論 文

Technical Papers

逐次ペア拡張に基づく帰納推論

Inductive Inference by Stepwise Pair Expansion

吉田 健一* 元田 浩*,†
Kenichi Yoshida Hiroshi Motoda

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

1995年9月18日 受理

Keywords: (classification rule/macro rule/concept) learning, (constructive) induction, colored digraph.

Summary

Inductive learning, which tries to find rules from data, has been an important area of investigation. One major research theme of this area is the data representation language that the learning methods can use. The conventional rule learning methods use an attribute-value table as a data representation language, whereas inductive logic programming (ILP) uses the first-order logic. We propose colored directed graph as a data representation language for inductive learning methods. *Graph-based induction* (GBI) uses this data representation language. The expressiveness of graph stands between the attribute-value table and the first-order logic. Thus its learning potential is weaker than that of ILP, but stronger than that of the conventional attribute-value learning methods. The real advantage of GBI appears in the domain where the dependency between data bears the essential information. The behavior analysis of computer users is a typical example of such a domain. In this domain, the complex structure of dependency between the user tasks prevents us from using the conventional attribute-value learning methods, and ILP cannot meet the requirement for the efficiency.

In this paper, we explain GBI method and give experimental results. We also discuss the relationship between this new method and conventional inductive inference methods such as conventional classification rule learning methods, constructive induction methods, inductive logic programming methods, macro rule learning methods, and concept learning methods. While this list of the methods covers the wide area of inductive inference, we find that most of them can use *Stepwise Pair Expansion* as their basic algorithm. The use of the pairs and the representation language define the function and characteristics of each method. We also discuss the use of the statistical measures such as gini index and information gain index to realize various inductive inference functions.

1. はじめに

与えられた個々の事実から一般的な規則を導き出そうとする帰納推論は，一般的な規則から個々の事実を説明する演繹推論に対立する概念として人工知能研究における一大テーマであり，データ分類規則の学習 [Breiman 84, James 84, Quinlan 86]，マクロ・ルール獲得による推論の高速化 [DeJong 86, Korf 85, Mitchell 86]，抽象的概念の獲得 [Fisher 87, Holder 89, Pagallo 90]，データやトレースからのプログラム生成 [Bauer 79, Muggleton 92, Pazzani 92, Quinlan 90, Shapiro 83] など，種々の研究分野を含んでいる．

本論文では，これら多岐にわたる推論機能を統一的に実現するアイデアとして「データに含まれる類型的ペアの逐次拡張」を提案する．さらに「類型性」の指標として統計的評価尺度を用いる方法を提案し，データの表現形式として有向グラフを用いて分類規則学習を行う方法をその具体例として説明する．

以下2章では「データに含まれる類型的ペアの逐次拡張」というアイデアを説明し，データの表現形式

† 現在，大阪大学産業科学研究所

として有向グラフを用いた分類規則学習方法（Graph Based Induction 法）を例に「類型性」の指標として統計的評価尺度を用いる方法を述べる．さらに，3 章で利用者の次コマンド予測問題を使って Graph Based Induction 法の性能評価を行い，アイデアの妥当性を検証する．最後に 4 章で，提案したアイデアの帰納推論一般への適用を議論する．

2. 逐次ペア拡張と Graph Based Induction

「データに含まれる類型的ペアの逐次拡張」というアイデアは，著者らによる「推論過程からの概念学習手法」[吉田 92a, 吉田 92b] および「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」[吉田 95] など一連の研究の基本アイデアを抽象化し，帰納推論一般へ拡張したものである．特に本論文では統計的評価尺度を導入し，さらに類型パターン抽出アルゴリズムとデータ表現方法を分離考察することで，各関連研究の関連を明確にした（4 章参照）．

アイデアの中核は，「データの類型的なパターンを，ペアの逐次抽出により抽出する」というものである．ここで「ペア」としては，条件と結論のように推論規則を構成するような「条件と結論のペア」だけでなく，「相関の高い共起関係にあるデータのペア」も扱う．また，「類型的」は「データ中によく現れる」という直観的な概念を意図しているが，「類型」性の評価に統計的な指標を導入する．

2・1 共起関係にあるデータからのペア抽出

図 1 を用いて，データの表現言語としてグラフを用いた場合を例に，何らかの外界の情報を表す入力データから，「相関の高い共起関係にあるデータのペア」を抽出する過程を示し，類型的ペアの「逐次拡張」というアイデアを説明する．ここで，「ペア」は「逐次拡張」されることにより複雑なパターンを構成するので，逐次拡張によりデータから抽出されたものを「類型パターン」または「抽出パターン」と呼ぶ．

図 1 では，すでに二つの類型パターン（A, B）が逐次拡張処理によりデータから抽出されていると仮定している．「逐次ペア拡張」は次の 3 ステップを繰り返し行うことで実施する．

Step 1. 入力データの中で抽出パターンと同じパターンがあれば，これを一つのノードに書き換える．

Step 2. 書換え後のデータに含まれる，二つのノードの組合せからなるすべてのペアを抽出する．

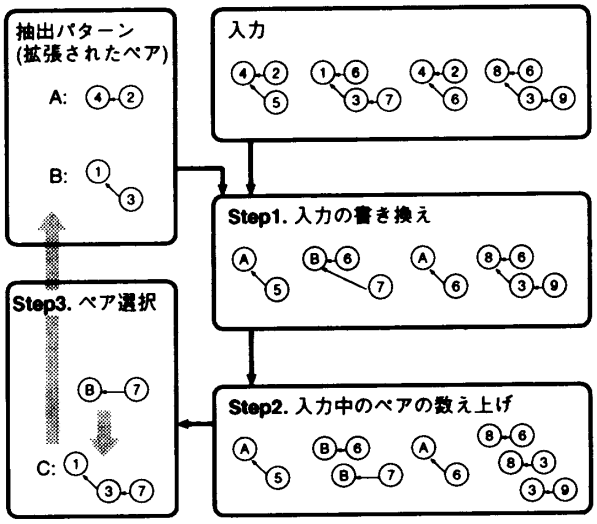


図 1 逐次ペア拡張アルゴリズムの基本アイデア

Step 3. 抽出したペアのうち，最も「類型的（次節参照）」なペアを一つ選び抽出パターンとして登録する．このとき，ペアを構成するノードが Step 1 で書き換えられたノードであれば，もとのパターンに復元してから登録する．

類型パターンが何も抽出されていない状態から始めて上記 3 ステップを繰り返すことで，ペアを逐次拡張し，データに含まれる特徴を類型パターンとして抽出することができる．

2・2 条件と結論に関するペア抽出

前述のアルゴリズムでは Step 3 における「類型性」の評価基準が重要である．本研究では統計的指標の利用を試みた．この利用方法を説明するため，図 1 に示した処理過程を，分類規則の抽出処理過程として再解釈したものを 図 2 に示す．分類規則学習アルゴリズムとして見た「逐次ペア拡張」処理は次の 3 ステップより構成されていると見なせる．

Step 1. 入力したケース・データを途中まで作成した分類規則（図 2 では分類木）で分類する．ここで，抽出パターンを分類木の条件，入力グラフの根ノードの色情報をケース・データのクラス情報，それ以外のノードの色情報を属性情報と見なしている．

Step 2. 個々の分類結果をさらに分類するために，分類用属性表を各葉ノードにおいて作成する（具体的な作成方法は 3 章参照）．

Step 3. Gini Index [Breiman 84], Information Gain [Quinlan 86] といった統計的指標により新規のテスト条件を選び，新たなテスト条件として

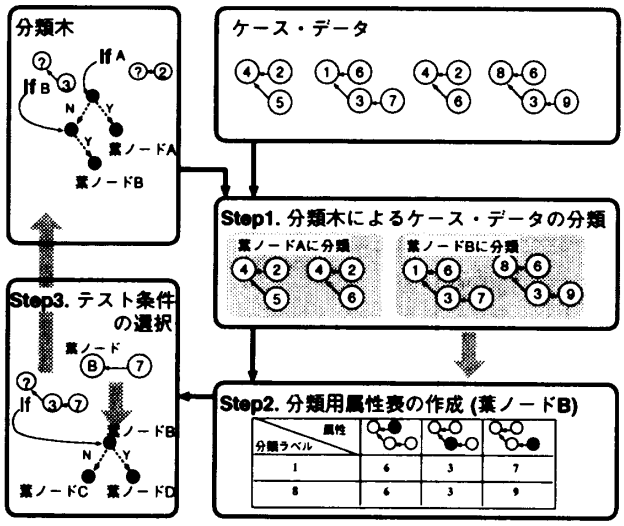


図2 分類規則学習アルゴリズムとして見た逐次ペア拡張アルゴリズム

分類規則に追加する.

上記処理過程は，Step 2 を除いて [Breiman 84, Quinlan 86] などの固定した属性表を用いた分類規則学習と同じであり，Step 3 の新規テスト条件の選択には同じ指標を用いることができる．すなわち「条件と結論のペア」だけでなく,「相関の高い共起関係にあるデータのペア」の抽出においても，データの一方を結論，共起関係にある他方のデータを条件と対応づけて考えれば，同じ指標で 図1 Step 3 における「類型性」の判定を行える.

図2 では 図1 と異なり，Step 1 のパターンマッチの時に，抽出パターンの根ノードの情報を無視している．例えば 図1 抽出パターン A: 4←2 は，図2 ではパターン: ?←2 として根ノードの情報 4 を無視して扱われている．根ノードの情報をどう用いるかは「逐次ペア拡張」の持つ具体的な帰納推論機能を決める重要な点である．これについては 4・2 節で考察する．また，図1，図2 はデータの表現形式にグラフを用いているが,「逐次ペア拡張」というアイデア自体は異なったデータ表現形式にも応用可能である（4・1 節参照）．特に 図2 に示した例はデータ表現形式としてグラフを用いた点で新規な分類規則学習方法であり，著者らはこれを Graph Based Induction（GBI）法と呼んでいる*1.

*1 実際には GBI 法は分類規則学習以外の帰納推論機能を持っている（4・2 節参照）が，この論文では説明の都合上 GBI 法の意味を分類規則学習に限定して用いている.

3. Graph Based Induction 法による分類規則の学習

前章でデータ表現形式としてグラフを用いた分類規則学習方法である GBI 法を提案した．この方法はデータ表現形式として，従来の固定的な属性表ではなく，グラフを用いている．データ表現能力を考えた場合，グラフは固定的な属性表 [Breiman 84, Quinlan 86] と Inductive Logic Programming（ILP）[Muggleton 92, Pazzani 92, Quinlan 90, Shapiro 83] が用いている述語論理の中間に位置する（4・1 節参照）．したがって，固定的な属性表ではうまくデータを表現できないが，ILP ほど強力な枠組を必要としないような問題領域では，探索空間が広すぎることによる学習効率の低下を招かずに，従来法より学習精度を向上させることが期待できる.

本章ではそのような応用問題の一例として計算機インタフェースのユーザー適応機能について実験結果を含めて考察し,「逐次ペア拡張」という考え方と問題領域に合わせたデータ表現形式を組み合わせることで，効率的な規則学習が行えることを示す．なお，本章で述べる技術を使った計算機インタフェース・システム ClipBoard の詳細については [Yoshida 96] を参照されたい.

3・1 アプリケーション選択問題

UNIX の上で emacs エディタと latex ドキュメント・プロセッサを組み合わせて文章を作成する状況を考える．この場合，ドキュメントの原稿が記憶されたファイル（例えば paper.tex）に対し，交互に emacs と latex を使った操作を施すのが普通である．また，このとき次の操作に用いるコマンドの選択はユーザーごとに異なり，作業の進捗状況に応じても変化する.

図3 に，ユーザーがプレビューアで内容確認（Step A）したドキュメントの内容を emacs で修正した後，latex で清書し，別のプレビューアで修正結果を確認（Step B）した後，結果を印刷（Step C）した場合に，ClipBoard が OS から入力として受け取る計算機利用履歴を示す．ClipBoard は，このような履歴から，次に作業に必要となるアプリケーション・プログラムを選択する規則を学習し，各ファイルのサフィックスごとにルール化する機能を持っている.

図3 に示した例は，サフィックスが dvi のファイル用アプリケーション選択ルールを学習する問題となっている．従来のユーザー適応機能を持ったインタフェー

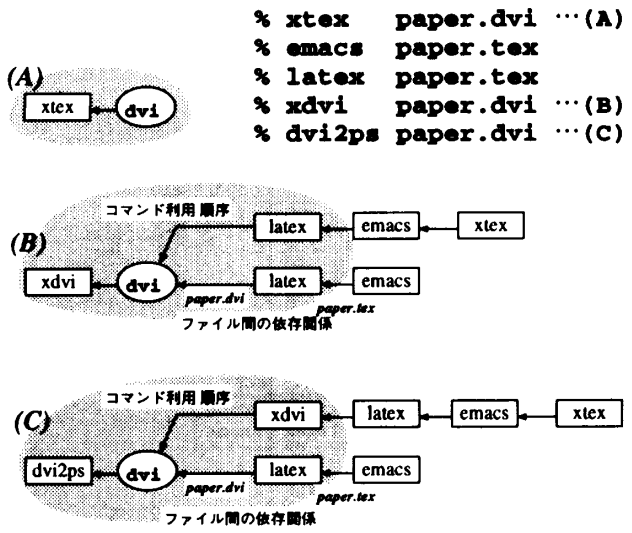


図3 計算機利用履歴のグラフ表現

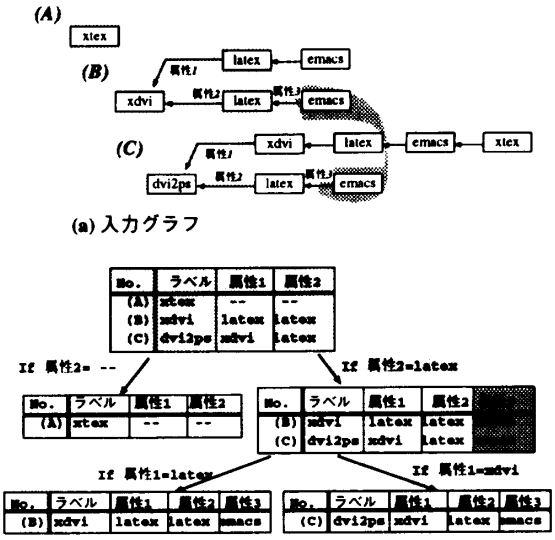


図4 GBI 法による分類規則学習過程

スの研究（例えば [Cypher 91, Dent 92, Greenberg 88, Hermens 93, Maes 93, Masui 94]）では，コマンド利用順序のような構造を持たず単純な属性表で表現可能なデータのみを解析しているものが多かった．図3ではそのような順序情報と共に「paper.dvi ファイルは emacs で作成された paper.tex から latex により作成された」といった，一般的には通常の属性表で表現できない複雑な構造を持った各ファイルの依存関係も入力としている．具体的には，図3(C) において，上段 xdvi←latex←emacs←xtex は，履歴中 dvi2ps で dvi ファイルの内容を印刷する前のコマンド利用順序を示しており，下段の latex←emacs は，ファイル間の依存関係を示している．このファイル間の依存関係は，OS に各アプリケーション・プログラムの実施する I/O オペレーションの履歴を作成させることで，簡単に収集可能である．

3・2 分類規則学習過程

図4 に GBI 法による分類規則学習過程を示す．サフィックスが dvi のファイル用アプリケーション選択ルールを学習するため，ClipBoard は 図3 に示したグラフを 図4(a) に示したケース・データに変換する．この場合，図3 のグラフはすべて dvi ファイルに関するものなので，図4(a) の入力グラフ（分類規則学習問題におけるケース・データ）は，根ノードに直接つながった dvi ファイルに関するノードが削除された．その他は 図3 のグラフと同一のグラフである．

逐次ペア拡張による分類規則の学習プロセスは，図1，図2 にアイデアを示したものによる．図4(b) は，この過程を若干詳細に説明している．初めは何も類型パターンが抽出されていない状態で入力中のペアの数え上げが行われる（図1 Step 2）．分類規則学習として見た場合この処理は，すべてのケース・データからなる集合を分類する最初のテスト条件を選ぶための前処理（図2 Step 2）に相当し，葉ノード（この場合はすべてのケース・データを含む）において属性表が作られる．この表は仮想的なもので，実際には属性表を作るのと等価と見なせる処理により，データ中の統計的な情報が収集される．具体的には，初めの属性表は各ケース・データの根ノードの情報と，根ノードに直接つながっているノードの情報を使って作られている（図4(b) 最上段の表）と見なせる．

テスト条件の選択（図2 Step 3）には通常の分類規則学習と同じ評価指標を用いる．ClipBoard では CART [Breiman 84] で使われている Gini Index を用いている．Gini Index は，新しいテスト条件で分類を行った後の葉ノードのクラス構成をなるべく純粋なものにしようとデザインされた評価指標で，各葉ノード t に分類されるデータがクラス i に属する確率を $p(i|t)$ とした時に

$$\sum_{i \neq j} p(i|t)p(j|t)$$

を小さくするよう，テスト条件を選択する．ClipBoard で Gini Index を用いたのはプログラミングが簡単で計算時間も短いという理由のみであり，この研究の趣旨

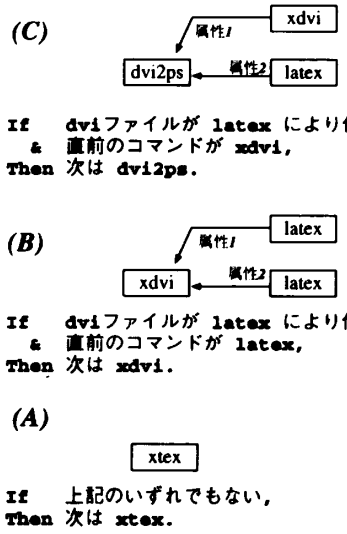
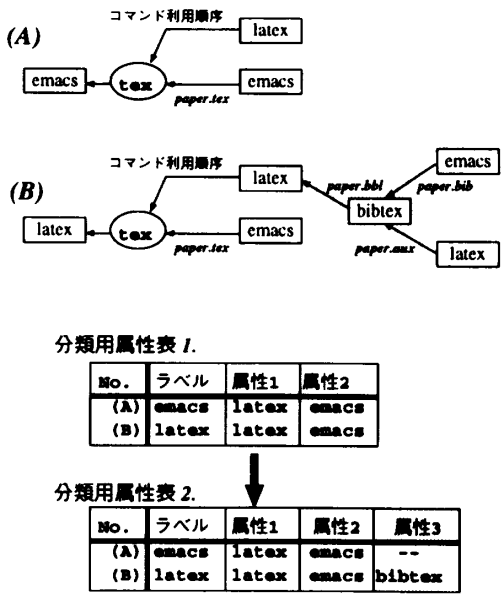


図5 抽出パターンの分類規則としての解釈



分類用属性表 1.

No.	ラベル	属性1	属性2
(A)	emacs	latex	emacs
(B)	latex	latex	emacs

分類用属性表 2.

No.	ラベル	属性1	属性2	属性3
(A)	emacs	latex	emacs	--
(B)	latex	latex	emacs	bibtex

図6 逐次ペア拡張による分類用属性の拡張

からすればInformation Gain [Quinlan 86] のような別の指標でも本質的には変わらない.「類型的」や「データ中によく現れる」といった直観的な概念に，客観的な評価基準を与えることが，ここでの眼目である．

図4(b)では属性2がテスト条件として選択され，クラスxtexが他の二つのクラスから分別されている．こうしてテスト条件が一つ選択されると，次のStep 1では選択済みの分類条件によるケース・データの分類が行われる．逐次ペア拡張によるパターンの抽出処理としてみた場合，この処理は「any ← 属性2」というペアにより書換えが行われることに相当する．一方，分類規則学習処理として見た場合，書換えが行われたケース・データが 図4(b) 中段右の属性表を形成する集団に，それ以外が 図4(b) 中段左の属性表を形成する集団に，それぞれ分類されたことに相当する．

いったんテスト条件が抽出された後は，Step 2において，Step 1でパターンの書換え処理が行われたケース・データごとに属性表を作成し直す．この属性表の書換えは，それまでの処理でテスト条件として選択された属性（図4(b)の例では属性2）に対応するノードに直接つながったノード（図4(b)の例では属性3）を新しい属性として表に追加することにより行う．この新しい属性の属性表への追加は，Step 1のパターンの書換え操作により自動的に行うことができる．すなわち，それまでに抽出されたテスト条件を構成するパターン中のノードは書換え後には一つの根ノードに書き換わっているので，Step 2では常に根ノードと根ノードに直接つながったノードのみ処理を行えば，自然に属性表への新しい属性の追加が行われることになる．

最終的にケース・データが分類できた後，類型パターンを分類規則として取り出す．このとき，取り出されるパターンは 図4(b) の各末端ノードに到達するまでの分類条件をまとめたものとなっており（図4(c)），dviのファイル用アプリケーション選択ルールとして解釈可能である（図5）．

図3のような単純な例では 図4(b) の処理中に属性表に追加される属性は結果として不要であったが，図6に示したような複雑なケースでは，この処理により追加する属性が必要となる．図6はlatex用文献処理システムであるbibtexプログラムを使った文章処理過程（latexを繰り返し使用）と，使わない処理過程（latexとemacsを交互に利用）を，追加された属性により分類している．

ここで，図6属性3が表に組み込まれるには，その前の段階で属性2が表に組み込まれている必要がある．属性2は通常他のデータ（図6では省略）を分類するための属性として使われていることが多いので，この組込み処理はうまく働くのが普通である．また現在のGBI法のインプリメントでは，何も使える属性がなければ，無作為に一つ属性を追加するようになっている．GBI法のデータに対する重要な仮定として「グラフの枝で接続された属性間は，何らかの重要な関連を持っている」がある．言葉を変えれば，GBI法ではこのような仮定が満たされ「無作為に属性が追加されることよりは，他データの処理のため必要な属性が自然に追加されていくことが多い」と期待している．後述の実験結果を見る限りでは，実際のデータ処理において上記のような考え方は有効に働いている．

表1 コマンド選択精度の比較

手法	def.	Rule	LD	1-NN	Cart	GBI
精度%	22.6	20.7	22.6	20.8	34.6	57.8

def.: 最も使われるコマンドを次コマンドとして予測
Rule: 一つ前のコマンドを次コマンドとして予測
LD: 線形識別法
1-NN: 1 Nearest Neighbor 法
CART: 分類木学習方法

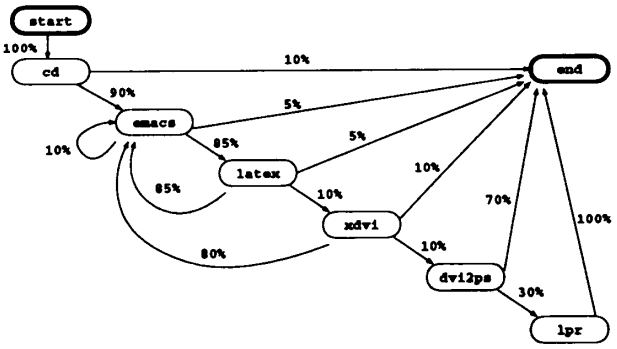


図7 ユーザーの挙動モデル

3・3 実験結果

表1に，GBI法と既存の分類規則学習法のアプリケーション選択ルールの学習問題に対する学習精度比較を行った結果を示す．実験に用いたデータは著者の1人が3ヵ月間計算機を使用した履歴（コマンド数にして約2,000，プロセス数約9,700，I/Oオペレーション数約127,000）であり，全履歴データのうち2/3から，各々の方法により過去の操作履歴に基づき次のコマンドを予測する分類規則を学習し，残り1/3を用いてその分類精度を比較した．

表1でGBI法以外はすべて固定的な属性表を用いた方法であり，I/O情報のような構造的なデータを効率的に扱えないため，これらの方法はコマンド利用順序のみから分類規則を学習した．既存手法でも構造的なデータを扱える手法として代表的なILPシステムの一つであるFOCL[Pazzani 92]による実験も試みたが，Macintosh LC630（68LC040，33MHz CPU，12Mbyte DRAM）上のFOCLで4時間実験を継続しても幾つか必要な分類規則のうち初めの一つのみ取り出せただけであったので，単純にこの方法を用いたのでは必要となる処理時間が長過ぎて解を発見できないと判断し，精度比較をあきらめた．

表1でRuleは[Greenberg 88]で報告されているヒューリスティクスによる．既存手法で最も分類精度が良かったのはCARTであるが，GBI法はデータ表現が異なるだけで概ね同じアルゴリズムを使用している（どちらも分類木の初期作成にGini Indexを使用し，過剰学習防止にcross validation技法を使用）にもかかわらずCARTより大幅に優れた分類精度を示している．これは両者の用いたデータの持つ情報量の差による．また，FOCLが処理時間の関係で精度評価ができなかったのは，述語論理という強力すぎる枠組による膨大な探索空間を探索できなかったことによる．アプリケーション選択ルールの学習問題では，I/O関係がGBI法に探索をガイドする強力な道案内を提供しており，ClipBoardはパソコン上でも動く軽い処理として実現されている[Yoshida 96]*2．

3・4 補足実験

3・3節で示した実験結果は1人のユーザーの挙動を分析した結果であり，被験者が計算機を使い慣れていない場合は単純な分析だけで次のコマンドを選択できる可能性がある，割込みの多い作業環境にいる被験者の挙動はノイズの影響が多く意味のある分析が困難，といった問題点がある．

このような問題を除去し，提案手法によりどの程度の問題に対して意味のある分析ができるかを調べる目的で，シミュレーションを用いた補足実験を行った．シミュレーションでは，計算機を用いて作業中のユーザーはマルコフ・モデルに従い次のアプリケーションを選択すると仮定した．図7はユーザーが文章作成をする過程をマルコフ・モデルで表現した例である．図では，emacsを使って文章の編集作業を終えたユーザーは確率85%で次にlatexを選択し，10%の確率で再びemacsを選択し，残り5%の確率で作業を終了するなどと仮定した．シミュレーションではこれ以外にもコンパイラを用いたプログラム作成など5種類のマルコフ・モデルを用意し，ユーザーは各モデルに基づく作業をランダムに選択するとして操作履歴を生成し，生成した履歴の2/3を使って学習し，1/3を使って学習

*2 表1において最も精度の良い予測方法であるGBI法でも，過去の履歴から次コマンドを予測した場合の正解率は低い．これは過去の操作履歴には次コマンドのファイル入力に関する情報はないことを考慮し，GBI法の入力からも次コマンドの入力情報を削除したことによる．具体的には，図3中，グラフの根ノードにはコマンド利用順序に対応する枝のみ残し，代わりにコマンド利用順序に対応する枝中のコマンドごとに，各コマンドに関するファイル間の依存関係を示す副枝をつけた．このようにすると，次コマンドの予測はClipBoardのアプリケーション選択より難しい問題となり，ClipBoardの実際のアプリケーション選択精度はこの表に示したものより高い．詳しくは[Yoshida 96]を参照されたい．ここでの要点は，一つ前までのコマンド間の依存関係だけでも，予測精度が向上することである．

表2 補足実験結果

(a) 精度の比較

ノイズ	精度		精度の向上度			
	def.	GBI	LD	1NN	Cart	GBI
15%	35.5	51.5	0	3	37	45
20%	33.8	52.1	0	6	34	54
25%	33.0	47.2	−2	−10	26	43

(b) 理論的上限値との比較

ノイズ	上限値	精度 / 上限値(%)				
		def.	LD	1NN	Cart	GBI
15%	58	62	62	63	85	90
20%	54	62	62	66	83	96
25%	50	65	64	59	83	94

した分類規則のコマンド選択精度を評価した．

表2に実験結果を示す．実験では上記以外に，ユーザーは各コマンドを実行する前に特定の確率でマルコフ・モデルにないコマンドを選択すると仮定した．これはmailなどの割込みにより作業が中断することを模擬した一種のノイズで，この確率を15～25%に変えて各方式の結果を比較した．表2(a)は最も使われるコマンドを次コマンドとした場合と比較して，各方式によるコマンド選択精度の向上を示している．また，このシミュレーションによる実験ではシミュレーションに使用したモデルを使えば最適なコマンド選択ルールを構成することが可能であり，表2(b)は最適解を上限として，各方式が上限の何パーセントの精度を達成したか示している．

シミュレーションによる結果も実際の履歴データに基づく実験結果と同様GBI法が他の手法より優れた結果を示している．言葉を換えれば，単純に操作順序を使った方法では 図7のようなマルコフ・モデルを分析することは難しいが，GBI法はグラフというデータ表現形式を用いることにより，データに特徴的なパターンを抽出できることを示している．

4. 帰納推論と逐次ペア拡張

本論文で述べている「逐次ペア拡張」というアイデアは特に目新しいものではない．しかし，この単純なアイデアをベースに種々の考察を行うことにより，様々な帰納推論研究間の関係が明確になってくる．本章では，「データ表現言語」，「推論機能」，「探索手法」の三つの観点から種々の帰納推論の関係の再整理を試みる．また，最後に「逐次ペア拡張」というアイデアが実際にどのような場合に適用可能であるか考察する．

```
Algorithm 分類規則学習
 variable
    G_in : ケース・データ
    T    : 分類規則
 begin
  T ← ∅
  repeat
    T による G_in の分類
    Proc. 1, 2, 3 による分類用属性表の作成
    新規テスト条件の選択と，T への追加
 end

Proc. 1 伝統的分類規則学習法
   常に同じ属性表を使用

Proc. 2 Graph Based Induction 法
   新しい属性をグラフの中から追加

Proc. 3 Inductive Logic Programming 法
   新しい属性を論理式の集合の中から追加
```

図8 分類規則学習手法の比較

4・1 データ表現言語に関する考察

図8に，あらかじめ固定された属性表を用いる分類規則学習法，GBI法，ILP法の比較を示す．著者らはこれらは，データの表現言語は異なるが，分類規則に逐次新しいテスト条件を付け加えていくことで実現できるという点で同じ側面を持った手法であると考えている．また，2章でアイデアを示したように，このテスト条件の逐次追加は「逐次ペア拡張」というアイデアで説明する（または解釈し直す）ことができる．すなわち，ほとんどの手法は新規テスト条件として分類規則に追加するのは一度に一つのみであり，各々のデータ表現形式が許す範囲内で結論と条件のペアを考え，それを逐次拡張していると解釈できる．例えば伝統的分類規則学習方法であれば，常に同じ属性表からテスト条件を選択している．またILP法では，論理式の集合の中からテスト条件を追加している．

データの表現能力を考えた場合，グラフは固定的な属性表と論理式の中間に位置する．前章で実験に用いたアプリケーション選択問題は，固定的な属性表ではうまくデータを表現できないが，論理式ほど強力な枠組を必要としないような問題領域の例である．各手法の関係を上記のように理解すれば，問題の性質（問題を記述するのに必要なデータ表現能力）に合わせて適切な手法を選ぶのは単純な問題であるが，実用的なシステムを作ろうとする場合には大切な考慮点となる．

また近年ILPの世界でも，PAC learnabilityのように最悪ケースを想定した理論的研究だけでなく，U learnabilityの研究 [Muggleton 94] のように平均的な

ケースを想定し，古くから固定的な属性表を用いた学習方法でとられているアプローチにより学習効率をあげようとする動きもある．上記の対応関係を前提とすれば，さらに議論を進め「述語論理を用いたデータ表現と統計処理で扱うデータ表現の関係と，両者を考慮にいれた学習アルゴリズム」といった今まで十分扱われていなかった領域の考察も興味深い．ただし，述語論理と統計処理が前提とするデータ表現の対応関係を考察することは今のところ困難であり，今後の研究課題である．

4・2 推論機能に関する考察

帰納推論の範疇にはデータ分類規則の学習，マクロ・ルール獲得による推論の高速化，抽象的概念の獲得，データからのプログラム生成など，種々の研究分野が含まれている．表3に種々の帰納推論研究を，データ表現言語と推論機能の二つの観点から整理したものを示す．

著者らの一連の研究 [吉田 92a, 吉田 92b, 吉田 95] は，データ表現言語としてグラフを用いた場合に，逐次ペア拡張により，データの分類規則の学習や推論の高速化，概念獲得が行えることを示した例となっている．この中でデータの分類規則の学習は,「条件と結論のペア」を逐次拡張し，拡張処理の過程では根ノードの色情報を無視して処理を行った．これは抽出パターンを使う時に，クラス情報が未知のデータと抽出パターンを比較することによる．一方，推論の高速化と概念抽出では，共起関係にあるデータからペアを抽出するため，根ノードの情報も無視せずに処理を行った．これは共起関係にあるデータからのペア抽出では根ノードと葉ノードの区別が便宜的なものであり，実際に抽出パターンを使う時には，どちらの情報も入手できると考えたからである．すなわち，ペアの逐次拡張処理中に根ノードの情報を無視するかしないかは，抽出パターンの使い方，何をしたいかという推論の目的・機能により判断すべきものである．

固定属性の属性表では推論過程を表現することは困難であり，表という表現形式は推論の高速化には向かないと考える．[Pagallo 90] などの分類規則を表現するための中間概念を生成する方式を見直してみると，生成される中間概念は分類規則のテスト条件として採用された複数の属性を組み合わせたものであり,「逐次ペア拡張」で生成可能であると考える．実際に ClipBoard に組み込んだ GBI プログラムにより [Pagallo 90] で扱われている例題を再実験し，単純なものは再現可能であることを確認した．ただし，ClipBoard の GBI プログラムは中間概念生成に利用した属性をもとの属性表から削除してしまい，複製障害（replication problem [Pagallo 90]）を発生するため複雑な例題は再現できなかった．中間概念生成に利用した属性をもとの属性表から削除しないようにプログラムを改造することは容易であるが，実際の実験による確認は今後の課題である．

[吉田 95] では，推論過程をグラフ表現し逐次ペア拡張によりマクロ・ルールを抽出すれば推論の高速化が可能であることが述べられている．推論過程自体は論理式で表現することが自然であり，述語論理を表現言語とした逐次ペア拡張による推論の高速化という技法も考えられる．また，ILP 研究における [Pagallo 90] に相当する研究として，中間的な述語の生成研究 [Muggleton 89] もあるが，ここにも逐次ペア拡張という技法は適用可能に思われる．

4・3 探索手法に関する考察

[吉田 92a] では，データ中に暗黙のうちに含まれる抽象概念の探索に [Goldberg 89] と類似の並列探索技法を用いた．この方法で抽出される個々の概念は，基本的には逐次ペア拡張により作成されたものである．図1，図2に示したような局所的な情報に基づく探索技法は，ローカルな極小点に落ち込み大局的な最適解を得られない場合があることは広く知られており，[Breiman 84, Quinlan 86] のような手法が必ずしも常に良い規則を学習できないことの一つの理由になっている．

人工知能研究において [Goldberg 89] のような種々の探索技法は一つの大きな研究領域を形成している．問題によっては探索空間の形状が複雑で，単純な探索手法ではうまく答えを見つけられないケースも存在すると考えられるが，どのような場合にどのような探索手法を組み合わせて用いるかという検討は，帰納推論研究の枠組の中でも今後重要性を増すと考える．

4・4 適用範囲に関する考察

表3に示したように，逐次ペア拡張というアイデアは，データ分類規則の学習，マクロ・ルール獲得による推論の高速化，抽象的概念の獲得，データからのプログラム生成など，種々の推論機能の実現に応用可能である．

しかし，このことは逐次ペア拡張により帰納学習問題のすべてが解けることを意味しているわけではない．例えば，3章で扱ったアプリケーション選択問題において，計算機内部の履歴データをグラフ表現に変換したのは人間（正確には人間により特別に設計されたプ

表3 帰納推論手法の分類

	固定属性	有向グラフ	述語論理
データの分類	[James 84] [Breiman 84] [Quinlan 86]	[吉田 95] 本論文3章	[Shapiro 83] [Quinlan 90] [Pazzani 92] [Muggleton 92]
推論の高速化	–	[吉田 95] [Numao 95]	[Mitchell 86] [DeJong 86] [Korf 85]
概念獲得	[Fisher 87] [Pagallo 90]	[Holder 89] [Holder 92] [吉田 92a] [吉田 92b]	[Muggleton 89]

ログラム）であり，実世界のデータを計算機上でどう表現するかという問題を解いているのは人間である．3章で例として取り上げたデータ表現形式にグラフを用いた分類規則の学習も，実世界のデータが単純な表形式に変換できない性質があった場合にのみ有効な方法であり，表形式に変換可能な問題であれば，データ表現に表を用いたCARTのような手法を用いるのが妥当である．

逐次ペア拡張というアイデアは，実世界のデータをどう表現するかという問題が解かれた後に，採用されたデータ表現上での規則抽出を実現するアイデアにすぎない点を明記しておく．すなわち，逐次ペア拡張というアイデアは，実世界のどのデータをどのような形で表や論理式に変換するかの問題に解答を与えるアイデアではない．

帰納学習過程において，データ表現の決定と，そのデータ表現を用いた規則抽出は，必ずしも上記のように綺麗に分割して議論できる性質のものではない．しかし，規則抽出の部分を逐次ペア拡張という単純なアイデアで理解することにより，どのデータ表現を用いれば学習効率が良さそうかといった見通しは立てやすくなると思われる．主観的報告に留まるが，著者らの一連の実験（3章や[吉田 92a, 吉田 95]で扱った例題）で使用したデータ構造は，分類規則学習（3章），定性推論[吉田 92a]，EBL[吉田 95]など関連研究の研究者であれば，容易に理解可能な表現となっており，実世界のデータからの変換も容易に推察できる範囲であると考える．

5. おわりに

本研究では，帰納推論の多岐にわたる推論機能を統一的に実現するアイデアとして「データに含まれる類型的ペアの逐次拡張」を提案した．さらにデータの表現形式として有向グラフを用いた分類規則学習方法（Graph Based Induction，GBI法）を例に，「類型性」の指標として統計的評価尺度を用いる方法を示した．

アプリケーション選択問題を性能評価問題として，GBI法があらかじめ内容が固定された属性表を用いる分類規則学習手法より，分類精度において大幅に優れている問題領域があることを示した．この分類精度の向上はデータ表現能力の差に起因し，従来使われていた属性表や論理式の中間的なデータ表現言語として有向グラフが有望であることを示唆している．

提案したアイデアの帰納推論一般への適用を議論し，種々の帰納推論が，データの表現言語が異なるだけで「類型的ペアを逐次拡張してパターンを抽出することで実現可能である」という同じ側面を持つことを示し，関連研究分野の対応による分野間でのアイデアの融通を可能とした．本研究のアイデアを用いれば，一見雑多に見える帰納推論の諸分野に統計的評価尺度を分析の足掛かりとして与えることができる．人工知能の研究が中心となり開拓した個別のデータ表現言語と，各言語を使い表現された探索問題としてとらえ直した個々の帰納推論の問題に，統計的な分析手法を持ち込むことが今後の研究課題である．

謝　辞

本論文の初稿に丁寧なコメントをいただいた(株)日立製作所基礎研究所杉本晃宏氏，竹内　勝氏，岩山　真氏に感謝します．

◇参考文献◇

[Bauer 79] Bauer, M.A.: Programming by Examples, *Artificial Intelligence*, pp.1–21 (1979).

[Breiman 84] Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J.: *Classification and Regression Trees*, Wadsworth & Brooks/Cole Advanced Books & Software (1984).

[Cypher 91] Cypher Eager, A.: Programming Repetitive Tasks by Example, *CHI'91*, pp.33–39 (1991).

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

[Dent 92] Dent, L., Boticario, J., McDermott, J., Mitchell, T. and Zabowski, D.: A Personal Learning Apprentice, *AAAI-92*, pp.96–103 (1992).

[Fisher 87] Fisher, D.H.: Knowledge Acquisition via Incremental Conceptual Clustering, *Machine Learning*, pp.139–172 (1987).

[Goldberg 89] Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley (1989).

[Greenberg 88] Greenberg, S. and Witten, I.H.: How Users Repeat their Actions on Computers: Principles for Design of History Mechanisms, *CHI'88*, pp.171–178 (1988).

[Hermens 93] Hermens, L.A. and Schlimmer, J.C.: A

Machine-Learning Apprentice for the Completion of Repetitive Forms, *Proc. 9th Conf. on Artificial Intelligence for Applications*, pp.164–170 (1993).
[Holder 89] Holder, L.B.: Empirical Substructure Discovery, *ML-89*, pp.133–136 (1989).
[Holder 92] Holder, L.B., Cook, D.J. and Bunke, H.: Fuzzy Substructure Discovery, *ML-92*, pp. 218–223 (1992).
[James 84] James, M.: *Classification Algorithms*, A Wiely-Interscience Publication (1984).
[Korf 85] Korf, R.E.: Macro-operators: A Weak Method for Learning, *Artificial Intelligence*, pp.35–77 (1985).
[Maes 93] Maes, P. and Kozierok, R.: Learning Interface Agents, *AAAI-93*, pp.459–465 (1993).
[Masui 94] Masui, T. and Nakayama, K.: Repeat and Predict – Two Keys to Efficient Text Editing, *CHI'94*, pp.117–123 (1994).
[Mitchell 86] Mitchell, T.M., Keller, R.M. and Kedar-Cabelli, S.T.: Explanation-Based Generalization: A Unifying View, *Machine Learning*, pp.47–80 (1986).
[Muggleton 89] Muggleton, S.: Duce, An Oracle Based Approach to Constructive Induction, *IJCAI89*, pp.287–292 (1989).
[Muggleton 92] Muggleton, S. and Feng, C.: Efficient Induction of Logic Programs, S. Muggleton (ed.), *Inductive Logic Programming*, pp.281–298, Academic Press (1992).
[Muggleton 94] Muggleton, S.: Bayesian Inductive Logic Programming, *ML-94*, pp.371–380 (1994).
[Numao 95] Numao, M., Morita, S. and Karaki, K.: A Learning Mechanism for Logic Programs Using Dynamically Shared Substructures, *MI'95* (1995).
[Pagallo 90] Pagallo, G. and Haussler, D.: Boolean Feature Discovery in Empirical Learning, *Machine Learning*, Vol.5, pp.71–99 (1990).
[Pazzani 92] Pazzani, M. and Kibler, D.: The Utility of Knowledge in Inductive Learning, *Machine Learning*, Vol.9, pp.57–94 (1992).
[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, *Machine Learning*, Vol.5, pp.239–266 (1990).
[Shapiro 83] Shapiro, E.Y.: *Algorithmic Program Debugging*, MIT Press (1983).
[吉田 92a] 吉田健一, 元田 浩: 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] 吉田健一, 元田 浩: 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).
[吉田 95] 吉田健一, 元田 浩, Indurkhya, N.: 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, 人工知能学会誌, Vol.10, No.1, pp.61–71 (1995).
[Yoshida 96] Yoshida, K. and Motoda, H.: Automated User Modeling for Intelligent Interface, *International Journal of Human Computer Interaction*, Vol.8, No.3, pp.237–258 (1996).

〔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. 工学博士. 1984年日本原子力学会論文賞, 1990年電気学会論文賞, 1992年人工知能学会論文賞, 1991, 1996年人工知能学会全国大会優秀論文賞, 1995年度人工知能学会研究奨励賞受賞. 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, 基礎研究所を経て平成7年退社. 現在, 大阪大学産業科学研究所教授 (知能システム科学研究部門, 高次推論研究分野). 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, とくに機械学習, 知識獲得, 知識発見, 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. 日本認知科学会常任運営委員, Artificial Intelligence in Engineering (Elsevier Applied Science) 編集委員, International Journal of Human-Computer Studies (Academic Press) 編集委員, 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.