論 文 Technical Papers

逐次ペア拡張による分類規則学習とその応用

Classification Rule Learning by Stepwise Pair Expansion and Its Applications

鹿山 俊洋 *,† 堀内 匡 * 元田 浩 * 鷲尾 隆 *
Toshihiro Kayama Tadashi Horiuchi Hiroshi Motoda Takashi Washio

* 大阪大学 産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University, 8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan

19YY 年 MM 月 DD 日 受理

Keywords: classification rule learning, graph-based induction, command prediction problem, WWW browsing problem

Summary

A machine learning technique called Graph-Based Induction (GBI) efficiently extracts typical patterns from directed graph data by stepwise pair expansion (pairwise chunking). In this paper, we apply GBI to the classification problem by interpreting the root node as a class node and the links attached to it as the primary attributes. The node at the other end of each link is the value of the attribute, which has secondary attributes. Thus, each attribute can have its own attributes recursively, and the graph becomes a directed tree. In this case, the pairwise chunking must start at the root node and go backwards following the links. Though inducing classification rules by GBI has been already proposed, we change the evaluation criterion in selecting the pair to be chunked and also make an addition of pruning mechanism to GBI in order to avoid overfitting to the training data. We call such an extended version of GBI as CL-GBI. Through applying CL-GBI to the following two kinds of problems, we show the effectiveness of our approach. One is the command prediction problem where CL-GBI must predict what the user wants to do next, and the other is the WWW browsing problem where CL-GBI must extract chracteristic routes from the access log which means the browsing history of many users.

1. はじめに

人工知能の分野において，計算機に学習能力を持たせることを目的とする研究は機械学習と呼ばれ，人工知能における最も挑戦的な分野の一角を形成している．その中でも，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 [Quinlan86]，抽象的概念の獲得 [Fisher87]，データやトレースからのプログラム生成 [Shapiro83] など，種々の研究分野を含んでいる．

本研究では，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似的ペアの逐次拡張」を基本とした Graph-Based Induction（GBI 法）[吉田 97] に分類規則学習の手法 [Quinlan93] を導入し，有向グラフから分類規則学習を行う手法 CL-GBI を提案する．さらに CL-GBI の応用例としてコマンド予測問題と WWW 巡回問題への応用とその結果について述べる．

コマンド予測問題は計算機ユーザが過去に実施したオペレーションの中から規則性を抽出して，ユーザが次に何をするかを予測してユーザに示し，ユーザはその指示に従えば目的のタスクが実行できるようになるという知的インターフェイスの研究に関するものである．特にユーザのオペレーション情報としてコマンドの入力順序とそれに起因するファイル I/O 情報をグラフ構造に表現し，グラフ構造の中から分類規則を抽出

† 現在，(株）東芝

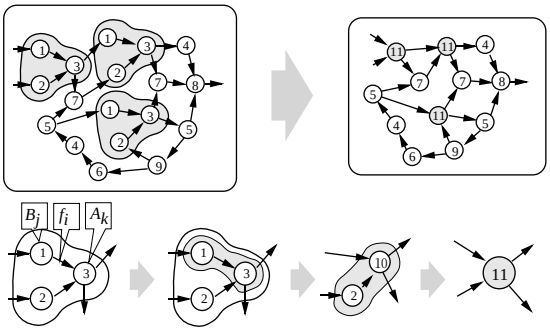

図 1　GBI 法の基本アイデア

する学習方法に関し検討したものである．

また WWW 巡回問題はサーバ側で収集したアクセス履歴ファイルからユーザの傾向を読みとる問題である．履歴ファイルから「利用者がある URL を通過すると，その利用者はある別の URL をすぐ後に通過することが多い」というパターンが得られれば，コンテンツ作成者にとって HTML 文章のレイアウト，URL 同士のリンク等を考える際に有用な情報となると考えられる．

以下，**2 章**では分類規則に適用された GBI 法とその改良について論じ，CL-GBI を提案する．この CL-GBI を用いて **3 章**ではコマンド予測問題への応用について，**4 章**では WWW 巡回問題への応用について述べ，最後に **5 章**で結論を示す．

2.　GBI 法とその改良

2・1　GBI 法の概要

GBI 法は，有向グラフ中に頻繁に表れるパターンを抽出することによって特徴的なパターンを発見することを目的として提案された（図 1参照）．[吉田 95] では，GBI 法は回路のシミュレーション結果から NOT, NOR 動作という特徴を抽出している．つまり，トランジスタ間の電圧・電流に関する因果関係（特にロジックを意図した値ではない）を有向グラフとして表現し，その有向グラフから GBI 法は抽象的な概念を導き出している．

元来，GBI 法は有向グラフ中の特徴的なパターンを一つのノードに置換すること（チャンキング）によって，与えられたグラフを小さくするアルゴリズムである．しかしこのとき，置換によってどれだけグラフが小さくなるかという評価だけでパターンを選択すると，置換が極端に進んだ結果グラフ全体が一つのノードに置き換わる可能性があるため，選択されたパターンの大きさも考慮している．またグラフ中のサブグラフを発見する（隣接した N 個のノードの置換についてそれぞれ評価する）というアルゴリズムは NP-完全問題であることが知られているため，$N = 2$ とする，つまり隣接した 2 ノード（置換されたノードも含む）のペアについてのみ評価するよう限定する．そして評価の高いペアをチャンクすることによって，GBI 法は

(1)　グラフに含まれる，2 つのノードの組からなる全てのペアを取り出す．
(2)　取りだされたペアの中から評価関数に応じてペアを一種類選択し，これを置換すべき（抽出）パターンとして登録する．
(3)　グラフの中に登録されたパターンと同じパターンがあれば，これをチャンクする．

という手順を繰り返してチャンクを逐次拡張し，与えられたグラフの特徴を抽出するという動作を実現している．

このように探索・チャンクの動作は逐次的なものであるため GBI 法は最適解を保証しないが，十分に納得のいく抽出を行うことが経験的に知られてている．ただそのためにはより複雑かつ多数のパラメータから導かれる評価基準を導入するほうが望ましい．そして終了条件にも繰り返し回数・抽出パターンのサイズなど様々なパラメータがある．

2・2　GBI 法の改良

前節で説明した GBI 法はグラフから類型パターンを抽出するためのアルゴリズムであるが，本節ではこれを分類規則学習に適用することに限定して考える．

GBI 法による分類規則の学習は [吉田 97] でも提案されているが，本研究では逐次ペア選択の際の評価基準を分類規則の学習により特化したものに変更するとともに，訓練データへの過剰適合を避けるために枝刈りの機能を追加する．このように本研究で構築した分類規則学習用 GBI を CL-GBI と呼ぶ．

〔1〕　評価基準の変更

2・1 節 の GBI 法の中で，チャンクするペアを選択する評価関数にペアの頻度等を用いると，特徴を抽出するという GBI 法本来の動作になる．本研究では，対象とするグラフとして木構造データのみを考え，木構造データの根ノードを結論部に，その他のリンク・子ノードを属性・属性値と対応付けることにより，GBI 法を分類規則学習に適用する．ここで，GBI 法を事例集合分類のための規則抽出に用いることを考えると，評価基準としては Information Gain [Quinlan86], Gain Ratio [Quinlan93], Gini Index [Breiman84] などが

考えられる．

すなわち，親ノード A_k のリンク f_i に子ノード B_j が繋がっているという状態（$A_k \overset{f_i}{\leftarrow} B_j$ と繋がっている状態）を三つ組 (A_k, f_i, B_j) として表現し，この三つ組みを ID3 [Quinlan86] における クラス・属性・属性値に対応させることで，ペアを選択するための評価関数に Gain Ratio の適用が可能になり，GBI 法の抽出したパターンを分類規則と見なすことが出来る．今回は，分類規則の学習により特化した評価基準として Gini Index ではなく Gain Ratio を用いた．

Information Gain および Gain Ratio の計算について簡単に説明する．まず，チャンクの遍歴が等しい事例の総和を N とし，そのうち親ノードが A_k である事例の数を n_k とする．このとき，分類前の情報量は

$$\mathrm{I(init)} = -\sum_k \frac{n_k}{N} \log_2 \frac{n_k}{N}$$

となる．ここで条件 (f_i, B_j) を設定し，これに適合する事例の総和を $N_{i,j}$，そのうち親ノードが A_k である事例数を $n_{k,i,j}$ とすると，条件に適合した事例集合の情報量は

$$\mathrm{I(i,j)} = -\sum_k \frac{n_{k,i,j}}{N_{i,j}} \log_2 \frac{n_{k,i,j}}{N_{i,j}}$$

となる．そして先の条件に適合しない事例の総和を $N_{\overline{i,j}}$，そのうち親ノードが A_k である事例数を $n_{k,\overline{i,j}}$ とすると，条件に適合しない事例集合の情報量は

$$\mathrm{I(\overline{i,j})} = -\sum_k \frac{n_{k,\overline{i,j}}}{N_{\overline{i,j}}} \log_2 \frac{n_{k,\overline{i,j}}}{N_{\overline{i,j}}}$$

となる．よって Information Gain は下の式で計算することが出来る．

$$\text{Information Gain} = \mathrm{I(init)} - \left(\frac{N_{i,j}}{N}\mathrm{I(i,j)} + \frac{N_{\overline{i,j}}}{N}\mathrm{I(\overline{i,j})} \right)$$

さらに，分類すること自体によって生じる偏りを表す分割情報量 Split Info は

$$\text{Split Info} = -\left(\frac{N_{i,j}}{N} \log_2 \frac{N_{i,j}}{N} + \frac{N_{\overline{i,j}}}{N} \log_2 \frac{N_{\overline{i,j}}}{N} \right)$$

と定められ，Information Gain を Split Info で割ることにより Gain Ratio を求めることが出来る．但し Split Info が 0 ないしそれに近くなった場合の Gain Ratio の爆発を考慮し，Information Gain がある程度以上高い場合にのみ Split Info と Gain Ratio を計算させる．

このような基準で例えば $(f_{i'}, B_{j'})$ の条件が選択された場合， 親ノード A_k に関わらず条件を満たすペア全てをチャンクすることによって，GBI 法はルール $(f_{i'}, B_{j'})$ を導くことができ，これを繰り返すことによって最終的に事例集合の分類規則学習が可能となる．

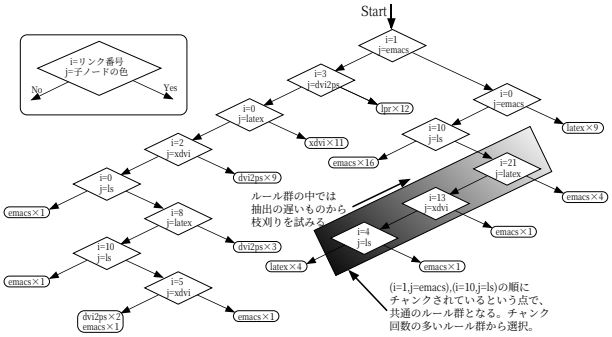

図 2　二分木と枝刈り

〔2〕 枝刈り機能の追加

ID3 では，分類規則として決定木を作成する．それに対して分類規則学習用 GBI は抽出パターンの集合を分類規則として持つ．但し，チャンクされなかったという情報を抽出パターン自体が持っていないため，分類規則の精度評価を行うためには抽出パターンの順序を厳密に規定し，抽出パターンを用いてテストデータを逐次チャンクする必要がある．これは抽出パターン（分類規則）を図 2のような二分木として構成することとほぼ同義である．

また GBI 法では分類規則抽出の際に逐次的にチャンクを行なっているため，照合の際にも同様にテストデータを逐次チャンクしなければならない．このため，サイズの小さな抽出パターンから先にテストデータと照合しチャンクを試みる必要がある．この規定はパターンの抽出された順序より優先され，プログラム上ではまず抽出順でソートし，次にサイズ順でソートされている．

前述の順序で分類規則とテストデータを照合するが，照合の段階では親ノードは比較せず，最後にとった照合の親ノード同士を比較する．こうして抽出データとテストデータの親ノードが同一であれば予測に成功したとみなせる．

ID3 を発展させた C4.5 [Quinlan93] では一旦作成した決定木から訓練データ集合に過剰適合 (overfitting) したと思われる部分木を切り，葉に置き換えることを行っている．このように決定木上で枝を刈る動作を枝刈り (pruning) という．

C4.5 では，決定木作成後に各部分木とそれを葉に統

合した場合の予測分類誤り率を計算し，統合した方が予測分類誤り率が低い場合に部分木を葉に置き換える（枝を刈る）．そして予測分類誤り率は二項分布に対する信頼限界 $U_{CF}(E, N)$ に分類された事例の数 N を掛けたものが用いられる．

本研究では，抽出パターン（分類規則）が図 2のような二分木として表現されるため，C4.5 と同様の手法を用いて枝刈りを導入することが可能となる．今回は作成された二分木からチャンク遍歴が最後の 1つを除いて同一のルール群（図 2 中の影部分）をチャンク数の多い順に抜き出し，ルール群の中では抽出が最も遅かったものが二分木の葉に来るため，そこから枝刈りの判断を順次行うようにした．

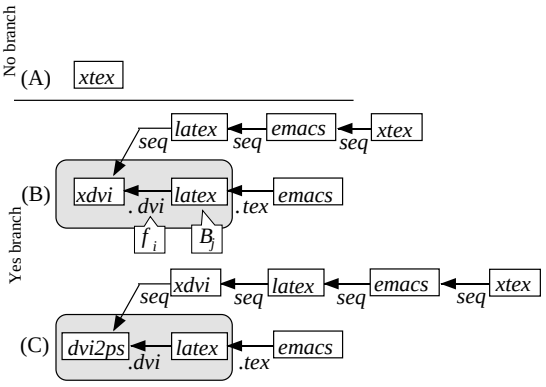

図 3 計算機利用履歴のグラフ表現

3. コマンド予測問題への応用

3・1 コマンド予測問題と CL-GBI

〔1〕 コマンド予測問題とデータ表現形式

前章の **2•2 節** で説明した CL-GBI 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，グラフ形式を用いている．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 [Breiman84, Quinlan86] と Inductive Logic Programming(ILP [Shapiro83]) が用いている述語論理の中間に位置する．従って，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ーフェイ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,「チャンクの逐次拡張」という考え方と問題領域にあ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基にした計算機インターフェイス・システム ClipBoard の詳細については [Yoshida96] を参照されたい．

〔2〕 ユーザーの操作とコマンド予測問題

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，このとき操作に用いるコマンドの選択はユーザー毎に異なり，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．

こうして受け取った利用遍歴データは木構造データの集合であるため，コマンド予測問題は「いかにして事例データの根ノードを精度良く予測する分類規則パターンを発見するか」という問題に定式化でき，CL-GBI を適用することができる．

図 3に示した例は，拡張子が dvi のファイルへの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ーフェイスの研究（[Greenberg88, Cypher91] 等）では，コマンド利用順序のように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コマンド間のファイルを介した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，コマンド間のファイル依存関係，ここでは拡張子が tex の文書ファイルの read/write の関係を示している．

以後，この「ファイルを介したコマンド間関係の情報」を便宜上「ファイル I/O 情報」と呼ぶ．

3・2 実験 (1):履歴情報と予測精度

前節ではユーザの操作を計算機に取り込む方法について述べたが，ユーザに依存するデータだけでなく，もっと具体的なデータに対しても分類規則学習用 GBI

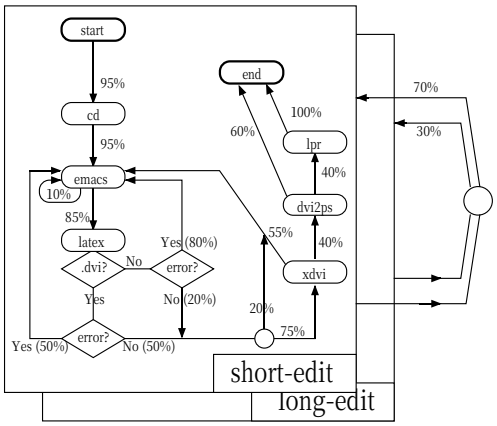

図 4 人工データ作成のための条件付き確率モデル

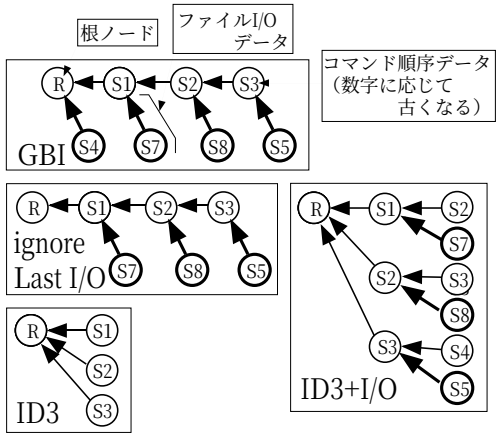

図 5 事例データの形式

(CL-GBI) の精度を評価することが重要である．そこで，まず人工的にデータ履歴情報を作成し，CL-GBI の予測精度を実験で検証した．また，履歴情報の表現形式として種々のデータ形式を挙げ，各々について予測精度を評価し，関連を見た．

〔1〕 人工データの生成

人工データは，ユーザーの操作を図 4 に示す条件付き確率モデルによって近似し，コマンド系列及びファイル I/O 情報を自動生成させることによって作成する．図 4中の short-edit, long-edit はそれぞれ簡単な報告書と図・参考文献入りのレポートを latex 等複数のアプリケーションを使用して作成するタスクをシミュレーションしたものである．このモデルでは，ユーザーはそれぞれ 30% ,70% の確率で short-edit タスク，long-edit タスクに移行し，各タスクの start から始まる．そして，矢印の脇に書かれた確率に従って状態を遷移 させ，そこに書いてあるコマンドを逐次的に実行したとみなす．ただし，例えば dvi ファイルの有無によって latex ドキュメントプロセッサの成否の確率が変動するなど不定長の過去に実行されたコマンドが次のコマンドに影響を与えることがある．

また，ここで実際ユーザーがよくするように，操作の流れに無関係なコマンド（例えば，ls, ps, mv など）をノイズとして挿入する．このノイズの挿入は再帰的に行われ，例えばノイズの確率を n とするとき n^2 の確率でノイズが 2 度入る．各コマンドが読む／書くファイルの設定はモデル中の全コマンドについて予め定められており，生成されたコマンド系列はこの設定に従ってファイルを読み書しているとみなされ，ファイル I/O の依存情報をデータに取り込むことができる．

〔2〕 事例データの形式および履歴情報の深さ

このようにして木構造の事例データを作成するが，事例データの形式を変化させることによって，同じ CL-GBI で処理するにしても異なった動作をさせることができる．

今回は，下の 4 つのデータ形式（図 5 参照）を採用し，CL-GBI により結果を比較した．

CL-GBI	コマンド順序データを直列に，ファイル I/O データを並列に接続したもの．
ID3	コマンド順序データのみを並列に接続したもの．従来のコマンド系列のみを考えた予測手法の例として取り上げた．
Ignore last I/O	CL-GBI とほぼ同じだが，各事例データの根ノードからのファイル I/O 情報を消去したもの．
ID3 + I/O	コマンド順序データのみを根ノードに並列接続し，各コマンド順序データについて CL-GBI と同様にコマンド順序・ファイル I/O データを付けたもの．

また，過去幾つのコマンドまで事例データに取り込むか（履歴情報の深さ）についても変化させ，予測精度の推移を見た．これは，例えば 5 つあるいは 6 つ前のコマンドのように予測に寄与する度合いが低いと思われるものでも分類規則として抽出されるという，いわゆる過剰適合（overfitting）について精度の低下を観察したものである．ただし，ファイル I/O 情報はそのコマンドがいつ実行されたかについては問わない．つまり，無限の過去に実行されたコマンドをファイル

表 1　人工データに対する予測精度

noise=10%

	CL-GBI	ID3	Ignore	ID3+I/O
depth=1	73%	54%	54%	55%
2	78%	61%	68%	69%
3	75%	61%	66%	66%
4	73%	54%	65%	66%

noise=20%

	CL-GBI	ID3	Ignore	ID3+I/O
depth=1	64%	47%	47%	47%
2	66%	52%	49%	49%
3	59%	51%	52%	51%
4	62%	52%	52%	51%

I/O 情報として持っていることも原理的にはあり得る．

〔3〕 実験結果と考察

100 個の事例データ集合について 10-fold cross-validation を用いて分類の精度を見た．また先にも述べたように，各事例データの形式や，過去いくつのコマンドまでグラフに取り入れるか（履歴情報の深さ：depth）を変化させた．その際，ノイズを挿入する確率 10%と 20%について実験し，その結果を表 1 に示す．

その結果，まず ID3 形式データと CL-GBI 形式データの比較により，ファイル I/O 情報を用いた CL-GBI の有効性を確認した．さらに ID3 形式データと ID3+I/O 形式データの比較により，根ノードに直接関与しないファイル I/O 情報もわずかながら分類精度に寄与することを示した．また，コマンド予測に必要十分な情報が 2 つないし 3 つ前のコマンドとそのファイル I/O 情報であることを示し，最後にノイズを増加させたときの CL-GBI 形式データと他形式との分類精度の比較により CL-GBI の頑健性を示した．

過去のコマンド予測問題では予測精度はほぼ 20%〜35%という値が出ており [吉田 97, 久保 95]，そのため ID3 形式データとの比較から，今回用いた人工モデルは実在のモデルより単純である，または実在のモデルはもっとノイズが多いとの示唆が読みとれる．

また，CL-GBI 形式の従来手法に対する優位性はノイズが増えても変わらないため，結果として CL-GBI の優位性は変化しない，もしくは強くなると予想される．

3・3　実験 (2):枝刈りと予測精度

本研究では **2•2 節**の〔2〕に示す枝刈りのアルゴリズムを導入したため，本節では CL-GBI の予測精度に対する枝刈りの効果をみるために実験を行なう．この実験では人工データだけでなく実際にユーザが操作した履歴もデータとして扱っている．

なお，人工データの作成（**3•2 節**〔1〕）に関しては先の実験と同じ方法であるため，ここでは省略する．また今回は事例データの形式（**3•2 節**〔2〕）は CL-GBI 形式と ID3 形式のみ用い，分類精度の評価には 4-fold cross-validation を用いた．

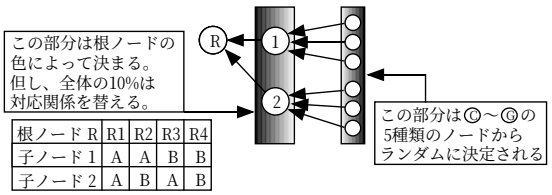

図 6　枝刈り検査用事例データ集合

表 2　枝刈りに関する予備実験の結果

CF [%]	精度 [%] train	精度 [%] test	ルールサイズ
0.1	92.7	89.2	5.25
0.5	94.0	86.7	10.00
1	94.6	82.5	13.25
100	97.1	83.3	37.00

註）精度 (train/test) = (訓練/テスト) データの予測精度

〔1〕 実データの収集

ユーザの計算機利用履歴データは，Linux OS のカーネルにユーザ情報取得用パッチをあてた計算機を実際に操作することによって収集することができる．このときデータは（実行属性＋シェル＋コマンド）または（読み／書き属性＋シェル＋コマンド）が一式になった形で収集されるため，AWK 言語で Idle 命令などの余分なデータを処理した後，Lisp 言語などでファイルを読み書きしたプロセスを関連付けるように変換することで前述の人工データと同様の処理が可能となる．ここでは，ユーザが頻繁に行う作業として latex 等を用いて文章を作成・整形する処理と，C コンパイラ等を用いて簡単なプログラムを作成・実行する処理の 2 つを行い，実際のユーザの利用履歴データを収集した．

今回収集したデータは 126 個で，データ形式及び履歴の深さは人工データと同じである．

〔2〕 枝刈りに関する予備実験

枝刈りの効果を見るために，図 6 で表される形式の事例データを 120 個作成し，分類精度・抽出ルールサイズの合計を調べた．

完全に枝刈りが行われたと仮定すると，訓練データおよびテストデータに対する予測精度はどちらも 90%となり，全ルールのチャンク数の和（以降ルールサイズと表記）は 4 となる．

〔3〕 実験結果と考察

i.　枝刈りに関する予備実験

まず，予備実験の結果について述べる．表 2 により，枝刈りの効果である余剰ルールの減少および予測精度の向上を確認できた．また，CF 値が小さいほど枝刈

りがされやすく，ルールサイズおよび訓練データに対する分類精度が減少する傾向も確認できた（なお 表2 中における CF=100% は，枝刈りを行わないことと同義である）．

しかし，C4.5 で一般的といわれている CF 値（C4.5 のデフォルト値は 25%）では分類精度がかえって悪化しており，CF 値を相当低くしなければ分類精度の向上がみられないことも見て取れる．これは決定木が二分木であることに起因すると思われる．例えば属性ないしリンクが L 種類存在し，各属性に M 種類の属性値または子ノードがある事例集合を仮定する．この事例集合から決定木を作成する場合，C4.5 では L 種類のうちから Gain Ratio を最大とする属性を選択するが，CL-GBI は $L \times M$ 種類の中から Gain Ratio を最大とする（子ノード・リンク）のペアを選択する．つまり C4.5 より CL-GBI の方が選択の幅が広く，そのためリンクと子ノードの両方を見ることによって分類しにくい（属性・属性値と結果の相関が薄い）ルールの抽出を避けることが容易であり，結果としてそのようなルールは決定木の葉まで到達するルールとなることが多い．

また，分類が完全に行われないにも関わらず分類に寄与するペアが発見できない場合，CL-GBI はその時点での各分類規則に即した根ノードの分布で多数決を取り，最も数の多い根ノードを分類規則として加える．ここで加えられた分類規則は決定木の葉[*1]に相当し，よって最初に枝刈りが行われる規則である．このように多数決を取る方法は，結果と相関のない（または極めて薄い）属性を持つ事例を処理する際にしばしば有効である．例えば，クラスと属性値に相関のない事例について，多数決を取るよりも分類を行った方が分類精度が悪化することが知られている [Quinlan93]．そのため，中途半端な枝刈りはかえって分類精度を低下させるという一見奇妙な事象が起こり得ると考えられる．

ii. コマンド予測に関する実験

人工データに関しては，10%の割合でノイズが混入した 100 個の事例データ集合を用い，分類精度とルールサイズを評価した．また実データに関しては，収集した 126 個の事例データ集合を用い，同様に評価した．この際，ファイル I/O 関係が確認できないコマンドを根ノードとする事例データが 41 個あったが，そのうちコマンドの順序が同じであるものが 28 個あるためこれを差し引くと，ノイズ混入率は 10.3% と考えることができる．

表 3 予測精度とルールサイズの比較 (人工データ)

	CL-GBI			ID3		
CF	精度 [%]		ルール	精度 [%]		ルール
[%]	train	test	サイズ	train	test	サイズ
0.1	95.0	77	20.50	86.5	48	31.50
1	95.3	77	20.50	86.5	48	31.50
100	99.0	75	28.75	92.0	51	61.75

表 4 予測精度とルールサイズの比較 (実データ)

	CL-GBI			ID3		
CF	精度 [%]		ルール	精度 [%]		ルール
[%]	train	test	サイズ	train	test	サイズ
0.1	91.3	76.2	22.75	87.3	61.9	31.50
1	91.7	76.2	23.75	87.3	61.9	31.50
100	94.0	78.6	37.25	91.7	63.5	47.75

結果は 表 3および表 4の通りで，分類精度をほぼ保ちながら抽出されたルールの合計サイズを枝刈り前の 55 〜 65%程度まで減少させることができた．ルールの削減という点では満足のゆく結果が得られたが，分類精度の点では枝刈りの効果は期待していた程ではなかったというのが実状である．

この原因の一つとして，CL-GBI の作成するルールを決定木と見たとき，これが二分木であることが挙げられる．二分木は分割による無駄が生じにくく，先の予備実験でも示したように，枝を効率的に刈るためはかなり CF 値を低くしなければならないためである．また，決定木は縮小できたことから元々のデータがそのような特性であったことも一因と考えられる．

また，今回用いた実データでは，人工データに比べて CL-GBI 形式と ID3 形式の予測精度の差が小さくなった．これは，実データの収集時に行った作業が極めて単純な処理であったことが原因として考えられる．

4. WWW 巡回問題への応用

4・1 WWW 巡回問題の特徴

インターネットのサービスの 1 つである WWW では URL から別の URL へとリンクが張られ，WWW 利用者はブラウザでリンクをたどり，次々に URL へアクセスしている．一方，URL を提供する WWW サーバにはクライアントのアクセスに関する情報が貯えらてれおり，そのデータベースのサイズは極めて膨大なものである．そこからユーザの特徴的な経路が得られれば，WWW 管理者やコンテンツの提供者がコンテンツの配置や HTML 文章のレイアウトを考える際に有用であると考えられる．

過去にユーザの参照した URL の履歴からユーザの動向を抽出した研究としては，グラフ構造データからの

*1 正確には，ルール群の最後

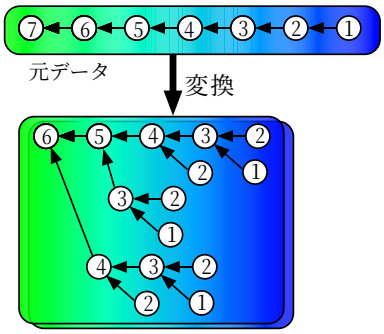

図 7　履歴情報の事例集合への変換

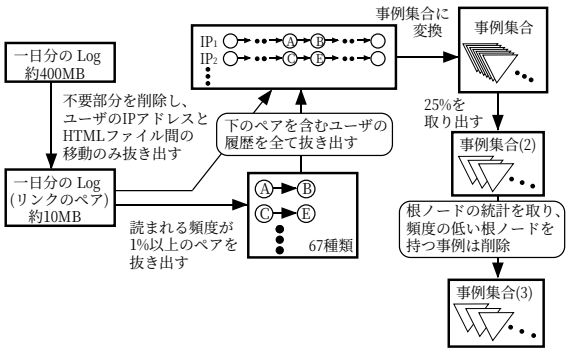

図 8　アクセスログのフィルタリング

バスケット分析 [猪口 98] などが挙げられる．バスケット分析は大規模データからの相関ルールの抽出に適したアルゴリズムであり，基本的に支持度 (support) と確信度 (confidence) の 2 つの閾値を設け，これらの閾値を越えた相関ルールを抽出する手法である．しかしバスケット分析によるルール抽出は分類規則学習という見地からの抽出ではなく，またグラフ構造を扱うことは出来るが，抽出された相関ルールの条件部と結論部の時間的な前後関係が明らかではなく，巡回経路の特徴を正しく反映しない場合がある．そこで，時間的な前後関係が忠実に保存される CL-GBI により従来のグラフ構造データからのバスケット分析とは異なった経路を抽出することを目的に実験を行った．

4・2 実 験 方 法

本研究では（株）リクルートの商用 WWW サイト「あちゃら」WWW サーバへのアクセス履歴の傾向分析を行った．アクセスログは，ユーザの IP アドレス・アクセス時刻・URL を一行に並べたテキスト形式のファイルである．また，このアクセスログはある 1 日分のデータで，ファイルサイズは約 400MB，この日「あちゃら」にアクセスした人は約 19,000 人であった．また，WWW サイト「あちゃら」において分析対象とした部分には，約 8,300 の URL とその間に張られた約 4 万のリンクが存在する．

このアクセスログの各行は URL 単位で記述されており，ユーザに個別の情報である IP アドレスに着目するとアクセス時刻によってユーザの WWW 利用履歴を取得することができる．そうすれば **3 章**のコマンド予測問題と同様にユーザの履歴をある程度の深さの事例集合に分割し，CL-GBI に分類規則を抽出させることが可能である．

各事例の形式は図 7に示す形に変換される．今回は幅 2, 深さ 5 と定め，ユーザが一つ前に辿った URL と二つ前に辿った URL を並列に接続させている（図 7の左下）．このようなデータ形式の表現により時間的な前後関係が正確な経路を分類規則として抽出することが可能であり，さらにユーザの履歴が一つおきに何らかの特徴を示す場合にもこれを抽出することが可能である．幅を 3 と定めれば二つおきに特徴を示す履歴にも対処できるが，今回は計算時間の問題から見送った．

しかし，今回はデータの規模が **3 章**と大きく異なることが問題となる．例えば事例データ数は **3 章**の 120 個に比べて約 13 万個，各ノードの色の種類数も数十に比べて 8,000 以上と桁違いの大きさであり，本研究で作成した CL-GBI ではアルゴリズムの複雑さ・記述言語（Lisp）の能力から有効な時間内に解析することが不可能である[*2]．そのため，数種類のフィルタリングを事前に試み，データの圧縮を行った．

〔1〕 フィルタリング (1)

図 8の左半分に示すように，1 日分のアクセスログから支持度が 1%を越えるペアを取り出し，このペアを含むユーザの全履歴を取り出した．支持度 1%を越えたペアは 67 種類であるが，このペアを通るユーザの殆どは 67 種類以外にも多数の URL を参照しており，結局 幅 2, 深さ 5 の事例集合に変換した時点で合計事例数は約 44,000，分類すべき根ノードは 11,000 種類となった．

〔2〕 フィルタリング (2)

前節で述べたフィルタリングの後でも現状の CL-GBI による抽出は時間的に不可能である．そのため〔1〕節で作成した事例データからランダムに 1/4 の事例を取り出し，（この時点で事例数 11,000，根ノード種類数

*2　今回は Lisp のインタプリタに GCL を用いたが，非常にメモリを消費するため実メモリ以外に仮想記憶を使用せざるを得ず，そのため時間的にも相当不利になってしまう．データ型を厳しく定義する C 言語などはその点で有利であると思われる．

2000），ノードの種類の分布をとった．そして同種の根ノードが少ない（5 以下）事例を削除した結果，事例数は約 8,000，色の種類数は約 300 となり（図 8の右半分参照），これにより CL-GBI による抽出が時間的に可能となった．今回の圧縮は頻度を基準としたフィルタリングのため，元データの特徴もある程度残っており，圧縮方法として妥当と考えられる．

4・3 実験結果と考察

CL-GBI による分析の結果として抽出された経路のうち，網羅する事例の数が 10 以上の経路は合計 12 種類であった．このうち代表的なものを 3 つ以下に示す．

1）/XYZ/s07.html（ビジネス）
　→ /XYZ/s07/ss04.html（企業情報）
2）/XYZ/s03/ss13.html（スポーツ）
　→ /XYZ/s03/ss13/d11.html（野球）
3）/XYZ/s09.html（飲食）→ 任意の URL
　→ /XYZ/s06.html（ニュース）
　　→ /XYZ/s07.html（ビジネス）
　　　→ /XYZ/s08.html（マルチメディア）

このように時間的な前後関係を正確に反映した経路が抽出されているが，3) の例のように任意の URL を経由する経路も抽出されているのが特徴である．また，このうち 2) はバスケット分析によって抽出されたルールでもあり，このような共通の経路も幾つか得られた．逆に CL-GBI では抽出されずバスケット分析によってのみ抽出されたルールの 1 つを以下に示す．

<支持度=1.4%，確信度=40.2%>
条件：/XY/category2.html（サーチエンジン）
　→/XYZ/Sub/s03.html（エンターテイメント）
結論：/XYZ/s03.html（エンターテイメント）
　→/XYZ/s03/ss13.html（スポーツ）

このように，バスケット分析と CL-GBI による解析の結果は，いくつかの共通部分とそれぞれの独自な部分を持っていることがわかった．

まず，CL-GBI で抽出した経路の特徴として時間的な前後関係が正確に反映され，なおかつユーザの履歴が一つおきに何らかの特徴を示す場合にもこの特徴を抽出していることがわかる．CL-GBI の抽出した経路に CGI 関係の履歴が多く含まれているのもそれが理由と考えられる．

またバスケット分析は巨大で空要素が非常に多いデータを高速に処理するという目的から生まれた手法であるため相関ルール抽出の際には支持度の重みが大きくなっているが，CL-GBI で分類規則抽出に用いた評価基準 Gain Ratio は「どれだけ結果を特定できたか」という基準であり，バスケット分析における支持度よりもむしろ確信度に近い評価基準であることも興味深い．CL-GBI によって抽出された経路の中に同じ URL が繰り返し出ているという現象がバスケット分析によるものより少ないのは，この評価基準が原因ではないかと考えられる．

5. おわりに

本研究では，グラフ構造から類型パターンを抽出するアルゴリズム GBI 法を分類規則学習に適用し，CL-GBI というシステムを構築した．GBI 法による分類規則の学習は [吉田 97] でも提案されているが，本研究では逐次ペア選択の際の評価基準を変更するとともに枝刈りの機能を追加した．この CL-GBI をユーザが次に入力するコマンドを予測するコマンド予測問題とユーザの辿る URL の履歴（アクセスログ）から特徴的な経路を抽出する WWW 巡回問題に応用し，その有効性を確認した．

今後の課題としては，コマンド予測問題ではまず作成した人工データや収集した実データが単純であり，そのためコマンド系列のみを用いた従来手法でもある程度の予測精度を得ることが出来た点にある．また，枝刈りの予測精度に対する効果が薄いということが示された．これには決定木が二分木であることが大きく関わると考えられる．

WWW 巡回問題においては，グラフの形式が一つ前と二つ前の URL を並列させただけの単純な形であったことが課題として挙げられる．一つ前の URL とそのカテゴリ・参照された時間帯を並列させるなどグラフの特性を生かしたデータ構成について検討が必要であり，これによってまた異なったパターンが抽出されることも期待できる．

本研究では，対象とするグラフを木構造データに限定し，GBI 法を分類規則学習に適用したが，今後は木構造データだけでなく自己ループを含む有向グラフや多入力多出力の有向グラフなどのより一般のグラフを扱えるように GBI 法を拡張することも極めて重要であり，現在検討しているところである．

謝　辞

GBI 法のプログラムの原型，および Linux OS 用ユーザ情報取得パッチを提供していただくとともに，熱心に議論していただいた日立製作所（株）吉田 健一氏に深く感謝します．また，商用 WWW サイト「あちゃら」の WWW サーバのアクセスログを提供していた

だいた（株）リクルートの熊澤 公平氏，荒井 尚英氏に深く感謝します．

◇ 参 考 文 献 ◇

[Breiman84] Breiman L., Friedman J. H., Olshen R. A., and Stone C. J.: *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[Cypher91] Cypher A.: Eager: Programming Repetitive Tasks by Example. In *CHI'91*, pp. 33–39, 1991.

[Fisher87] Fisher D. H.: Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning*, pp. 139–172, 1987.

[Greenberg88] Greenberg S. and Witten I. H.: How Users Repeat their Actions on Computers: Principles for Design of History Mechanisms. In *CHI'88*, pp. 171–178, 1988.

[猪口 98] 猪口, 鷲尾, 元田, 熊沢, 荒井：バスケット分析のグラフ構造データへの拡張と通信ネットワークデータへの適用，人工知能学会第 *33* 回人工知能基礎論研究会 *(SIG-FAI-9801)*, pp. 55–60, 1998.

[久保 95] 久保, 山本，守田，田中：システムの状態と依存関係に基づくコマンド予測，情報処理学会研究報告ヒューマンインターフェース， 62-11, pp. 75–82, 1995.

[Mitchell86] Mitchell T. M., Keller R. M., and Kedar-Cabelli S. T.: Explanation-based Generalization : A Unifying View. *Machine Learning*, pp. 47–80, 1986.

[Quinlan86] Quinlan J. R.: Induction of Decision Trees. *Machine Learning*, Vol. 1, pp. 81–106, 1986.

[Quinlan93] Quinlan J. R.: *C4.5: Programs For Machine Learning*. Morgan Kaufmann Publishers, 1993.

[Shapiro83] Shapiro E. Y.: *Algorithmic Program Debugging*. MIT Press, 1983.

[Yoshida96] 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.

[吉田 95] 吉田, 元田, and Indurkhya: 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.

[吉田 97] 吉田, 元田：逐次ペア拡張に基づく帰納推論，人工知能学会誌， Vol. 12, No. 1, pp. 58–67, 1997.

〔担当編集委員：××○○，査読者：○○××〕

著 者 紹 介

鹿山 俊洋(学生会員)
〈kayama@ar.sanken.osaka-u.ac.jp〉

堀内 匡(正会員)
〈horiuchi@sanken.osaka-u.ac.jp〉

元田 浩(正会員)
〈motoda@sanken.osaka-u.ac.jp〉

鷲尾 隆(正会員)
〈washio@sanken.osaka-u.ac.jp〉