の概念（論理式）によりクラス 1（正の事例）かクラス 2（負の事例）に分類されるとする。概念学習とは、分類された事例集合から、その識別概念を求める問題である。

論理式の表現には、記述力と理解容易性を考慮して、集約関数（aggregate function）を含む一階述語論理（first-order predicate calculus）を採用する。以下に、論理式の再帰的な定義を示す。ターム（term）は、定数/変数であるか、または、論理式に集約関数（count, sum, average, max, min）を施したものである。原子式（atomic formula）は、2つのタームに比較演算子（ $>$, \geq , $<$, \leq ）を施したものである。論理式は、原子式であるか、または、論理式に論理結合（ \wedge , \vee ）や限量子（ \forall , \exists ）を施したものである。

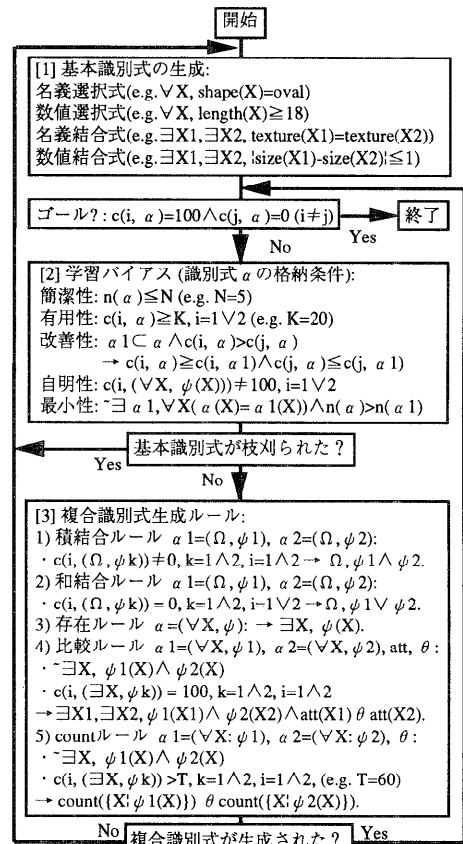
3. RF4 アルゴリズム

RF 4 は、望ましくない論理式を枝刈りする学習バイアス、および、存在限量子や集約関数を含む論理式を生成する複合化ルールを用いて、事例の識別概念を深き優先で探索する。論理式の探索順序は、過去の経験に基づいて、原子式が識別概念の構成要素となる確率を推定することにより、動的に決定される。

3.1 基本機能

3.1.1 概要

図 1 に、RF 4 の基本処理フローチャートを示す。基本フローは、以下の処理を繰り返す。まず、入力記述より基本識別式を生成する。もし、その識別式がゴール条件を満たさず、学習バイアスにより枝刈られるならば、別の基本識別式を生成する。さもなければ、その識別式を格納するとともに、既に格納されている識別式を用いて複合識別式を生成する。ここで、新たに



注) 変数Xはすべて $\forall F \in \text{facts}, X \in F$ で規定; att: 属性名;
 θ : 比較演算子; F, X: 変数; Ω : 限量子; ψ : 条件; a : 論理式;
 $c(i, a)$: a のクラス i でのカバー率; $n(\psi)$: 条件の原子式数;
 $c(i, a k)=0, k=1 \wedge 2, i=1 \vee 2$
 $\equiv (c(1, a1)=0 \wedge c(1, a2)=0) \vee (c(2, a1)=0 \wedge c(2, a2)=0)$.

図 1 RF 4 の基本フロー
 Fig.1 General flow of RF 4.

生成された識別式が優先して処理される。もし、生成した識別式がゴール条件を満たさず、学習バイアスにより枝刈られるならば、別の複合識別式を生成する。もし、新たな複合識別式を生成できなければ、別の基本識別式を生成する。また、新たな基本識別式も生成できなければ、解答不能としてアルゴリズムが終了する。ここで、カバー率 $c(i, a)$ は、クラス i で論理式 a にカバーされる事例数とクラス i の事例数の比である。

3.1.2 基本識別式の生成

3.1.1 項で述べたように事例から直接生成される最も単純な識別式として、4 種の基本識別式を生成する。図 1-[1] に、各識別式の例を示す。名義属性からは、属性とその値の対より、名義選択式を生成する。数値属性に対しては、簡単なクラスターリングを施す。すなわち、まず、数値をソートして昇順に配置し、隣合う値のギ

ギャップが最大となる箇所を“以上”，“以下”の2つの識別式を生成する。例えば，属性値が $\text{att}(X) = \{8, 9, 10, 12, 14, 18, 19, 21, 23, 25\}$ であるとき，結果の2つの条件は $\text{att}(X) \leq 16$, $\text{att}(X) > 16$ となる。また，この手続きは，さらに別の条件を得るため，分割した値の集合に対して再帰的に適用できる。RF 4では，最も大きい2つのギャップを選択し，条件を生成する。

複数のオブジェクトがあるときには，2つのオブジェクトの属性値を比較するケースが考えられる。各義属性からは，存在限量子を用いて，名義結合式を生成する。数値属性に対しては，2つの値の差にクラスタリングを施して，数値結合式を生成する。

3.1.3 学習バイアス

新たに生成される識別式に対して，5種の学習バイアス（簡潔性，有用性，改善性，自明性，最小性）を用いて，不要な識別式を枝刈りする。図1-[2]に，各バイアスの定義を示す。これらは，複合式に含まれる原子式数が一定値（e.g. $N=5$ ）を越えないこと，識別式のカバー率が一定値（e.g. $K=20\%$ ）を越えること，複合化前の識別式より複合化後の識別式のカバー率が改善されること，識別式が全てのオブジェクトをカバーしないこと，および，同じ内容を複雑な識別式で表現しないことを要求する。なお，パラメタ値の設定については，ユーザが許容する識別概念の複雑さの上限を N とし， N 個の原子式から成る識別式を考慮して，1個あたり少くとも $K \approx 100/N$ を目安にする。また，一般に， N が大きく， K が小さければ，複数の識別式の生成も可能となるが，多くの学習時間が必要になる。

3.1.4 複合識別式の生成

図1-[3]に，各複合化ルールの定義を示す。識別式が2つのクラスのオブジェクトをカバーするときには，論理積結合を用いて，識別式を特殊化する。逆に，識別式が特殊すぎるときには，論理和結合を用いて，識別式を一般化する。ただし，RF 4では，一方のクラスのどの事例もカバーしない論理式のペアを和結合し，積和標準形（DNF：Disjunctive Normal Form）として，識別概念を生成する。また，全称限量子を存在限量子に置き換えて，識別式の条件を弱めるルールも用意する。比較ルールでは，異なるオブジェクトの属性値を比較するので，それらのオブジェクトがすべての事例に存在することが条件となる。count ルールでは，異なる2種の識別式を満足するオブジェクトが存在する場合，それらを条件として count したオブジェクト数を比較する。ただし，count の場合には，事例にオブジェクトが存在しないときにも0と値が定まるので，閾値 T を導入する。RF 4では，集約関数として

count だけを扱うが，他の関数の実現は簡単な拡張で行える。

3.2 適応機能

識別概念は原子式の組合せであり，探索効率の点では，どのような順番で原子式の組合せを探索するかが，非常に重要な問題となる。その順番は問題ごとに異なると考え，問題の特徴を表現する問題特徴 f を導入する。また，その順番はこれまでの探索でどの原子式を使って来たかにも依存するので，それを探索状態 r として管理する。詳細には， f の各要素は，その問題において，予め定義したある論理式が真となるかどうかを示すブール値であり，一方， r の各要素は，1対1で1つの原子式に対応し，現時点の探索に利用されているかどうかを示すブール値である。以下では， f の要素と r の要素をそのまま並べた1つのベクトルを考え，状況ベクトル $s = (f, r)$ と呼ぶ。

ある状況 s において，現時点の探索で利用されていない各原子式が識別概念の構成要素となる条件付き確率が分かるならば，その確率が最大となる原子式を次に用いて探索を実行することを繰返すことにより，探索の期待コストを最小にすることができる。RF 4では，以下で述べるように，過去の問題解決の経験に基づいて，その近似確率を推定する。

3.2.1 $P(s)$ の2次近似

$s = (s_1, \dots, s_d)$ を状況ベクトル， $P(s)$ を状況ベクトル s の生起確率とする。まず，Bahadur-Lazarsfeld 展開²⁾の1次項だけで $P(s)$ を近似すれば，

$$P_1(s) = \prod_{i=1}^d p_i^{s_i} (1-p_i)^{1-s_i} \quad (1)$$

となる。ただし， $p_i = P(s_i = 1)$ である。この近似は， s_i の値が互いに独立に定まるときに他ならない。次に，2次項まで用いて $P(s)$ を近似すれば，

$$P_2(s) = P_1(s) \left(1 + \sum_{i=2}^d \sum_{j=1}^{i-1} (b_{ij} - p_i p_j) y_i y_j \right) \quad (2)$$

表1 観測事例と推定確率

Table 1 Observation and estimated probabilities.

観測事例	推定確率		
	(s_1, s_2, s_3)	$P_1(s)$	$P_2(s)$
(1, 0, 1) (0, 1, 0)	(0, 0, 0)	1/8	0
	(0, 0, 1)	1/8	0
	(1, 0, 0)	1/8	0
	(1, 0, 1)	1/8	1/2
	(0, 1, 0)	1/8	1/2
	(0, 1, 1)	1/8	0
	(1, 1, 0)	1/8	0
	(1, 1, 1)	1/8	0

となる。ただし、 $p_{ij} = P(s_i=1, s_j=1)$, $y_i = (s_i - p_i)/(p_i(1-p_i))$ である。 $P(\mathbf{s})$ を k 次項まで展開すれば、推定すべき確率の数は $O(n^k)$ となり、一般には多くの事例を得られないので、それらを高い信頼性で推定できなくなる。一方、1次近似を採用すれば、出現した事例の確率を十分に反映できない場合がある。例えば、表1に示すように、2つの3次元ベクトルが現れた後、1次近似を採用すれば、すべてのベクトルは同じ確率で出現すると推定されるが、2次近似を採用すれば、出現した2つのベクトルは他よりも高い確率で出現すると推定できる。つまり、RF4では、推定確率の大きい順に探索が行われるので、1次近似を採用すれば、その探索はランダムになるが、2次近似を採用すれば、確率の大きいものから探索できる。よって、RF4では2次近似を採用する。

3.2.2 p_i, p_{ij} の推定

2次近似の要素確率 p_i, p_{ij} の値は、最尤推定²⁾ に基づき、1つの問題の解決ごとに再推定される。つまり、これらの値は、状況ベクトル $\mathbf{s}=(\mathbf{f}, \mathbf{r})$ 要素の値が真となる問題数とこれまでに解いた問題数の比である。ここで、問題特徴 \mathbf{f} の各要素が真となるのは、対応する論理式が少なくとも一方のクラスの全事例をカバーするときとし、さもなければ、偽とする。一方、探索状態 \mathbf{r} の各要素が真となるのは、対応する原子式が識別概念の構成要素となるときとし、さもなければ、偽とする。

3.2.3 確率推定

現時点で、探索に利用していない原子式に対応する探索状態要素を s_k とすれば、値が既知の状況ベクトル要素群 $\{s_1, \dots, s_n\}$ に対して、 s_k が真となる（識別概念の構成要素となる）条件付き確率は：

$$P_2(s_k=1 | s_1, \dots, s_n) = p_k + \frac{\alpha^{(h)}}{\beta^{(h)}} \quad (3)$$

となる。ただし、

$$\alpha^{(h)} = \sum_{i=1}^h (p_{ik} - p_i p_k) y_i, \quad (4)$$

$$\beta^{(h)} = 1 + \sum_{i=1}^h \sum_{j=1}^{i-1} (p_{ij} - p_i p_j) y_i y_j \quad (5)$$

である。（(3)式の導出は付録参照）。RF4では、この条件付き確率を最大にする原子式を次に用いて探索を実行する。なお、(3)式の右辺は、 s_k に対応する原子式が識別概念の構成要素となる頻度 p_k と2次近似に基づく補正值 $\alpha^{(h)}/\beta^{(h)}$ を加えた値と解釈できる。

RF4では、探索の過程で、問題特徴要素の真理値を求める。すなわち、探索を開始する前には、問題特徴要素の値は未知であり、探索が進むにつれて、それらの値を次第に明らかにする。また、探索状態要素の値

も探索の過程で変化するので、RF4では、条件付き確率を再帰的に求める計算法を採用する。すなわち、状況ベクトル要素群 $\{s_1, \dots, s_{h-1}\}$ の値を既知とし、さらなる探索での状況ベクトル要素 s_h の値が確定したとすれば、 $\alpha^{(h)}, \beta^{(h)}$ の値は、次の漸化式を用いて効率良く計算できる。

$$\alpha^{(h)} = \alpha^{(h-1)} + (p_{hh} - p_h p_k) y_h, \quad (6)$$

$$\beta^{(h)} = \beta^{(h-1)} + \alpha^{(h-1)} y_h. \quad (7)$$

ただし、初期値は $\alpha^{(0)}=0, \beta^{(0)}=1$ である。

4. チェス終盤戦問題への適用

KRK問題¹¹⁾と呼ばれるチェス終盤戦の概念学習問題を用いて、RF4の性能を評価した。KRK問題とは、白のKingとRook対黒のKingの戦いであり、与えられた局面において、黒のKingが白のどちらかの駒で直接攻撃されるかどうかを判定する概念を求める問題である。可能な駒の配置は、 $64 \times 63 \times 62$ であり、このうち約1/3がこの概念を満たす配置となる。実験での問題表現には、行/列について、各駒のペアの位置が等しい/隣接/小さいかを示す6述語を利用した。ただし、KRK問題には1つの概念しか現れないので、問題特徴 \mathbf{f} は考えなかった。

実験では、毎回ランダムに50事例を選択し、その識別概念を学習することにより、確率 p_i, p_{ij} を推定した。また、この学習を続けて15回繰り返すことにより、各段階において、探索した論理式数、および、訓練事例とは独立な500事例に対する正答率を評価した。図2に、これらの一連の試行を100回繰り返したときの平均値を示す。RF4を用いて学習を繰り返せば、探索効率が改善されるだけでなく、未知の事例に対する正答率も向上する傾向にあることが、KRK問題において確認できた。

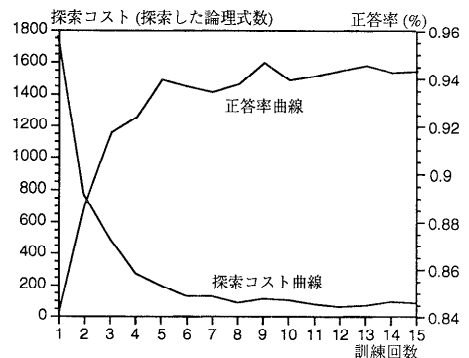


図2 KRK問題での学習曲線
Fig.2 Learning curves for KRK problem.

5. ボンガルドへの適用

5.1 ボンガルド問題

ボンガルド問題^{1),3)}とは、M. Bongard が考案した図形に関する多彩な識別概念を如何に発見するかという問題である。そこには、さまざまな識別問題の型が存在し、識別の困難さも多様なレベルで現われる⁶⁾。従って、機械学習の研究において、興味深く有用なテストケースであると考えられる。しかし、その解決に向けた試みはほとんどなされていない。ボンガルド問題の例を図3に示す。各問題は、図形オブジェクトが描かれた12個のボックス(事例)からなり、左側の6個のボックスはクラス1に属し、右側の6個はクラス2に属すとする。問題はクラス1とクラス2の識別概念を見つけることである。なお、ボンガルド問題は、例え

ば、概念学習アルゴリズムの評価に広く利用されている“東行き/西行き列車の識別問題”⁷⁾と基本構造が同じであり、同様な現実問題は少なくないと考える。

5.1.1 問題表現

各問題は、図形オブジェクトの作図過程から得られる情報を利用して表現する。例として、黒い楕円を描く場合を考える。まず、アイテム“oval”を図形オブジェクトの“shape”メニューから選択し、次に、楕円の左上と右下の座標をマウスで指定し、そして、値“black”を“texture”メニューから選択する。このように設定された値を用いれば、簡潔な問題記述が得られる。ここでは、オリジナル属性(shape name, texture name, shade direction, rotation angle, line width)の値はユーザが直接設定し、派生属性(size, convexity, relation, number of angles, roundness, aspect ratio, gravity)の値はオリジナル属性値から計算で求める。

5.2 基本機能の評価

5.2.1 RF4の結果

パソコン上にCで実現したRF4のプログラムは、100のポルトガル問題のうち41問に対して、各々数秒で正答できた。正答できた問題のタイプは、概念の生成に必要な操作で分類できる。すなわち、単一の図形オブジェクトの条件に関する問題(Single)、ある条件を満たす図形オブジェクトの存在に関する問題(Existence)、図形オブジェクト間の属性比較をする問題(Comparison)、図形オブジェクトを数える問題(Counting)である。図3に、各問題タイプの典型例とRF4の出力結果を示す。

5.2.2 既存法との比較

RF4の能力を比較評価するため、代表的な概念学習アルゴリズムであるGOLEM¹²⁾、INDUCE⁹⁾、FOIL¹³⁾を用いて、ボンガルド問題解決を試みた。ここで、実験に用いたプログラムのバージョンは、“Golem alpha version”, “induce 3-version as of feb 10 1984”, “FOIL.2”である。なお、今回の実験では、各プログラムにはその初期設定のパラメータを用いた。表2に、各アルゴリズムの主な特徴を示す。

各アルゴリズムを同一条件で比較するため、既存法には、RF4が生成する基本識別式を問題表現として与

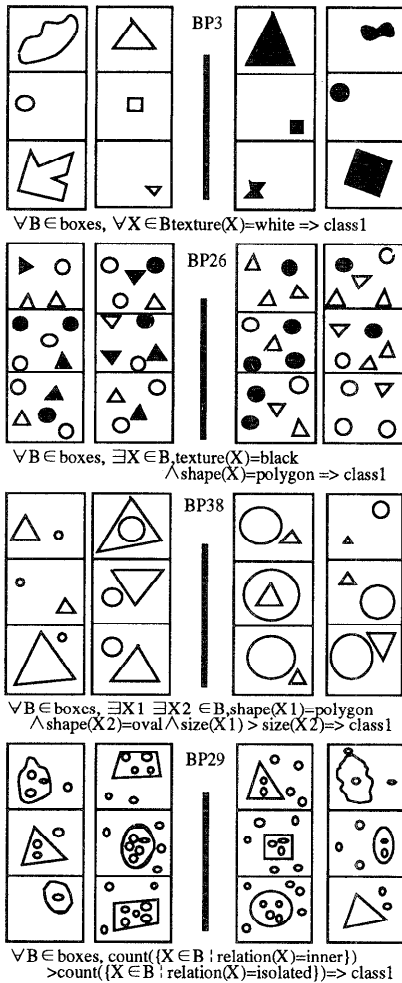


図3 ボンガルド問題の例
Fig. 3 Examples of Bongard problems.

表2 各アルゴリズムの主な特徴

Table 2 Major characteristics of four algorithms.

手法	探索法	枝刈り尺度	生成概念の特徴
INDUCE	ビーム	正答率&簡潔性	count (limited)
GOLEM	ランダム	determinate literals	再帰
FOIL	欲張り	情報量の期待値	再帰
RF 4	深さ優先	5種の学習バイアス	比較&count

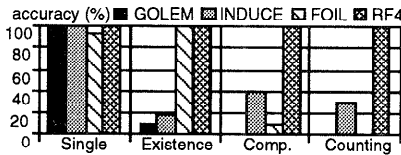


図4 問題タイプ別の正答率

Fig. 4 Performance in four problem categories.

えた。例えば、RF4と同じクラスタリング手法を用いて、2つの図形オブジェクトのサイズがほとんど等しいときには、*eqsize* ($X1, X2$) のような述語を与えた。

実験では、既存法が正答できた問題は、いずれもRF4で正答していた。図4に、問題タイプ別の各アルゴリズムの正答率を示す。以下に、既存法が正答できなかった主な理由を考察する。

- 各ボックスに2つ以上の図形オブジェクトがある場合、GOLEMはどの問題も正答できなかった。なぜなら、GOLEMにおいて本質である“determinate literal”と呼ばれるヒューリスティックス（新リテラルの変数の値は、ホーン節で既に出現した変数の値を与えることにより、一意に決定されねばならない）を採用しているためである。すなわち、ボンガルド問題では、任意の解の形式は、 $class\ 1(B) : \text{-contain}(B, X), \dots$ と表現される。しかし、2つ以上の図形オブジェクトがある場合、変数 B の値を与えても、変数 X の値は一意には決定されない。
- FOILは比較型をほとんど正答できず、INDUCEはこの型の半分以上の問題を正答できなかった。なぜなら、両者はヒューリスティック探索（前者は欲張り探索、後者はビーム探索）を行うためである。例として、FOILによる、図3に示したBP38の問題解決について考える。正答の概念は、 $class\ 1(B) : \text{-contain}(B, X1), \text{contain}(B, X2), \text{polygon}(X1), \text{oval}(X2), \text{gtsize}(X1, X2)$ と表現できる。FOILでは、事例のクラス分布に関する情報量を評価尺度とするので、 $\text{polygon}(X1)$ はすべての事例に現れるため、情報量は0（分類に関する情報量は増えない）となり、それを探索に用いることができない。なお、探索法を変えても、FOILの評価尺度に基づけば、情報量は0なので、識別概念を求めるには、ほとんど全探索をすることになり、現実的ではない。
- count型では、GOLEMとFOILはどの問題も正答できず、INDUCEは一部の問題しか正答できなかった。なぜなら、FOILとGOLEMにはオブジ

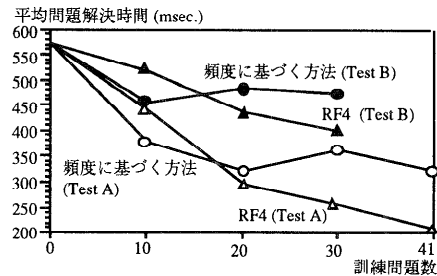


図5 ボンガルド問題での学習曲線

Fig. 5 Learning curves for Bongard problems.

ェクトのcount機能がなく、一方、INDUCEはcountできるが、countした値を比較する機能がないためである。ここで、他のアルゴリズムに対して、単純にcount機能を追加しても、組合せの数が極端に増大し、妥当な時間で問題を解くことができなくなると考えられる。

5.3 適応機能の評価

5.3.1 RF4の結果

RF4の適応機能がどれだけ概念学習効率を改善したかを評価するため、原子式の利用頻度だけに基づいて探索順序を決定する方法との比較を行った。ここで、問題特徴 f は、名義属性からなる原子式に存在限量子を施した論理式の集合とした。実験では、訓練問題は、RF4が解決した41のボンガルド問題からランダムに選択し、テスト問題には、41問すべてをテスト問題とするタイプ (Test A)、および、訓練に利用しなかった残りの問題をテスト問題とするタイプ (Test B) を用いた。また、訓練問題数は10単位で増やし、これらの試行をそれぞれ40回繰り返して評価した。

図5に、テスト問題を解くのに要した平均CPU時間を示す。ただし、2つの方法には、同じ順番で訓練問題を与えた。以下に、実験結果を考察する。

- RF4では、41の訓練問題を解いた後、同じ訓練問題を解くのに要した平均時間は1/3に減少した。また、30の訓練問題を解いた後、残りのテスト問題を解くのに要した平均時間は2/3減少した。ゆえに、RF4の適応機能はうまく働いたと考える。
- 訓練問題数が少ない段階では、頻度に基づく方法の問題解決効率はRF4の効率よりも優れていた。その理由は、RF4には、確率計算と原子式選択の2つの付加タスクが必要であり、もし、推定確率の信頼性があまり高くなければ、これらのタスクがRF4への単なるオーバーヘッドになるためと考える。
- 訓練問題数が増えても、頻度に基づく方法の問題

解決効率は、早い段階で改善されなくなった。その理由は、ボンガルド問題では、頻度に基づく方法は、RF 4 と比較して、かなり早い段階でその限界に達したためと考える。

5.3.2 知識の検証

RF 4 の適応機能が探索効率を改善したのは、推定確率を利用したからであるが、確率は多数の数値集合により表現されるので、実際に獲得した知識が何を意味するかを明確に知ることは困難である。ここでは、 χ^2 -検定を用いて、一部の顕著な知識（予め定義した状況ベクトルと識別概念の因果関係）の描出を試みる。

状況ベクトルのある要素が真となる事象を e_1 、ある原子式が識別概念の構成要素となる事象を e_2 、2つの事象 e_1, e_2 が同時に起こる事象を e_{12} とする。また、推定に利用した事例数を m 、3つの事象 e_1, e_2, e_{12} が起こる確率の推定値をそれぞれ $\bar{p}_1, \bar{p}_2, \bar{p}_{12}$ とする。もし、2つの事象 e_1, e_2 が独立ならば、

$$\chi^2 = m \frac{(\bar{p}_{12} - \bar{p}_1 \bar{p}_2)^2}{\bar{p}_1(1 - \bar{p}_1) \bar{p}_2(1 - \bar{p}_2)} \quad (8)$$

は自由度 1 の χ^2 -分布に従う。もし、 χ^2 の値がある値より大きければ、指定した信頼率で、2つの事象 e_1, e_2 が独立であるという仮説を棄却できる。実験では、信頼率 95% ($\chi^2 \leq 3.84$) で棄却できる仮説を求めた。

得られた結果をルール形式で表せば、「oval が存在するならば、他の図形オブジェクトに対する内部/外部の関係を調べろ ($\chi^2=5.45$)」, 「polygon が存在するならば、図形オブジェクトの角数を調べろ ($\chi^2=4.97$)」, および、「rectangle が存在するならば、他の図形オブジェクトに対する内部/外部の関係を調べろ ($\chi^2=4.14$)」である。第 2 のルールについては、図形オブジェクトが polygon であれば、その角数を比較するという、人間の直観的な知識と符号する。一方、第 1 と第 3 のルールについては、oval や rectangle は、他の図形オブジェクトの内部に描かれる小さな図形オブジェクトとして、しばしば利用されることに符合する。よって、ボンガルド問題において、問題解決の高速化に寄与した知識を検出できたと考える。なお、著者らは、第 1 と第 3 のルールについては、全く気付いていなかった。

6. おわりに

過去に解いた問題を用いて、自ら適応した探索効率を改善する概念学習法 RF 4 を開発した。すなわち、複数の枝刈り尺度と概念の複合化ルールを利用した深さ優先探索において、状況ベクトルから、原子式が識別概念の構成要素となる確率を計算することにより、適応して探索を高速化する。KRK 問題では、RF 4 の探

索効率が改善されるだけでなく、未知の事例に対する正答率も向上することを確認した。ボンガルド問題では、100 のボンガルド問題に対して、既存の代表的な概念学習アルゴリズムでは、20 数問程度しか正答できなかったが、RF 4 は 41 問を正答できた。また、RF 4 が 41 問を解答した後では、その問題解決時間は平均して 1/3 に短縮され、問題解決を高速化する知識の一部も抽出できた。RF 4 の限界と課題については、多数の原子式から構成される識別概念の生成とノイズを含む問題への適用性が挙げられる。今後は、アルゴリズムの改良を進めるとともに、幅広い問題に RF 4 を適用して評価する予定である。

参 考 文 献

- 1) Bongard, N.: *Pattern Recognition*, Spartan Books (1970).
- 2) Duda, R. and Hart, P.: *Pattern Classification and Scene Analysis*, John Wiley & Sons (1973).
- 3) Hofstadter, D.: *Gödel, Escher, Bach: An Eternal Golden Braid*, Basic Books Inc. (1979).
- 4) Laird, J., Rosenbloom, P. and Newell, A.: *Chunking in SOAR: The Anatomy of a General Learning Mechanism*, *Machine Learning*, Vol. 1, No. 1, pp. 11-46 (1986).
- 5) Michalski, R.: A Theory and Methodology of Inductive Learning, Michalski, R., Carbonell, J. and Mitchell, T. eds., *Machine Learning: an Artificial Intelligence Approach*, pp. 331-364, Morgan Kaufmann, San Mateo, CA (1983).
- 6) Michalski, R. and Kodratoff, Y.: Research in Machine Learning: Recent Progress, Classification of Methods, and Future Directions, Kodratoff, Y. and Michalski, R. eds., *Machine Learning: an Artificial Intelligence Approach*, Vol. III, pp. 3-30, Morgan Kaufmann, San Mateo, CA (1990).
- 7) Michalski, R. and Larson, J.: Inductive Inference of Inductive Learning, *SIGART Newsletter*, ACM, No. 63, pp. 38-44 (1977).
- 8) Minton, S.: Quantitative Results Concerning the Utility of Explanation-Based Learning, *Artif. Intell.*, Vol. 42, pp. 363-392 (1990).
- 9) Mitchell, T.: Generalization as Search, *Artif. Intell.*, Vol. 18, pp. 203-226 (1982).
- 10) Mitchell, T., Keller, R. and Kedar-Cabelli, S.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, Vol. 1, No. 1, pp. 47-80 (1986).
- 11) Muggleton, S., Bain, M., Hayes-Michie, J. and Michie, D.: An Experimental Comparison of Human and Machine Learning Formalisms,

Proc. Sixth International Machine Learning Workshop, pp. 113-118 (1989).

- 12) Muggleton, S. and Feng, C.: Efficient Induction of Logic Programs, *Proc. First Conf. Algorithmic Learning Theory*, pp. 368-381 (1990).
- 13) Quinlan, J.: Learning Logical Definitions from Relations, *Machine Learning*, Vol. 5, No. 31, pp. 239-266 (1990).
- 14) Samuel, A.: Some Studies in Machine Learning Using the Game of Checkers, *IBM J. Res. Dev.*, Vol. 3, pp. 211-229 (1959).

付 録

(3)式の導出について述べる。まず、近似確率の定義(1)式と(2)式、および、 $\beta^{(h)}$ の定義(5)式を考慮すれば、

$$P_2(s_1, \dots, s_h) = P_1(s_1, \dots, s_h) \beta^{(h)}$$

であることを確認する。ここで、

$$P_2(s_k, s_1, \dots, s_h) = p_k^{s_k} (1-p_k)^{1-s_k} P_1(s_1, \dots, s_h) \times \left(\beta^{(h)} + \sum_{j=1}^h (p_{kj} - p_k p_j) y_j \right)$$

であり、 $s_k=1$ のとき、 $y_k=1/p_k$ となるので、

$$P_2(s_k=1, s_1, \dots, s_h) = p_k P_1(s_1, \dots, s_h) \beta^{(h)} + P_1(s_1, \dots, s_h) \sum_{j=1}^h (p_{kj} - p_k p_j) y_j$$

となる。従って、 $\alpha^{(h)}$ の定義(4)式を考慮すれば、

$$P_2(s_k=1 | s_1, \dots, s_h)$$

$$= \frac{P_2(s_k=1, s_1, \dots, s_h)}{P_2(s_1, \dots, s_h)} = p_k + \frac{\sum_{j=1}^h (p_{kj} - p_k p_j) y_j}{\beta^{(h)}} = p_k + \frac{\alpha^{(h)}}{\beta^{(h)}}$$

となり、(3)式が導出できる。

(平成6年7月25日受付)

(平成7年1月12日採録)



齊藤 和巳 (正会員)

1963年生。1985年慶應義塾大学理工学部数理学科卒業。同年、NTT入社。1991年より1年間オタワ大学客員研究員。神経回路網、機械学習の研究に従事。現在、NTTコミュニケーション科学研究所主任研究員。人工知能学会、神経回路学会、日本認知科学会各会員。



中野 良平 (正会員)

1947年生。1971年東京大学工学部計数工学科卒業。工学博士。同年、日本電信電話公社(現NTT)入社。以来、統計解析、分散処理、データベース、人工知能の研究に従事。現在、NTTコミュニケーション科学研究所主幹研究員。人工知能学会、神経回路学会、日本応用数学会各会員。