論　文

Technical Papers

類型パターンの抽出に基づく帰納的学習と演繹的学習の統合

Towards a Common Framework for Inductive and Deductive Learning Using Colored Digraphs

吉田　健一*　元田　浩*　Nitin Indurkhya*
Ken'ichi Yoshida　Hiroshi Motoda

*　(株)日立製作所 基礎研究所
Advanced Research Laboratory, Hitachi, Ltd., Hatoyama, Saitama 350-03, Japan.

1993年2月22日　受理

Keywords: inductive learning, deductive learning, colored digraph.

Summary

We describe a new learning method, CLiP (Concept Learning from Typical Patterns), that performs induction over colored directed graphs. CLiP is capable of performing both inductive and deductive learning by mapping the problems into a colored digraph representation. In contrast to earlier approaches, CLiP uses a single learning algorithm to solve both kinds of problems. The learning procedure can be characterized as a variation of beam search guided by a simple, but effective, heuristic: *typical pattern heuristic*.

We demonstrate the applicability of CLiP to the tasks of (1) inductive learning for classification and (2) deductive learning for efficient problem-solving. We show that the performance of CLiP on these tasks is comparable to that of standard approaches. Our preliminary results suggest that the generality of CLiP can be attributed to the expressiveness of the colored digraph representation which allows a number of seemingly different learning problems to be solved by a single algorithm. The other functions of CLiP and the limitations are also discussed together with the related work.

1. はじめに

近年，計算機パワーの増大に伴い，大量データから分類ルールを帰納的に学習[Mitchell 84, Quinlan 86]する研究が実用の域に入りつつあるが，帰納的学習に関する既存の研究では領域の専門家知識を併用してデータを整理することは難しい．また，これとは異なる観点から，既存の領域知識を利用する学習手法としてEBLなど演繹的なマクロルール学習[DeJong 86, Mitchell 86]の研究も進められている．しかし，これまで両者の研究は別々に進められており，既存の領域知識を併用しながら大量データを解析する技術の研究は十分進められていない．

本研究では，既存の領域知識を併用しながら大量データを解析する学習手法の研究の一環として，現在まで行われている帰納的学習と演繹的学習の統合を試みる．最終的なねらいは「既存の領域知識を併用しながら大量データを解析する」技術にあるが，本研究では，「関連する研究の統合的枠組み」を提案し，そのような技術への足掛りとする．

以下では**2章**で，帰納的な分類ルールの学習と演繹的なマクロルールの学習の二つの学習機能を併せ持つアルゴリズムCLiP(Concept Learning from Typical Patterns)を提案する*1．CLiPは「よく現れるパターンは何らかの意味を持つ」というヒューリスティック(typical pattern heuristic)に基づいた一種のビーム

*1　実際にはCLiPは，推論過程からの概念学習手法CLIP(Concept Learning from Inference Patterns)[吉田 92a, 吉田 92b]として提案済みのアルゴリズムの使い方を一般化(**Inference** Pattern から**Typical** Pattern に一般化)しただけであり，正確には再提案である．**5・3節**において，上記研究と本研究の関係を詳細に検討する．

サーチアルゴリズムで，色つき有向グラフを用いて表現された例題によく現れる類型的パターンを抽出する．3章では抽出された類型パターンを①帰納的学習におけるデータの分類ルールや②演繹的学習におけるマクロルールとして解釈する方法を示し，4章で①特定のDNA(Promoter)シーケンスの分類ルールの学習や，②1次方程式解法用マクロルールの学習に応用した実験結果を示す．5章ではCLiPのその他の機能と限界を，関連研究との比較を交えて考察する．

2. CLiP : Concept Learning from Typical Patterns

CLiPは「よく現れるパターンは何らかの意味を持つ」というヒューリスティックに基づいた一種のビームサーチアルゴリズムを採用している．このアルゴリズムは色つき有向グラフによく現れる類型的パターンを抽出し，抽出したパターンを用いて入力したグラフを小さなグラフに変換(縮約)するものである(図1)．

ここで，変換後のグラフでは，変換前のパターンを，パターンに対応して新しく作られた色(グラフ理論でラベルの意)を持つ一つのノードに書き換える．例えば図1でノード⑧はパターン⑥→⑤…⇒①に対応している．また，処理の対象となるグラフは一つにつながった大きなグラフでもよいし，小さなグラフの集まったものでもよい．

さらに，縮約操作の過程で利用するグラフマッチング処理に制約を課し，計算機処理に要する時間が指数オーダとなることを避けている．一般的にグラフのマッチング処理というと[足立 88]に述べられているようなグラフ同型問題を考えることが多いが，グラフ同型問題ではマッチングを調べるときに，対応関係を調べるノードに関して何番目のエッジであるかは考慮せずに，考えられるすべてのエッジの組合せに対して対応関係を調べている．この問題を多項式オーダで解くアルゴリズムは知られておらず，処理負荷の大きな問題である．一方CLiPでは，事前にエッジに番号づけを行い，グラフ間(CLiPの場合は一方が抽出処理中のパターン)のマッチングを調べるときに，同じ番号のエッジ間でのみ対応するか否か調べている．この場合，「同じ番号のエッジ間でのみ」という制約を用いて効率良いアルゴリズムを作成することは簡単である．

例えば図1では左下の三つのパターンのうち二つのみが上に示したパターンと同じとして右下で⑧に書き換わっているが，残り一つはそのまま違うパターンとして書き換わらずに残っている．CLiPではこのような処理を行うが，グラフ同型問題では右下に残ったパターンも同じとして処理される．

このグラフ理論の立場から見た場合のマッチング処理に関する制約は，次章で述べるように，抽出したパターンを利用するうえでは何ら制約となっていないことに注意されたい．すなわちCLiPではエッジ番号にもデータ処理上での意味があり，「同じ番号のエッジ間でのみの比較」は「意味の異なるデータは比較しない」ことになり，実質上の制約になっていない．

図2にCLiPで用いている類型パターン抽出アルゴリズムの概略を示す．詳細は[吉田 92a]に報告済みで

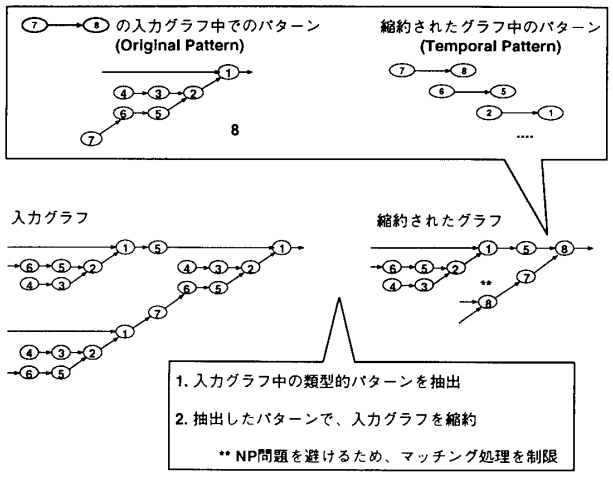

図1 CLiP法の基本的なアイディア

```
Algorithm CLiP(G_in, C, L, W)
  Input    G_in   : Colored Directed Graph
           C      : Selection Criterion
           L, W   : integer
  Output   Sequence of V_i where each V_i is
                  a Set of Typical Patterns in G_in
  Variable B, B_next : Set of Views
  begin
    V_0 ← ∅ ;   B ← {V_0} ;   i ← 1
    repeat L do
      B_next ← ∅
      for each V_tmp ∈ B do
        Call Pattern Modification
      Call Pattern Combination
      B ← Top W views in B_next according to C
      V_i ← Best view in B_next according to C
      i ← i + 1
    return Sequence of V_i
  end

Procedure Pattern Modification
  begin
    G_tmp ← Graph that is contracted from G_in
                  according to the patterns in V_tmp
    for each Temporal Pattern P in G_tmp do
      V_new ← V_tmp ∪ {Original Pattern of P}
      Append V_new to B_next
  end

Procedure Pattern Combination
  begin
    for each V_tmp1 ∈ B do
      for each V_tmp2 ∈ B do
        if V_tmp1 ≠ V_tmp2 then
          V_new ← V_tmp1 ∪ V_tmp2
          Append V_new to B_next
  end
```

図2 CLiP法の概略アルゴリズム

あるので，ここでは以下の説明に重要な入出力のみ説明する．

入力としては例題を表現する色つき有向グラフ G_{in} とグラフサイズの評価基準 C の二つが重要である．CLiP はこれに対して幅 W のビームサーチを L 回繰り返し，G_{in} を縮約した結果が評価基準 C に照らしたときに最も小さくなるパターンの組合せ(View)を出力する*2．

また，このアルゴリズムのなかで類型パターンの組合せ(View)の生成を行っているのは Pattern Modification 処理と，Pattern Combination 処理であるが，類型パターンの抽出には Pattern Modification 処理が効果的であることがわかっており[吉田 92b]，**4・1 節**，**4・2 節**で述べる実験では高速化のため Pattern Modification 処理のみ利用した．

3. 色つき有向グラフによる学習問題の表現

本章では，帰納的学習と演繹的学習の両分野において，学習すべき例題を色つき有向グラフ G_{in} として表現する方法と，グラフサイズの評価基準 C の与え方，および抽出したパターンの解釈方法を述べる．

3・1 データからの分類ルールの帰納的学習

大量のデータからその分類ルールを帰納的に学習する研究では，生物の遺伝情報を司る DNA の塩基配列や，生命活動のなかで重要な役割を持つタンパク質アミノ酸配列分析などが，例題として盛んに用いられている．

このような帰納的学習ステップを細かく分析すると，

1. データから抽出する情報(属性)を決定する．
2. 属性のリストとして表現されたデータから，分類ルールの鋳型を抽出する．
3. 分類ルールの鋳型を一般化し，分類ルールを作成する．

の3ステップより構成されている．通常第1ステップは人間が行い，第3ステップは「属性値の組合せが同じデータは同じクラスに属する」という暗黙のルールによる一般化を行うので，第2ステップの処理に重点が置かれ研究が進められている．

本節では DNA 塩基配列のうち promoter シーケンス*3[日経 91]と呼ばれる配列の特徴(分類ルール)をデータから学習するという例題を使って，上記の第2ステップを CLiP で扱う方法を示す．ここで，第3ステップは「属性値の組合せが同じデータは同じクラスに属する」という暗黙のルールを用いる．また，第1ステップについては**5・4 節**で考察する．

図3に DNA の塩基配列をグラフ表現した例(以下直接表現形式：direct representation と呼ぶ)を示す．この場合，CLiP の入力 G_{in} はおのおのが一つの塩基配列(例題)に対応した小さなグラフの集まりとなる．個々のグラフは根ノードが例題が promopter であるか否かのクラス情報(正または負)を色として持ち，根ノードの i 番目のエッジにつながった葉ノードが i 番目の塩基の種類に関する情報(A/T/C/G のいずれか)を持っている．

CLiP により抽出されたパターンは，パターンの葉ノードの色(塩基配列のなかで特定位置の塩基の種類を指定)を条件部に，根ノードの色を結論部(データが promoter であるか否かを指定)にすることで，DNA の塩基配列が Prompoter であるか否かを分類するためのルールとして利用できる．

ここで，グラフやパターンの根ノードでのエッジの位置情報がデータの属性に対応して利用されている(i 番目のエッジにつながった葉ノードが i 番目の塩基の種類，すなわち属性に関する情報を持つ)ことに注意されたい．帰納的学習での応用を考えた場合，**2 章**で述べたグラフマッチング処理に関する制約は，処理速度の向上に効果があるだけで，実質的な制約にはなっていない．

グラフサイズの評価基準 C は，なるべく多くの例題を，なるべく少ない分類ルールで分類するように，

$$\left[\begin{array}{l}(\text{縮約されたグラフのノード数}) \\ \quad +(\text{縮約されたグラフの持つ色の種類})^2\end{array}\right]$$

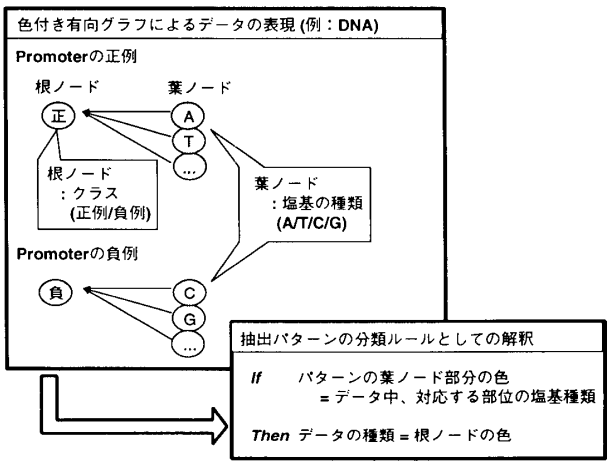

図3 帰納的学習のためのデータ表現

*2 正確には探索途中の各ステップで，そのときに見つかった最良の View を出力するので，View の列を出力している．

*3 DNA を鋳型に mRNA 合成を開始する DNA 上の特定塩基配列．遺伝情報に基づいたタンパク質合成をコントロールする役目を持っている．

を最小化するように指定する(以下これを評価基準①と呼ぶ).第1項は多くの例題によく現れる大きなパターンを見つけることを意図している.promoterの学習では,promoterに特有の塩基配列が類型パターンとして抽出できれば,抽出したパターンでグラフを縮約することで,第1項が小さくなり,良い評価結果を得ることができる.すなわち,この項は,いわゆるカバーリング[Quinlan 90]と呼ばれる動作をCLiPにさせないことを意図している.第2項は,なるべく少ないルールで分類することを意図している.すなわち,多種類のルール(パターン)はそれぞれ独自の色を持つので,第2項はルールが増えすぎたときのペナルティとなる.

一般的な帰納的学習の問題に類似の表現形式を利用することは可能と思われるが,直接表現形式には以下の二つの問題がある.

- CLiPと直接表現形式の組合せでは,"*If the i-th nucleotide is NOT A*"のような,否定的条件を扱えない.
- 正例・負例ともに共通の特徴を多く持っていた場合(DNAの例では,同じ位置に共通の塩基を常に持っていたなど),CLiPでは条件部が同じで,結論部のみ違うルールを別々に作成してしまう可能性がある.このようなルールは分類ルールとしては意味をなさない.

初めの問題には例題をグラフ表現する方法を変えることで対処できる.具体的には根ノードの色情報は直接表現形式と同じで,$(4\times i+j)$番目の葉ノードの色としてi番目の塩基が$A(j=0)/T(j=1)/C(j=2)/G(j=3)$のいずれであるかの情報を持たせた2値表現形式(binary representation)を利用すれば否定的条件を扱える.すなわち1番目の塩基がAであれば,1番目の葉ノードの色はtrue,2・3・4番目の葉ノードの色をfalseとし,抽出したパターンが葉ノードの色としてfalseを持っていれば,否定的条件を学習したことになる.

2番目の問題に対応するために,下記の値を最小化するという評価基準Cを考案した(以下これを評価基準②と呼ぶ).

$$\left[\begin{array}{l}(\text{縮約されたグラフのノード数})\\ \quad+(\sum f(\text{縮約されたグラフのなかの Color1}_j))^2\end{array}\right]$$

ここで,

f(Color1)
=(Color1を持つノードが持つ
Color2の種類)^2

第1項は評価基準①同様カバーリングをCLiPにさせることを意図している.また,Color1には塩基の種類またはパターンの種類を示す色を,Color2にはデータが正例であるか負例であるかのクラス情報を用いる.また,Color2はマッチング処理の過程では無視し,評価にのみ用いる.

分類ルールとして意味を持たないパターンがあると,そのパターンに対応するクラス情報(Color2)は正・負の2種類,f(Color1)は$2^2=4$となり,第2項がペナルティとなる.中間結果として意味のない分類ルールに相当するパターンができたとしても,Pattern Modification処理でそのパターンが大きくなると,分類ルールの条件部分は新しい条件が付け加わったのと同じことになり,結局判定条件が詳細化され,ペナルティの小さなパターンの評価結果が良くなる.結局分類ルールとして意味のあるものを評価結果の良いパターンとして探そうというのが,評価基準②の意図である.

3・2 マクロルールの演繹的学習

領域知識を用いた機械学習の手法として,EBLなどマクロルールの演繹的学習の研究が,データからの分類ルールの帰納的学習とは別に進められている.マクロルールの演繹的学習では,事前にベースとなる領域知識を用いて行った推論過程を分析し,推論の高速化に効果があるマクロルールを学習しようとする.このような演繹的学習の学習ステップを細かく分析すると,

1. 領域知識を用いて推論を行う.
2. 推論過程を分析し,マクロルールの鋳型を抽出する.
3. ルールの鋳型を一般化し,マクロルールを作成する.

の3ステップより構成されている.第3ステップが,本質的には証明木のMGUを抽出する操作であることは,Mooneyらにより解明されている[Mooney 86].また,実行された推論すべてをマクロ化すると,適切なマクロを選択する処理に必要な時間が増加し,かえって効率が落ちるという問題点が指摘され,鋳型として何を選ぶかという第2ステップに関する研究が盛んに行われている[Minton 90].

CLiPは,このタイプの学習(上記の第2ステップ)を行うことも可能である.図4に,この場合の入力となる推論過程の色つき有向グラフによる表現形式を示す.

この場合,グラフは一種の証明木となっている.各ノードは証明過程に現れたデータに対応しており,色

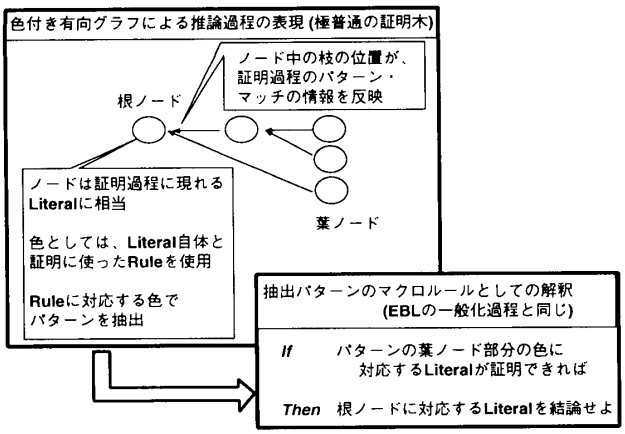

図4 演繹的学習のためのデータ表現

としてノードのデータを証明するのに用いたルール(Color1, prologの例ではclause)と，データそのもの(Color2, prologの例ではliteral)を持っている．ここでも，マッチング処理にはColor1のみ用いる．また，各ノードにおいて何番目のエッジであるかは，パターンマッチングの情報(prologの例ではbody部の何番目に現れるclauseの証明過程に対応しているか)を表している．

CLiPにより抽出されるパターンは証明過程でよく出てくるルールの適用方法に対応しており，パターンのマクロルールとしての解釈操作はEBLの一般化操作と同じ操作である．具体的には，パターンの末端の葉ノードに記憶されているColor2(データ)がEBLでいうところのoperational criteriaを満足し，根ノードに記憶されているColor2がgoal conceptであると考え，EBLと同じ仕組みで変数の一般化操作を行う．

ここでも，グラフやパターンのエッジの位置情報が，推論過程でのパターンマッチングの情報を表すのに利用されていることに注意されたい．すなわち，演繹的学習での応用を考えた場合にも，**2章**で述べたグラフマッチング処理に関する制約は，実質的な制約にはなっていない．

評価基準Cとしては，

$$\left[\begin{array}{l}(\text{縮約されたグラフのノード数}) \\ \quad +(\text{縮約されたグラフの持つ色の種類})^2\end{array}\right]$$

を最小化するように指定する．

第1項は推論過程に現れるデータの数を表している．通常のマクロルール学習と同じく，CLiPでも推論過程に現れるデータ数を削減する(推論ステップを減らす)ことで，推論の高速化を試みており，第1項は，ステップ数削減の効果を計算するための項である．

また，第2項は学習前から持っていた推論用のルールのうち学習後も使うものと，学習したマクロルールの総和を減らすためのペナルティ項である．機械的にすべての例題をマクロ化すると，むだなマクロのマッチング処理負荷が増え，かえって推論効率が悪くなる．第2項はそのようなむだなマクロによる効率の悪化を避けるために導入したペナルティ項である．

4. 実験結果

4・1 DNAの塩基配列の分類ルール学習

[Towell 90]に記載されているDNAの塩基配列データを用いて，CLiPの帰納的学習機能の実験を行った．データはpromoterと呼ばれるDNAの塩基配列53例(正例)と，それ以外のDNAの塩基配列53例(負例)から構成されている．個々の例は長さ57の文字列で，先頭からi番目の文字が塩基配列中i番目の塩基の種類(A/T/C/Gのどれか)を表している．

leave-one-out手法[Efron 82]により，学習結果のルールと「類型的パターンを含まなければpromoterではない」というルールを与えたときの，未知問題に対する誤答率が最も低くなるV_iを選択した．実験における誤答率の変化を図5に示す．この実験では，promoterは非常に特徴のある塩基配列を示し，類型パターンの探索が進む(図5において，探索繰返し回数が大きくなる)に従い，promoter配列に特有の塩基配列が類型的パターンとして抽出され，正例をカバーする分類ルールが学習され，図5において探索繰返し回数4のときに，未知問題に対する誤答率が最も低くなるV_iを学習した．それ以後は，正例に対する過剰学習が起こり，誤答率が上がった．

この結果を他の手法による結果と比較したものを表

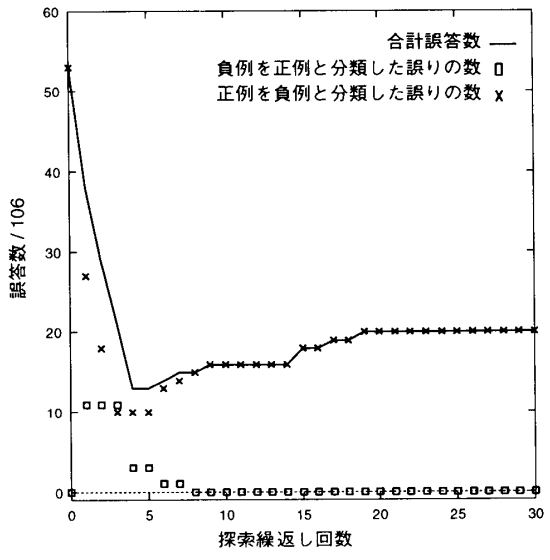

図5 学習曲線

表1 帰納的学習の結果

学習手法	既存の帰納的学習手法			CLiP		
	ID3	SWAP1	BP	Direct	Binary	Direct'
誤答数/106	19	14	8	14	29	13

$$塩基_{06} = A \ \wedge \ 塩基_{15} = T \ \wedge \ 塩基_{16} = T \quad \rightarrow \quad 正例$$

$$塩基_{16} = T \ \wedge \ 塩基_{17} = G \quad \rightarrow \quad 正例$$

$$塩基_{15} = T \ \wedge \ 塩基_{16} = T \quad \rightarrow \quad 正例$$

$$True \quad \rightarrow \quad 負例$$

図6 直接表現法により作成された分類ルールの例

$$塩基_{15} \neq A \ \wedge \ 塩基_{15} \neq C \ \wedge \ 塩基_{16} \neq A \ \wedge \ 塩基_{17} \neq A \ \wedge \ 塩基_{18} \neq G \ \wedge \ 塩基_{30} \neq C \ \wedge \ 塩基_{38} \neq C \ \rightarrow \ 正例$$

$$塩基_{17} \neq A \ \wedge \ 塩基_{26} \neq C \ \wedge \ 塩基_{38} \neq G \ \wedge \ 塩基_{39} \neq G \ \wedge \ 塩基_{40} \neq C \ \wedge \ 塩基_{54} \neq G \ \rightarrow \ 正例$$

$$塩基_{15} \neq A \ \wedge \ 塩基_{15} \neq C \ \wedge \ 塩基_{17} \neq A \ \wedge \ 塩基_{18} \neq G \ \wedge \ 塩基_{30} \neq C \ \wedge \ 塩基_{38} \neq C \ \rightarrow \ 正例$$

$$True \quad \rightarrow \quad 負例$$

図7 2値表現法により作成された分類ルールの例

1に示す．表1においてID3[Quinlan 86]とSWAP1[Weiss 91]は知識表現の形式としてルールを採用している．また，BPは一層のhidden layerを持つneural networkを標準的なback propagation法により学習させた結果である．

Directは直接表現形式と評価基準①によるCLiPの誤答率，Binaryは同じ評価基準と2値表現形式を用いた誤答率，Direct'は直接表現形式と評価基準②による誤答率である．図6に直接表現形式を用いた実験の，図7に2値表現形式を用いた実験の結果，作成された分類ルールを例示する．

直接表現形式を用いた場合，CLiPの誤答率は標準的な帰納的学習システムであるID3より低くなっていることに注意されたい．この例におけるCLiPはID3のような帰納的学習システムとして働いている．

4・2 First Order Equation Solving

[山田 89]で例題として扱っている85問の1次方程式の解決過程を用いて，CLiPの演繹的学習機能の実験を行った．

入力に用いた1次方程式の解決過程は，1次方程式変形の公理を表現したclause 46個からなるprolog programで作成した．抽出したパターンからは，clauseの適用方法を抽出し，複合clause(別の言い方をすればマクロルール)として合成し，もとのprolog programの先頭に追加したprogramにより処理速度の向上効果を計測した．

図8，図9に，実際に実験に利用したprolog program

```
%    両辺を同じ数Aで割っても(A≠1の時に式を変形，プログラム
%         上では"A<1"で表現)等値関係は維持される
solve(left([],[A],[]),right([],[],[B])) :-
        number(A),number(B),
        A<1,
        solve(left([],[[A,/,A]],[]),
                    right([],[],[[B,/,A]])).

%    b/b = 1
solve(left(A,[[B,/,B]],C),right(D,E,F)) :-
        solve(left(A,[1],C),right(D,E,F)).

%    式が割算 x/y を含めば，それを計算せよ
solve(left([],[1],[]),right([],[],[[X,/,Y]])) :-
        number(X),number(Y),
        XX is X/Y,
        solve(left([],[1],[]),right([],[],[XX])).

%    x = 数字 の形になれば終了
solve(left([],[1],[]),right([],[],[B])) :-
        number(B).
```

上記のprolog述語solveにおいて，第1引数は方程式の左辺，第2引数は方程式の右辺に対応している。また，構造体left/rightの第1項は"("と")"で囲まれた式，第2項は変数Xの係数，第3項は定数を表している。例えば方程式の左辺が

$$(5X + 4) + (6X + 5) + (3 + 4)X + 1 + 2$$

であれば、

```
left([[5,4],[6,5]],[3,4],[1,2])
```

となる。

図8 初めに与えたprolog clauseの例

```
% aX = b で、a,bが数であればX = b/a :
%    図8の四つのclauseを順に適用した過程をマクロ化
%    number(_5162)は、マクロ作成に使われた図8最後の
%        clauseのliteralが、一般化され残ったもの。
%    意味を考えると冗長だが、この冗長なliteralの除去には、今回
%        プログラム化しなかったメタ・レベルの知識が必要である。
%    一方、number(_1216)とnumber(_3353)の二つは、始めのclauseと
%        3番目のclauseの2回分、同じclauseがマクロ中に現れた。
%    こちらは、構文的に同一なので、一方をマクロから除去した。
solve(left([],[_1216],[]),
                right([],[],[_3353])) :-
        number(_1216)
        , number(_3353)
        , _1216<1
        , _5162 is _3353/_1216
        , number(_5162)
        .

% aX + b = c ならばaX = c - bに変形 :
%    「両辺から同じ数を引いても等値関係は維持される」
%        「(a-a)を式から消去」等の意味を持つclause列をマクロ化
%    上記結果と同様、意味を考えると冗長なliteralも残っている。
solve(left([],[_3704],[_1193]),
                right([],[],[_2965])) :-
        number(_1193)
        , _1193>0
        , _1214 is-1*_1193
        , number(_1214)
        , 0 is _1214+_1193
        , solve(left([],[_3704],[]),
                right([],[],[_1214,_2965]))
        .
```

図9 作成されたprolog clauseの例

gramと，学習結果として合成されたprolog programの一部を例示する．また図10に学習前と学習後のprogramに入力した1次方程式が変形されていく過程の例を示す．CLiPには，学習前のprolog program実行のトレース情報から，図中o1～o7で示したprolog clauseの識別番号や，マッチング情報を取り出し，グラフに変形したものを入力した．

図10は，学習前には公理(図8中の個々のprolog clauseが対応)により逐次小さな変形が繰り返されるが，学習後は複合clause(図9中のprolog clauseが対応)により何ステップかの変形が一期に行われ，ステップ数が削減・処理速度が向上されるようすを示している．

表2に結果を示す．表2で，「学習なし」は学習結果

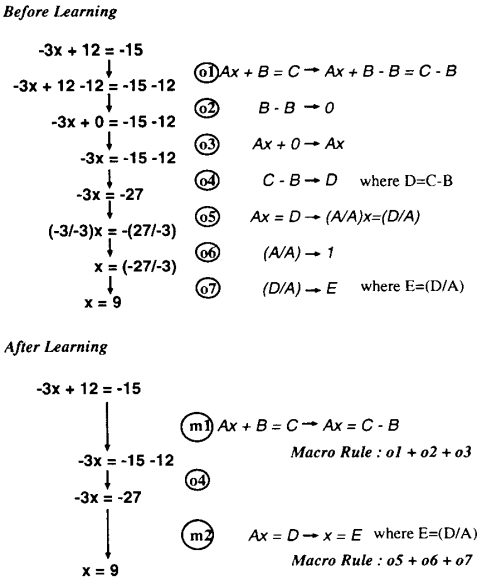

図 10 1次方程式の解法の効率化

表 2 演繹的学習の結果

テスト法	85例で学習 同じ例題で効果を測定			65例で学習 残り20例で効果を測定		
学習法	学習なし	EBL	CLiP	学習なし	EBL	CLiP
処理時間	100.0	279.9	87.5	34.4	46.9	33.3
時間比	1.00	2.80	0.88	1.00	1.37	0.97

を使わなかったときのCPU時間，「EBL」は個々の例題をgoal concept，prologの組込み述語をoperationalとして標準的なEBL手法で学習例題すべてに対してマクロを作成した場合のCPU時間を示している．また，CLiPの結果は学習効果の計測時には，学習例題に対して最も処理効率の改善が見られたViewを用いて作成したprolog programでテスト問題を解き計測した．さらに，学習例題と，テスト例題が同じ場合と異なった場合についても比較を行った．

標準的なEBL手法がむだなマクロの学習により学習なしの場合よりも効率が落ちているのに対して，CLiPでは処理効率が向上している．また，学習例題とテスト例題が異なる場合，学習例に対する過剰学習が起こり，テスト例に対する処理効率改善効果は減っている．

＊4 [山田 89]で用いた推論システムはSTRIPS流の推論システムで，prologを用いた今回の結果とは実験条件が異なり直接の比較はできなかった．しかし，[山田 89]の方法がSTRIPSのoperatorを使い再帰的でないマクロをうまく学習しているのに対して，今回の実験で用いたprolog programではclauseが再帰的になりEBL流の学習効果が得にくい問題設定となっている．具体的には，今回の実験に用いたprolog programでは，学習結果として適切なマクロが得られても，事前に与えたclauseとmacro clauseのheadの形がすべて同じため，得られたマクロに対応するprogram clauseのマッチング処理負荷の増加が著しく，速度向上効果が達成しにくい設定となっている．

この例におけるCLiPはEBLのような演繹的学習システムとして働いている．

5. CLiPと関連研究

CLiPの最大の特徴は，その適用範囲の広さである．個別の問題を考えた場合，CLiPより優れた手法は少なくないと考える．例えば，**4・1節**で扱った実験では，CLiPの結果より標準的なback propagation法により学習したneural networkのほうが良い結果を示している．また，**4・2節**で扱った実験に関しても，例えば，[山田 89]に示された方法の結果が優れていると考える＊4．

表3に著者らがCLiPがカバーしていると考えている機能を整理して示す．既存の研究事例のなかで単一の手法でこれだけの範囲をカバーするものは報告されていない．本章では表に従い関連研究との比較を行い，CLiPにおいて改良すべき点を述べる．

5・1 データからの帰納的学習との比較

CLiPはデータの表現形式としてグラフを用いた帰納的学習システムと考えることができる．評価基準②を用いた場合はID3のような分類ルールの学習システムとみなせる．また，評価基準①を用いた場合にはID3よりはCOBWEB[Fisher 88]のようなクラスタリングシステムに近い．前者は正例・負例のような背反概念を学習するのに対して，後者は似たものを階層的にまとめるという点で否背反概念を教師による概念の指示なしに学習するシステムである．

後者との比較は[吉田 92b]で行った．前者との比較を表4に示す．また，CLiPと後者との類似性を再確認するため，**4・1節**で使用したデータから正・負のクラス情報を削除したデータを用いてCLiP類型パターンの抽出を行った．抽出結果を図11に示す．入力データにはクラス情報が含まれていなかったのにもかかわらず，$Class_1$，$Class_2$は，ほぼpromoterとみなせ，*Other*は，ほぼpromoterではないDNAとみなせる．また，**4・1節**と同様に$Class_1$，$Class_2$，$Class_3$をpromoterと分類するルールと考えると誤答率は21となる．表4に

表 3 色つき有向グラフ上の類型パターンの帰納的学習

	データのみ利用	領域知識を利用
背反概念の学習	ID3流 分類ルールの学習 (文献 [Minton90],4.1章)	PRODIGY流 制御ルールの学習 (文献 [Minton90])
否背反概念の学習	COBWEB流のクラスタリングルールの学習 (文献 [Fisher87])	マクロ・ルールの学習 (文献 [Korf85],4.2章)
表現形式の生成	帰納学習用属性の生成 (文献 [Arikawa92])	階層的知識表現の生成 (文献 [吉田 92b])

表 4　ID3 と CLiP の比較

	ID3	CLiP
入力	• 属性別値のテーブル	• 色つき有向グラフ: ノードが属性に対応 色が値に対応 (左記テーブルを表現可能)
出力	• データの分類木	• 類型的パターン (分類木として利用可能)
他	• 途中まで作成した分類木に対して、情報量が最も増えるようなテストをノードとして追加	• 途中まで抽出したパターンに対して、グラフ中の出現頻度が高いノードを追加

$塩基_{15} = T \ \wedge \ 塩基_{16} = T \ \wedge \ 塩基_{17} = G \rightarrow Class_1$ (正例 26/ 負例 1)

$塩基_{15} = T \ \wedge \ 塩基_{16} = T \rightarrow Class_2$ (正例 9/ 負例 2)

$塩基_{16} = T \rightarrow Class_3$ (正例 8/ 負例 8)

$True \rightarrow Other$ (正例 10/ 負例 42)

図 11　クラスタリングルールの例

示した教授ありシステムの誤答率が 8〜19 であることを考えると，妥当なクラスタリングになっていることがわかる．

このようなクラスタリングが得られた本質的な理由は，promoter の強い規則性である．すなわち，データの強い規則性により，初めに「何か(実は promoter)の固まり」が図 11 の 3 番目のルールとして抽出され，次に，そのなかから，「さらに特徴的な固まり」(1〜2 番目のルール)が抽出されている．規則性があるデータからその規則性を抽出するのがクラスタリングシステムとすれば，定義どおりに動作していることになる．図 6 の結果は，教師からの指示に従い，さらに特徴的な固まりを三つに分類しているが，図 11 の実験では，そのような教師からの付加的な情報が得られず，結果として「固まり」の形が変わっている．

帰納的学習システムとして見たときの CLiP の問題点は，2 値表現形式(CLiP で帰納的学習を行うときに否定的条件を扱うため考案した表現形式)により学習した分類ルールの誤答率が高いことである．これは 2 値表現形式が直接表現形式に比べて，表現能力が高い反面，探索空間が大きくなり，より強力な探索能力を必要としていることによる．図 2 に示したアルゴリズムのなかで Pattern Modification 操作を否定的条件探索も行うように改良することは可能であり，帰納的学習能力向上の観点から有効と思われる．また，CLiP の帰納的学習能力向上が，その演繹的学習能力に与える影響の分析も興味深い研究課題である．

5・2　領域知識を利用した演繹的学習との比較

4・2 節において，CLiP をマクロルール[Korf 85]学習に適用した例を示した．表 5 に CLiP と EBL の比較を示す．CLiP を，PRODIGY[Minton 90]のような制御ルールの学習に適用することも可能と思われるが，実験において確認されてはいない．

表 5　EBL と CLiP の比較

	EBL	CLiP
入力	• 例題一つ • 領域理論 • Goal Concept • Operationality Criterion	• 複数の例題 • 領域理論 • 探索パラメータ (グラフサイズの評価式)
出力	• Macro Rule 一つ	• 複数の Macro Rules

PRODIGY は *SUCCEEDS*, *FAILS* など，制御ルールを区別して学習しているが，これは帰納的学習において背反概念(正例・負例)を学習していることに対応づけて考えることができる．このとき，マクロルール学習はクラスタリングシステム(否背反概念の学習システム)に対応づけて考えることができる．演繹的学習と帰納的学習両者の関係をより深く理解するうえで，PRODIGY 流の制御ルールの学習は興味深い CLiP の応用例であり，今後の研究課題である．

5・3　前研究との比較

1 章で述べたように，CLiP は，推論過程からの概念学習手法 CLiP[吉田 92a, 吉田 92b]として提案済みのアルゴリズムの使い方を一般化しただけであり，正確には再提案である．また，**4 章**で述べた実験も，上記文献で述べたプログラムを小さなメモリでも動くように改造した機能的には同じプログラムを用いて行ったものである．

前研究では定性推論結果の分析による概念抽出に主眼を置き，推論過程からのパターン抽出のみを考慮していた．本研究ではデータからのパターン抽出にまで対象を広げることで，パターン抽出という単純なアイディアにより今まで別々に研究が進められていた帰納的学習と演繹的学習を統合する枠組みが提供できることを明らかにした．すなわち本論文は，既存手法の再解釈方法を提案しただけであるが，再解釈により，帰納的学習と演繹的学習の対応関係を明確にし，両者の研究結果を相補的に使うための枠組みを与えることができたと考えている．

この枠組みにより，帰納的学習に従来演繹的学習が扱ってきた構造的なデータ(例えば推論過程)を扱うための基礎を与え，表現力を強化することができると考えている．また，演繹的学習においては，帰納的学習により得られてきた知見を使った，学習結果の評価や学習の効率化などの展望が開けると考えている．5・1

節，**5・2 節**で述べた今後の研究課題も，「帰納的学習と演繹的学習の両者は同じものという見方もできる」という認識をもとに今後の研究を加速しようという意図を持っている．

しかしながら，現在までの研究は枠組みを与えたレベルに留まっており，枠組みの利用に関して具体的な検討課題は多々残されている．今後の重要な課題である．

5・4 知識表現形式の生成

3・1 節において帰納的な学習の第1ステップがデータから抽出すべき情報(属性)の決定であることを述べた．**4・1 節**の実験ではこの部分は人手により行われている．すなわち，単なる文字列の並びである塩基配列情報を一般的な帰納的学習の問題に変換するために，何番目の塩基であるかを示す属性と，その値である塩基種類の組合せへの変換を人手によって行っている．

「よく現れるパターンは何らかの意味を持つ」というヒューリスティックは，このような表現形式への生成機能も持っていると思われる．表6に**4・1 節**の実験データをもとの文字列情報に戻し，CLiPで文字列(一本鎖となったグラフ)中によくあるパターンを抽出させた結果を示す．さらにデータを，表に示した文字列を含むか含まないかという属性を使って表現しなおし分類ルールを生成した結果，未知問題に対する誤答率が15の分類ルールが作成できた．

上記の例はCLiPのデータからの表現形式作成能力の一例である．また[吉田 92a, 吉田 92b]ではCLiPを用いて，定性方程式で表現された回路の物理レベルでの記述を，定性推論の結果を用いて論理レベルの記述などシステムには未知の記述に変換した実験を報告した．この実験は領域知識(この場合は定性推論の知識)を利用して新しい表現形式を生成した実験とみなせる．

5・5 他の統合的学習システムとの比較

機械学習の能力を人間なみに引き上げる研究の一環として，帰納的学習と演繹的学習の統合を試みた研究としては，[Lebowitz 86, Towell 90]などがある．CLiPとこれら研究事例との違いはCLiPが単一のアルゴリズムでさまざまな機能を持つことである．

[Lebowitz 86]ではあらかじめデータに対して帰納的な分析を行い，関係がありそうなデータを抽出した後，その関係の証明を試み，証明できた場合は演繹的な学習システムを起動する手法が提案されている．CLiPでは単一のアルゴリズムで帰納的な分析と演繹的な分析を同時に行える．図12に単一なアルゴリズムであることの利点を説明するための例として計算機の操作履歴解析結果を示す．

図12は，この論文を作成中に著者らが利用した文章生成のための計算機の操作履歴の分析思果である．著者らは文章清書システムとしてLaTeX(原稿であるtex fileを分析する主プログラム)を，文献データ検索システムとしてbibtex(LaTeXの出力から文献データを検索するプログラム)を，また，清書結果の表示システムとしてdvi2ps(LaTeXの出力から印刷用コマンドを作成するプログラム)およびghostview(印刷用コマンドを解析し端末に表示するプログラム)を利用している．LaTeXとbibtexの仕様から，文献データを処理するためにLaTeXコマンドが複数回起動されている．

図12に示した操作は，前半の(a)文献データの検索操作と，後半の(b)清書および表示操作に分割して考えることができる．また，論文作成初期の段階では，図12に示した処理全体を毎回実行していたが，後半の校正段階では参考文献はあまり変更しなかったため，(b)のステップのみ実行していた．

純粋な帰納的な学習手法では，このような操作履歴を分析するときに「突然来た電子メールを処理する操作などによるノイズ」の影響を必要以上に受けてしまう．また，純粋に演繹的な学習は，すべての操作やユ

表6 DNAデータから作成した帰納的学習用属性の例

正例から抽出した塩基列	
出現回数	塩基列
31	G T C
39	G T T
42	G T G
43	G G A
55	C C A
56	G G
61	G G C
80	T T
100	G C
102	A C
110	A T
112	C C
135	C T

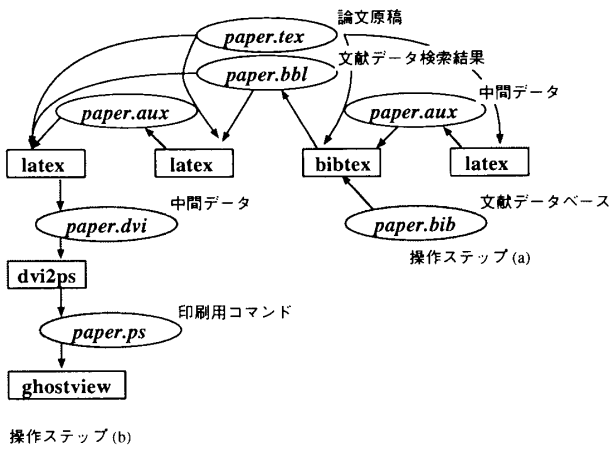

図12 操作履歴解析への応用

ーザ意図を解析可能な知識ベースが必要となり，実現が難しい*5．[Lebowitz 86]で提案されている方法を用いれば，清書するためのコマンド列として図12に示した操作全体と，後半の(b)のステップに相当する操作の二つを学習できると思われる．

CLiPでは(a)文献データの検索操作と(b)清書および表示操作，さらに，それらの複合操作としての3種類を学習可能である*6．このうち(a)の操作は[Lebowitz 86]のように帰納的分析と演繹的分析が独立しているシステムでは学習が困難であることが予想される．すなわち上記のような論文作成過程では，論文作成全般に全体の操作が何らかの固まりとして学習され，後半に(b)が別物として学習されることが予想されるが，後者(b)と前者(全体)の関係は別の仕組みによる解析が必要となってしまう．

また，[Towell 90]では不完全な領域知識をデータから修正する方法が提案されている．[Lebowitz 86]がデータからの学習の精度を領域知識を用いて向上しようとしたアプローチであったのに対し，[Towell 90]は逆に領域知識をデータを用いて良くしようとしたアプローチである．**2章**で説明したPatern Modification処理に，すでに持っているパターンを小さくしたパターンを生成する機能を付け加えれば，この機能もCLiPの枠組みのなかで実現可能*7と思われる．このような適用範囲の広さがCLiPの最大の特徴である．

さらに帰納的学習の第1，第2ステップを統合的に機械化しようとした研究として[Arikawa 92]があげられる．[Arikawa 92]では正規表現の生成機構をID3に統合したプログラムによりタンパク質データベースPIRを分析する手法が提案されている．具体的には，正規表現の生成機構が作成したアミノ酸列を例題のタンパク質が含むか否かという属性を使って，単なる文字列の並びであるタンパク質のアミノ酸配列情報を一般的な帰納的学習の問題に変換している．表6に示した実験は，この正規表現の生成機構に相当することがCLiPでも可能なことを示している．実際にPIRデータベースをCLiPで解析し帰納的学習用の属性を生成した結果を表7に示す．表中に「- -」というアミノ酸列がある．これは[Arikawa 92]において分類ルール作成時に生成・利用された正規表現と同一のものである．

表7 PIRデータから作成した帰納的学習用属性の例

負例から抽出したアミノ酸列		正例から抽出したアミノ酸列	
出現回数	アミノ酸列	出現回数	アミノ酸列
10362	* * -	30	* * + + * + *
13427	* * +	31	* * + +
18103	* -	37	* * + + + *
19463	- -	38	* * + + * + +
23659	* - -	192	* * + + * *
29628	* +	284	* * +
31179	- +	1031	* * + *
35782	+ +	2142	* *

帰納的学習や演繹的学習の第2ステップと第3ステップを統合し，一般化の程度を制御し例題への過剰学習を防ごうという研究もある．演繹学習の分野では属性の取り得る値の範囲を探る[Michalski 86]などの研究はこの範ちゅうの仕事とみなすこともできる．また，演繹的学習の分野において変数汎化の程度を制御する研究事例[Flann 89, 山村 89]もある．色つき有向グラフというデータ表現形式を採用すれば両者を関連づけて議論し，共通の学習原理を探索することも可能と思われるが，これに関しては今後の課題である．

6. まとめ

帰納的な分類ルールの学習と演繹的なマクロルールの学習の二つの学習機能を併せ持つアルゴリズムCLiPを提案した．CLiPでは「よく現れるパターンは何らかの意味を持つ」というヒューリスティック(typical pattern heuristic)に基づき，①特定のDNA(promoter)シーケンスの分類ルールの帰納的学習や，②1次方程式解法用マクロルールの演繹的学習ができることを計算機実験により示した．

このようなCLiPの汎用性は，「よく現れるパターンは何らかの意味を持つ」というヒューリスティックの妥当性と，色つき有向グラフの表現力によるものと考える．

本研究は，既存の領域知識を併用しながら大量データを解析する学習手法の研究の一環として，現在まで行われている帰納的学習と演繹的学習の統合を試みたものである．今後，最終的なねらいである「既存の領域知識を併用しながら大量データを解析する」技術の検討を進める．

＊5　例えば，文献に関する修正を行わなかったときには(b)の操作だけでよいことを，エディタの操作履歴から解析できる強力な知識ベースの作成は難しい．

＊6　本研究では，既存研究との比較実験を中心に行い，既存の方法で扱えないこのような例題の実験は行わなかった．実際に(a)(b)二つの操作をマクロ的な操作として抽出するにはグラフサイズの評価基準Cなどに若干の工夫を必要とする．しかし，この評価基準の与え方に関しては，[吉田 92b] 2・3節での検討結果が，そのまま利用可能である．

＊7　現在のままでは，データにより既存知識をより詳細化することはできるが，単純化することができない．

謝　　辞

4・2節の実験に用いた例題は，大阪大学の山田誠二氏の学位論文より引用させていただいた．また，九州大学の篠原　歩氏にはPIRデータベースをご提供いただいた．例題をご提供いただいたこと，日頃さまざまご教授いただいていることに対して，両氏に感謝する．

◇参考文献◇

[足立 88] 足立，西野：計算量理論概説，p. 131，朝倉書店 (1988).

[Arikawa 92] Arikawa, S., Miyano, S. and Shinohara, A.: Knowledge Acquisition from Amino Acid Sequences by Learning Algorithms, *JKAW92*, pp. 109-128 (1992).

[DeJong 86] DeJong, G. and Mooney, R.: Explanation-Based Learning: An Alternative View, *Machine Learning*, pp. 145-176 (1986).

[Efron 82] Efron, B.: The jackknife, the bootstrap and other resampling plans, *SIAM* (1982).

[Fisher 87] Fisher, D. H.: Knowledge acquisition via incremental conceptual clustering, *Machine Learning*, pp. 139-172 (1987).

[Fisher 88] Fisher, D. H. and Schlimmer, J. C.: Concept simplification and predictive accuracy, *Int. Conf. on Machine Learning*, pp. 22-28 (1988).

[Flann 89] Flann, S. and Dietterich, T. G.: A Study of Explanation-Based Methods for Inductive Learning, *Machine Learning*, pp. 187-226 (1989).

[Korf 85] Korf, R. E.: Macro-operators: A weak method for learning, *Artif. Intell.*, pp. 35-77 (1985).

[Lebowitz 86] Lebowitz, M.: Integrated Learning: Controlling Explanation, *Cognitive Science*, Vol. 10, pp. 219-240 (1986).

[Michalski 86] Michalski, R. S.: A Theory and Methodology of Inductive Learning, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (eds.), *Machine Learning*, pp. 83-134, Springer-Verlag (1986).

[Minton 90] Minton, S.: Quantitative results concerning the utility of explanation-based learning, *Artif. Intell.*, Vol. 42, pp. 363-391 (1990).

[Mitchell 84] Mitchell, T. M., Utgoff, P. E. and Baserji, R.: Learning by experimentation: Acquiring and refining problem solving heuristics, R. S. Michalski, G. G. Carbonell and Tom M. Mitchell (eds.), *Machine Learning*, pp. 163-190, Springer Verlag (1984).

[Mitchell 86] Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, S. T.: Explanation-Based Generalization: Explanation-Based Generalization : A Unifying View, *Machine Learning*, pp. 47-80 (1986).

[Mooney 86] Mooney, R. J. and Bennett, S. W.: A Domain Independent Explanation-Based Generalizer, *AAAI-86*, pp. 551-555 (1986).

[日経 91] 日経バイオテクノロジー最新用語辞典91，p. 580，日経BP社 (1991).

[Quinlan 86] Quinlan, J. R.: Induction of Decision Trees, *Machine Learning*, Vol. 1, pp. 81-106 (1986).

[Quinlan 90] Quinlan, J. R.: *Learning logical definitions from relations*, pp. 239-266 (1990).

[Towell 90] Towell, G. G., Shavlik, J. W. and Noordewier, M. O.: Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks, *AAAI-90*, pp. 861-866 (1990).

[Weiss 91] Weiss, S. M. and Indurkhya, N.: Reduced Complexity Rule Induction, *IJCAI-91*, pp. 678-684 (1991).

[山田 89] 山田：問題解決における戦略知識の学習，Ph.D. thesis，大阪大学 (1989).

[山村 89] 山村，小林：EBLの複数例題下への拡張，人工知能学会誌，Vol. 4，No. 4, pp. 389-397 (1989).

[吉田 92a] 吉田，元田：推論過程からの概念学習(1)類型的推論過程の抽出，人工知能学会誌，Vol. 7, No. 4, pp. 119-129 (1992).

[吉田 92b] 吉田，元田：推論過程からの概念学習(2)概念構造の構成要因，人工知能学会誌，Vol. 7，No. 4, pp. 130-140 (1992).

〔担当編集委員・査読者：原口　誠〕

著者紹介

吉田　健一(正会員)

1980年東京工業大学理学部情報科学科卒業．同年，(株)日立製作所に入社．同社エネルギー研究所にてプラントの異常診断などの研究に従事．1986年より，基礎研究所にて，知識表現，定性推論，機械学習などの研究に従事．博士(工学)．1984年日本原子力学会論文賞，1990年電気学会論文賞，1991年人工知能学会全国大会優秀論文賞，1992年人工知能学会論文賞受賞．情報処理学会，AAAI，ACM各会員．

元田　浩(正会員)

1965年東京大学工学部原子力工学科卒業．1967年同大学院原子力工学専攻修士課程修了．同年，(株)日立製作所に入社．同社中央研究所，原子力研究所，エネルギー研究所を経て，現在，基礎研究所主管研究長．原子力システムの設計，運用，制御に関する研究，診断型エキスパートシステムの研究を経て，現在は人工知能の基礎研究，特に機械学習，知識獲得，視覚推論などの研究に従事．工学博士．日本ソフトウェア科学会理事，人工知能学会理事，同編集委員，Knowledge Acquisition(Academic Press)編集委員，IEEE Expert編集委員を歴任．Artificial Intelligence in Engineering (Elsevier Applied Science)編集委員，人工知能学会知識ベース研究会主査，日本認知科学会編集委員．1975年日本原子力学会奨励賞，1977，1984年日本原子力学会論文賞，1989，1992年人工知能学会論文賞受賞．情報処理学会，日本ソフトウェア科学会，日本認知科学会，AAAI，IEEE Computer Society各会員．

Nitin Indurkhya
(ニティン インドゥルキア)

1991年ラトガース大学より計算機科学の学位を取得(Ph. D)．同年，日立製作所客員研究員，現在シドニー大学計算機科学講師．モデルの自動構築技法，隠れマルコフモデルを用いた時系列データ解析，ルールの帰納的学習，分類木学習，ニューラルネット，ノンパラメトリックな統計手法などを研究．