論 文

Technical Papers

一般グラフ構造データに対する Graph-Based Induction とその応用

Graph-Based Induction for General Graph Structured Data and Its Applications

松田 喬 Takashi Matsuda
大阪大学産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University.
matsuda@ar.sanken.osaka-u.ac.jp

元田 浩 Hiroshi Motoda
（同　上）
motoda@sanken.osaka-u.ac.jp

鷲尾 隆 Takashi Washio
（同　上）
washio@sanken.osaka-u.ac.jp

keywords: Graph-Based Induction, general graph structured data, data mining, machine learning

Summary

A machine learning technique called Graph-Based Induction (*GBI*) efficiently extracts typical patterns from graph data by stepwise pair expansion (pairwise chunking). In this paper, we introduce Graph-Based Induction for general graph structured data, which can handle directed/undirected, colored/uncolored graphs with/without (self) loop and with colored/uncolored links. We show that its time complexity is almost linear with the size of graph. We, further, show that *GBI* can effectively be applied to the extraction of typical patterns from DNA sequence data and organnochlorine compound data from which to generate classification rules, and that *GBI* also works as a feature construction component for other machine learning tools.

1. はじめに

近年，データマイニングに関して多くの研究があるが，多くは通常のデータベースを念頭においたものであり，複雑なデータ構造を有するデータからの知識発見の有効な手法はいまだ存在しない．一般に知識や概念はグラフ構造で記述できるため，グラフ構造データからの知識発見は極めて重要な研究課題である．

構造を有しない通常のデータに対して，現在非常に広く使われている方法はデータを属性とその値のペアで表現し，属性の値と同定したいクラスの関係を決定木 [Quinlan 86] や規則 [Michalski 90, Breiman 89] で表現するものである．データマイニングでよく使われる相関規則 [Agrwal 94] もこの表現形式に入る．しかし，属性と値のペアの表現形式は一般的な構造データを表現するのには適しておらず，問題によってはより強力な表現形式が必要となる．帰納論理プログラミング [Muggleton 94] は一階述語論理表現を用いているため，一般的な関係を表現でき，かつデータ表現と帰納された知識表現がともにホーン節であるため，獲得した知識を背景知識に加え，そのまま使うことができるという利点がある．しかし，強力ではあるが，誰もが手軽に使えるほど技術的に成熟していない．Graph-Based Induction（GBI 法） [吉田 97] は，有向グラフ中に頻繁に現れる特徴的なパターンを，隣接する二つのノードを逐次的にチャンクすることによって発見することを目的として考案された手法であり，知識表現に関しては両者の中間に位置する．

GBI 法は隣接する二つのノードを逐次的にチャンクしていくため（Greedy 探索を行うため）ノードのラベルに同じものが多数含まれるようなグラフ構造からのパターン抽出には向いていない．しかし，ノードのラベルがすべて異なるようなグラフ構造（Web のブラウジングデータなど）や，ノードのラベルに同じものが多数含まれていても，データ中に特徴的な構造が含まれているようなグラフ構造（ベンゼン環などを含む化学構造データなど）などのデータからのパターン抽出には有効であると考えられる．

本論文では，GBI 法の概要について説明し，ノード，リンクにラベルがあり，以前の実装[*1]では取り扱うこと

*1 以前の GBI 法の実装 [吉田 92, Fig. 12] ではノード，リンクにラベルがある木構造データしか入力データとして用いることができなかった．

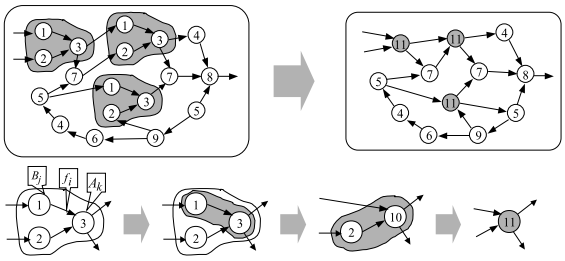

図 1 GBI 法の基本アイデア

のできなかった，各ノードに多入力・多出力，ノード間にループ（自己ループ）を許す一般グラフに適用できる知識表現の拡張について説明した後，この GBI 法の計算時間を理論的，実験的に評価する．さらに，GBI 法を DNA 塩基列の分類問題および有機塩素化合物からのパターン抽出に適用し，GBI 法により分類規則学習およびパターン抽出が可能となることを示す．そして，最後に GBI 法による分類規則学習のための属性構築について述べる．

2. GBI 法

2·1 GBI 法の概要

GBI 法は，有向グラフ中に頻繁に現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考案された [吉田 97]．類型的なパターンは直感的に概念をあらわしており，「類型的」は頻度もしくは頻度に基づく評価関数の値により特徴付けることができると考えられる．この仮定のもとで，GBI 法は 図 1 に示すように「ペアの逐次抽出により類型的なパターンを抽出する」というアイデアにより実現されている．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．

頻度もしくはそれに基づく評価関数としては，頻度，Information Gain [Quinlan 86]，Gain Ratio [Quinlan 93]，Gini Index [Breiman 84] などの統計的指標を用いることができる．逆に，これらの頻度に基づく評価関数で測ることのできない概念は取り扱わない．

図 2 を用いて，グラフ形式の入力データから，「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 2 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実施する．

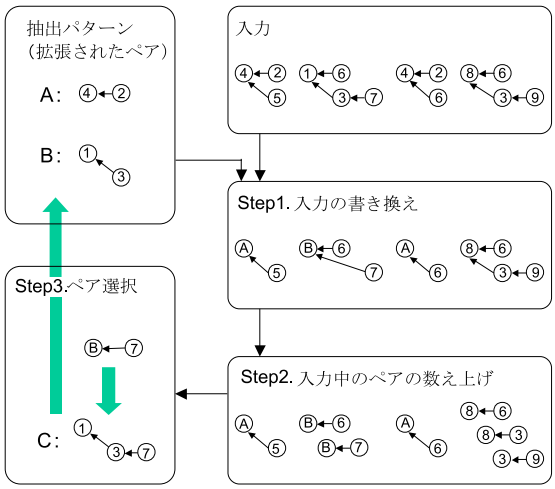

図 2 逐次ペア拡張の基本アイデア

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. 書き換え後のデータに含まれる，二つのノードの組み合わせからなるすべてのペアを抽出する．

Step 3. 抽出したペアのうち，最も「類型的」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．

類型的パターンが何も抽出されていない初期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．

ただし，**STEP 2.** でペアを抽出する際，一つのノードに書き換えられたパターンをもとのパターンに復元しての評価は行わない．つまり，GBI 法は逐次的な探索・チャンクを行う Greedy 探索手法であり，入力グラフデータ中に存在するすべての「類型的なパターン」を抽出できるわけではない．グラフからすべての部分グラフを抽出する問題は NP 困難問題であるため，GBI 法はすべての類型的パターンを抽出するのではなくある程度の大きさを持った有意の部分グラフを抽出することを目的としている．そのため，大規模なグラフ構造データからパターンを抽出するには，高速に動作するために非常に有用である．また，ペアの逐次拡張という Greedy 探索を採用しているため，すべてのノードのラベルが異なる場合はチャンクするペアの選択にあいまい性が生じずうまく動作するが，同じノードラベルが多数存在している場合には，評価値が同数のペアや同種類のペアの連鎖[*2]が生じるためにチャンクすべきペアの選択にあいまい性が生じる．

*2 $a \to a \to a$ といった構造の場合，どちらの $a \to a$ をチャンクすべきかを決定できない．

```
1)  G = {Input Graph-Structured Data};
2)  pair = count_pair( G );
3)  for( ; eval( pair )!=NULL ; ) do begin
4)        add_chunk( eval( pair ) );
5)        G = rewrite( G , eval( pair ) );
6)        pair = count_pair( G );
7)  end
```

図 3 GBI 法のアルゴリズム

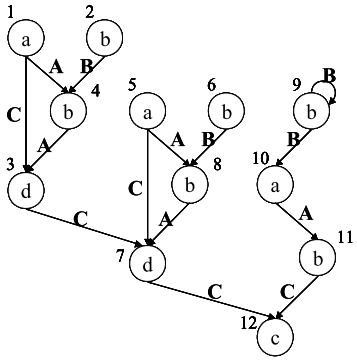

図 4 有向グラフの例

表 1 有向グラフを表へ変換した例

ノード番号	ノードのラベル	子ノードの番号 (リンクのラベル)	
1	a	3(C)	4(A)
2	b	4(B)	
3	d	7(C)	
4	b	3(A)	
5	a	7(C)	8(A)
6	b	8(B)	
7	d	12(C)	
8	b	7(A)	
9	b	9(B)	10(B)
10	a	11(A)	
11	b	12(C)	
12	c		

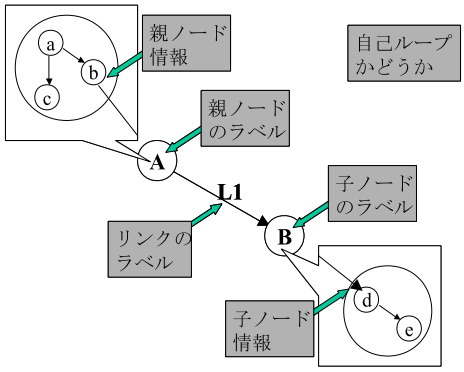

図 5 GBI 法におけるペアの概念

2·2 GBI 法のアルゴリズムと知識表現の拡張

GBI 法の基本的なアルゴリズムの流れは 図 3 のようになる．このアルゴリズムを C 言語を用いて実装した．

G には，入力グラフのグラフ構造が記憶される．具体的には GBI 法を多入力多出力の有向グラフ，自己ループを含む有向グラフなどの一般グラフへ適用するために，データを表形式を用いて表す．たとえば 図 4 の有向グラフは 表 1 のように表現される．この表の 1 行目はノード番号 1 のノードはノードのラベルが a であり，子ノードにノード番号 3（リンクのラベル C） と 4（リンクのラベル A） のノードを持つことを表しており，この表現方式を用いれば，有向グラフを表形式で扱うことができるようになる．そして，関数 count_pair(G) により，グラフ構造 G に存在するペアを全て数え上げ $pair$ に記憶する．

ペアは（親ノード $\overset{\text{リンクのラベル}}{\rightarrow}$ 子ノード）の組み合わせで数えるため，親ノードと子ノードが同じラベルのノードである場合（例えば $a \rightarrow a$ など）自己ループであるか，あるいは自己ループではないかを判別する必要がある．さらに，チャンクされたノードを元のパターンに復元するため，チャンク後もノード間のリンク情報が完全に維持できるようにした．これは,「親ノード情報」および「子ノード情報」を記憶することにより実現される.「親ノード情報」とは，ペアにおいて親ノードがチャンクされている場合リンクが親ノードチャンクの中のどのノードから出ているかを記憶するためのもので，同様に,「子ノード情報」とは，子ノードがチャンクされている場合リンクが子ノードチャンクのどのノードに入っているかを記憶するためのものである．つまり，GBI 法ではペアは 図 5 に示すように「親ノードのラベル」,「親ノード情報」,「子ノードのラベル」,「子ノード情報」,「リンクのラベル」,「自己ループであるかないか」という 6 つの要素で定義される．

したがって，次の条件を満たすときに限り $pair_1$ と $pair_2$ は同値のペアであると定義できる．

$$\begin{aligned}
&\text{parent}(pair_1) = \text{parent}(pair_2) \wedge \\
&\text{parent_info}(pair_1) = \text{parent_info}(pair_2) \wedge \\
&\text{child}(pair_1) = \text{child}(pair_2) \wedge \\
&\text{child_info}(pair_1) = \text{child_info}(pair_2) \wedge \\
&\text{link_label}(pair_1) = \text{link_label}(pair_2) \wedge \\
&\text{self_dist}(pair_1) = \text{self_dist}(pair_2)
\end{aligned}$$

ここで，$pair_i$ を G 中に存在するある一つのペアとしたとき，関数 parent($pair_i$) , child($pair_i$) は $pair_i$ の親ノードあるいは子ノードのラベルを，関数 parent_info($pair_i$) , child_info($pair_i$) は $pair_i$ の「親ノード情報」あるいは「子ノード情報」を，関数 link_label($pair_i$) はリンクのラベルを，関数 self_dist($pair_i$) は $pair_i$ が自己ループかどうかの値をそれぞれ返す関数である．

表 2 図 4 についての *pair*

親ノードのラベル	親ノード情報	子ノードのラベル	子ノード情報	リンクのラベル	自己ループ判別フラグ	頻度
a	0	b	0	A	0	3
a	0	d	0	C	0	2
b	0	a	0	B	0	1
b	0	b	0	B	0	2
b	0	b	0	B	1	1
b	0	c	0	C	0	1
b	0	d	0	A	0	2
d	0	c	0	C	0	1
d	0	d	0	C	0	1

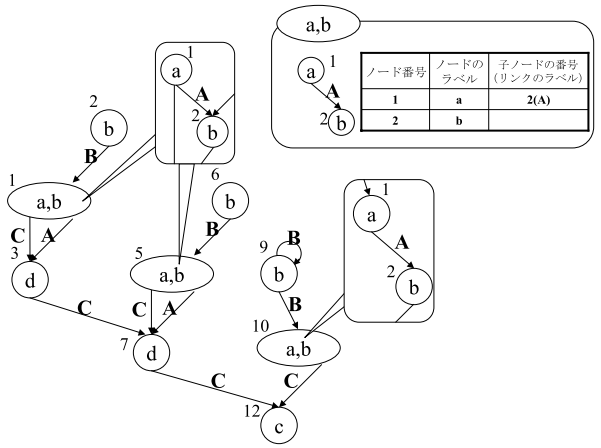

図 6 図 4 をチャンクした例

図 4 では，*pair* は 表 2 のように記憶されている．この表では，親ノード情報・子ノード情報の値が全て “0” になっているが，これは全てのノードがまだチャンクされていない状態のため，チャンクされていないノードを表す値 “0” が入っている．また，自己ループ判別フラグはペアが自己ループの場合には “1” が，自己ループでない場合には “0” が入る．

次に，*pair* の中から評価関数により，チャンクすべきペアを一つ選び出し（ eval(*pair*) ），そのペアを新しいチャンクとして記憶する（ add_chunk(eval(*pair*)) ）．この際，評価関数により選ばれたペアの親あるいは子ノードがすでにチャンクされているノードの場合は，もとのグラフ構造に戻して記憶する．さらに，グラフ構造 G を評価関数で選ばれたペア eval(*pair*) を一つのノードとすることによりグラフを新しく書き直す（ rewrite(G , eval(*pair*)) ）．図 6 は 図 4 の中で最も頻度が多い a $\stackrel{A}{\rightarrow}$ b のペアをチャンクした結果を示しており，このチャンクの情報は，図 6 右上図のように保持される．

図 6 における *pair* を 表 3 に示す．図 6 では，b $\stackrel{B}{\rightarrow}$ (a,b) のペアが 3 つある．しかし，これらのうち左側の 2 つのペア b $\stackrel{B}{\rightarrow}$ (a,b) はノード b から出ているリンクがノード (a,b) の中のノード b に入っているが，右側のペア b $\stackrel{B}{\rightarrow}$ (a,b) はノード b から出ているリンクがノード (a,b) の中のノード a に入っているため，実際はこれらのペアは異なるペアとして取り扱わなければならない．

表 3 図 6 についての *pair*

親ノードのラベル	親ノード情報	子ノードのラベル	子ノード情報	リンクのラベル	自己ループ判別フラグ	頻度
b	0	b	0	B	1	1
b	0	a,b	2	B	0	2
b	0	a,b	1	B	0	1
a,b	1	d	0	C	0	2
a,b	2	d	0	A	0	2
a,b	2	c	0	C	0	1
d	0	c	0	C	0	1
d	0	d	0	C	0	1

表 3 ではこのことは 2 行目と 3 行目に表れており，これらのペアは子ノード情報の値が違うため，異なるペアとして取り扱うことができる．

このステップを，評価関数によりチャンクすべきペアが選ばれなくなるまで繰り返すことで，*GBI* 法はグラフ構造から頻出する部分グラフ構造を高速に発見することを可能としている．

3. GBI 法の計算時間に関する評価

3·1 理 論 的 評 価

実装プログラムの計算時間について以下に考察する．i 番目のチャンクステップにおけるグラフ構造データ中の全ノード数を N_i，一つのノードあたりから出ているリンクの平均本数を l_i 本，ペアの種類数を P_i とし，グラフ中から抽出するチャンクパターンの種類数を C とする．

まず，前節で示した表形式で表された入力データをメモリ上に読み込む計算時間は，グラフ中のすべてのリンクデータを読み込むために $O(N_0 l_0)$ （入力グラフ中に存在する総リンク数は $N_0 l_0$ である）となる．次に，メモリ上に読み込まれたグラフ中に存在するすべてのリンクを探索し，ペアの種類ごとにカウントしていくために計算時間は $O(N_0 l_0 \log N_0 l_0)$ となる．

i 番目のチャンクステップにおいて，評価関数によりチャンクすべきペアを探し出すためには，ペア情報をすべて探索しなければならないため計算時間は $O(P_i)$ である．そして，評価関数に基づき選ばれたチャンクすべきペアをグラフ中から探し出し，チャンクしていく計算時間は，グラフ中に存在するすべてのリンクを探索するために，$O(N_i l_i)$ となる．さらに，チャンクされることにより，変化したペア情報を更新するためにかかる計算時間はすべての種類のペアを探索するために $O(P_{i+1})$ である．

GBI 法はチャンクすべきペアが見つからなくなるまで上記の動作をチャンクされるパターン数繰り返すため（C 回）総計算時間は，

$$O(N_0 l_0 + N_0 l_0 \log N_0 l_0 + \sum_{i=0}^{C-1} (P_i + N_i l_i + P_{i+1}))$$

となる．ここで，i 番目のステップにおいて，ノード数，リンク数は少なくとも入力データに存在したノード数，リ

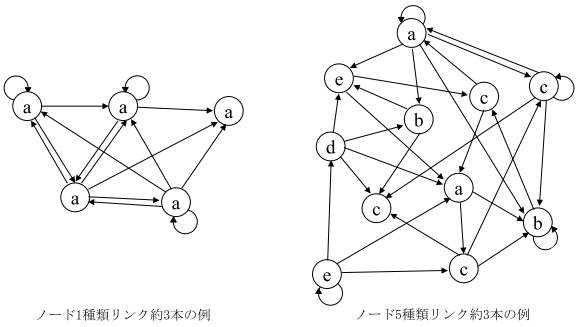

図 7 計算時間の評価に用いたデータ例

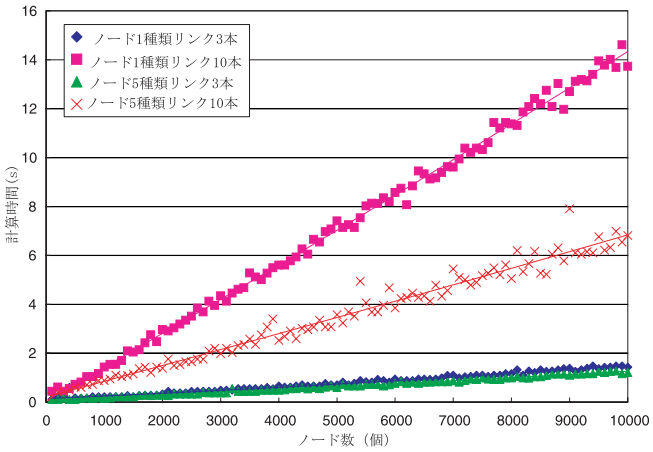

図 8 ノード数と計算時間の関係

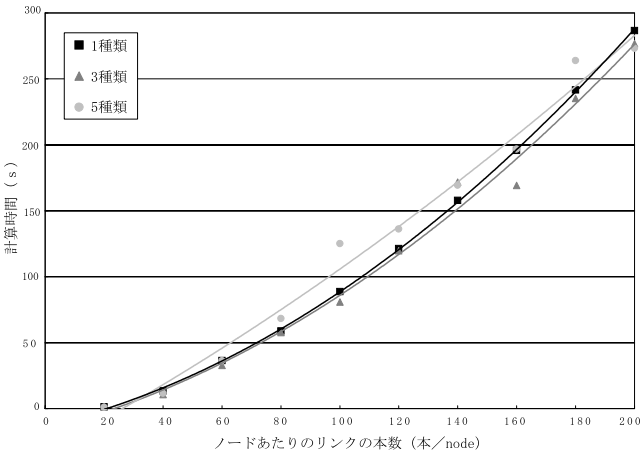

図 9 リンク数と計算時間の関係

ンク数よりも少なくなるために，$N_i < N_0$，$l_i < l_0$ であり，データ中に存在するペアの種類数は少なくともデータ中に存在するリンクの総数以下となるため，$P_i \leq N_i l_i < N_0 l_0$ である．つまり，GBI 法の計算時間は，

$$< O(N_0 l_0 \log N_0 l_0 + \sum_{i=0}^{C-1} N_0 l_0)$$
$$= O(N_0 l_0 \log N_0 l_0 + C N_0 l_0)$$

となる．ここで，チャンク回数 C は，最大でも入力データ中に存在するリンクの総数以下となるために，$C \leq N_0 l_0$ となり，GBI 法の最大時間計算量は，

$$O(N_0^2 l_0^2)$$

となる．

3·2 実験的評価

本節では，ランダムな構造をもつグラフデータを人工的に用意し，それらに対する開発プログラムの計算時間についての実験的評価を行う．なお，計算時間の評価に用いた計算機のスペックは CPU Pentium II 400MHz，メモリ 256 MB である．まず，ランダムな構造をもつグラフ構造データを人工的に用意し，計算時間をノードが 1 種類の場合と 5 種類の場合について評価した．具体的には，ノード数を 100 個から 10,000 個まで 100 個づつ増やしていき，ノード間のリンクを，一つのノードから出ているリンクが平均 3 本の場合と 10 本の場合になるように確率を設定した上でランダムに張り評価した（ 図 7 参照）．評価関数にはペアの頻度を用い，頻度の閾値には全ノード数の 4 % を用いた．つまり，最も頻度が多く，かつその頻度が閾値以上であるペアをチャンクするペアとして選択した．以上の設定で，プログラムの実行開始から終了までの時間を測定した．計算時間の結果を図 8 に示す．

図 8 より，計算時間はほぼノード数に比例していると言える．また，ノードの種類は少ない方が計算時間は多くかかっている．これは，ノード数が同じ場合，ノードの種類が少ない方が同じパターンが存在する確率が高くなり，チャンク回数が増える傾向があるためである．また，一つのノードあたり出ているリンクが多くなると，計算時間が増えている．これはリンクの数が多くなると同時に，ペアの数が多くなるためチャンクすべきペアを探索する空間が大きくなり，また，同じパターンが存在する確率が高くなるため，チャンク回数も増えるからである．

次に，グラフ構造データ中に存在するリンクの数と計算時間の関係を実験的に評価した．ノード数を 200 個に固定し，あるノードから他のノードにリンクを張る確率を 10 % から 100 % まで（100 % は完全グラフに対応）10 % 刻みで変化させることでランダムな構造をもつグラフ構造データを作成した．ノードの種類は 1 種類，3 種類，5 種類の場合について評価した．3·2 節 の実験と同様に，評価関数には頻度を用い，全ノード数の 4 % を閾値に用いた．その結果を 図 9 に示す．

図 9 より，計算時間はグラフ構造データ中に存在するリンク数の増加に対して 2 次関数で抑えられていることがわかる．しかし，現実のデータではノード数が増加しても，一つのノードあたりから出ているリンクの本数は

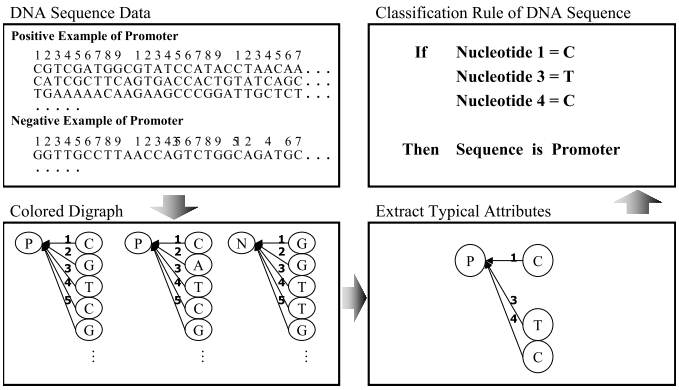

図 10 DNA 塩基列データからのパターン抽出

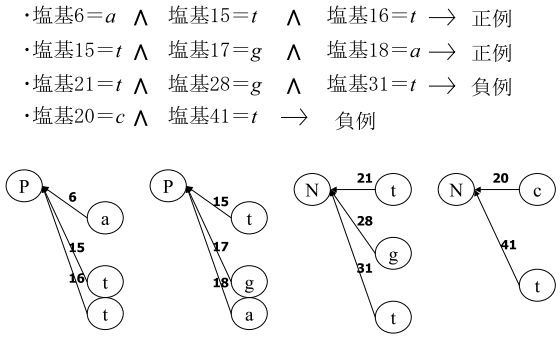

図 11 Promoter データセットからのパターン抽出例

表 4 予測精度の比較結果

Learning Method	ID3	C4.5	*GBI*
No. of Errors /106	19	18	16

ドメインごとにある程度決まった値に固定されると考えられるために現実データに適用する場合には GBI 法の計算時間は 図 8 に近い挙動を示すと考えられる．

4. GBI 法を用いた分類規則学習

本章では，DNA の塩基列データからの分類規則抽出に関する GBI 法の適用について述べる [Matsuda 00a]．

4･1 Promoter データセットを用いた分類規則学習

Promoter データセットは UCI Machine Learning Repository [Blake 98] により提供されているデータセットの一つである．Promoter データセットは塩基をあらわす A，G，T，C からなる長さ 57 の文字列データであり，クラスはその塩基列が “Promoter” *3を含むことを示す Positive と “Promoter” を含まないことを示す Negative の 2 つである．データセット中の事例数は 106 個で，クラス Positive のデータが 53 個，クラス Negative のデータが 53 個である．

図 10 は，このデータセットを GBI 法に適用できるようにグラフ構造データに変換する方法を示したものであり，一行の文字列データから一つのグラフを作り，クラスごとにまとめたものを入力データとして用いた．パターン抽出のための評価関数にはペアの頻度を用い，閾値は全事例の 4 % の値を用いた．計算機のスペックは CPU Pentium II 400MHz，メモリ 256 MB で，パターン抽出に要した時間は約 10 秒であった．

図 11 は，このデータセットからのパターン抽出を行った結果の一例である．

さらに，これらのパターンは根ノードを結論部，葉ノードと葉ノードへのリンクを条件部と見ることによりルールとして用いることができるため，これらの抽出したパターンを用いて leaving-one-out 法により分類規則学習の予測精度を測定した．パターン抽出の評価関数には頻度を用い，得られたパターンを評価関数の値が低かったものからパターン照合を行った．評価関数の値が同じ場合は，ノード数の多いものよりパターン照合を行った．これは，より具体的なパターンからパターン照合を行うためである．

表 4 は，代表的な分類規則学習法である ID3，C4.5 とエラーの数を比較した表である．ID3 と C4.5 は塩基の位置とその値をそれぞれ属性と属性値として用い，クラスにはその塩基列が “Promoter” であることを示す Positive と “Promoter” でないことを示す Negative を用いた．この表よりこのデータセットに対しては GBI 法は ID3，C4.5 より若干良い分類精度を持っているということが言える．

4･2 Splice データセットを用いた分類規則学習

Splice データセットは 4･1 節 と同様の DNA 塩基列データで，同じく UCI Machine Learning Repository [Blake 98] からのデータセットである．クラスは E/I*4，I/E，Neither であり，文字列の長さは 60 である．事例数は 3190 個で，クラス E/I のデータが 25 %，クラス I/E のデータが 25 %，クラス Neither のデータが 50 % からなる．

このデータを Promoter データセットと同様に 図 10 に従い，データをグラフ構造データに変換し，これを入力データとして用いた．パターン抽出のための評価関数にはペアの頻度を，閾値には，全事例の 1 % の値を用いた．計算機のスペックは CPU Pentium II 400MHz，メモリ 256 MB で，パターン抽出に要した時間は 10 分程度であった．

*3 DNA を鋳型に mRNA 合成を開始する DNA 上の特定の塩基列を指す．

*4 クラス “E/I” は，その DNA シークエンスが結合後の状態を維持している DNA の一部分を表す “exon” と結合中の DNA の一部分を表す “intron” の境界であることを示す．

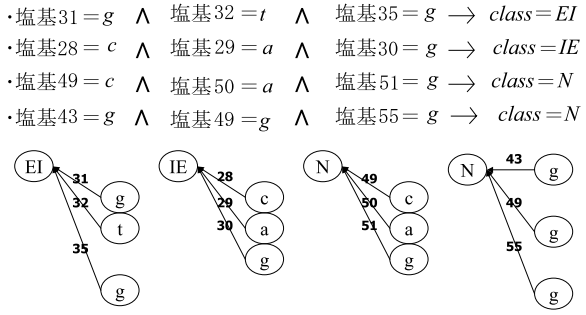

図 12 Splice データセットからのパターン抽出例

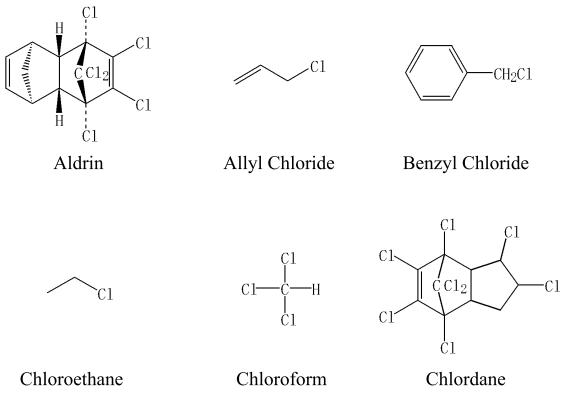

図 13 有機塩素化合物データの例

表 5 予測精度の比較結果

Learning Method	ID3	C4.5	*GBI*
Error Rate(%)	10.6	8.0	8.8

図 12 は，このデータセットからのパターン抽出を行った結果の一例である．

さらに，抽出されたパターンを用いて，10-fold cross-validation 法により分類規則学習の予測精度を測定した．パターンの照合方法は 4･1 節 と同様である．表 5 は代表的な分類規則学習法である ID3，C4.5 とエラー率を比較した表であり，この表より，このデータセットに対しては，ID3 より良い分類精度を持っているが，C4.5 には若干及ばないが同程度ということができる．

5. 化学化合物データからのパターン抽出

本章では，化学化合物データからのパターン抽出に関する GBI 法の適用について述べる [Matsuda 00b].

5･1 発ガン性化合物データへの適用

現在，新規化学物質が多数合成され，我々の生活および健康の向上に役立っているが，一方でそれらの化学物質の有害性が問題となっている．そのため，化学物質の有害性を評価する必要があるが，これらを実験的に測定を行うためには長期の年月と多額の費用を必要とし，全ての化学物質について有害性データを実験的に評価することは経済的，時間的に不可能である．つまり，化学物質の構造からその有害性の予測を試みることは，化学物質を合成する前にその有害性・安全性を事前に評価でき，新規化学物質の開発にとって意義はきわめて高い．

そこで，GBI 法を有機塩素化合物からの発ガン性予測に適用し，その結果について考察する．パターン抽出に用いたデータは，松本らがニューラルネットワークによる有機塩素の発ガン性予測 [松本 99] で用いるために NTP (National Toxicology Program) データベースから入手し，C，H，Cl のみからなる化合物を抽出したものである．このデータは，発ガン性を持つ化合物（正例） 31 個，発ガン性を持たない化合物（負例） 10 個の合計 41 個からなるデータである．今回用いた有機塩素化合物データの一例を 図 13 に示す.

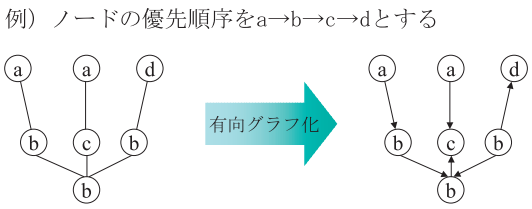

図 14 無向グラフを有向グラフに変換した例

まず，はじめに GBI 法を無向グラフに適用できるようにするためにノードの種類にある一定の順序を持たせることで， 図 14 のように無向グラフを有向グラフに変換する．この図は，ノードの種類に $a \to b \to c \to d$ という順序を持たせ，リンクの方向に一意性を持たせることで無向グラフを有向グラフに変換できることを示す例である．つまり，例えば a から b の間のリンクは，前出のノードの種類順に従い，$a \to b$ とリンクの方向を決めることができる．なお，a から a のように同じ種類のノード間のリンクの方向は任意に決めた．これにより， 図 14 左図の無向グラフは 図 14 右図の有向グラフに近似的に変換される．

前処理としてノードを炭素，塩素，ベンゼン環の 3 種類とし，水素は除いた．リンクのラベルには単結合，2 重結合，芳香族同士の結合を用いた．図 15 に有機塩素化合物のグラフ構造化の一例を示す.

なお，ここでもパターン抽出のための評価関数には頻度を用い，ペアの頻度が 2 以上であるものをチャンクした．GBI によるパターン抽出にかかった計算時間は約 1 秒（CPU:Pentium III , Memory: 384 MB）であった．正例から抽出されたパターンの一例を 図 17 に，負例から抽出されたパターンの一例を 図 18 に示す．さらに，正例から抽出されたパターンと負例から抽出されたパターンを比較することにより，どちらか一方のみに現れるパ

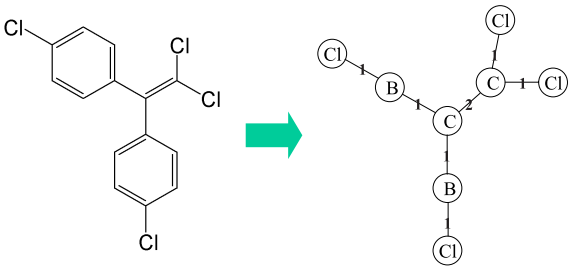

図 15 グラフ構造化の一例

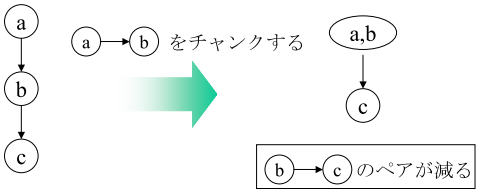

図 16 ペアの数え上げ

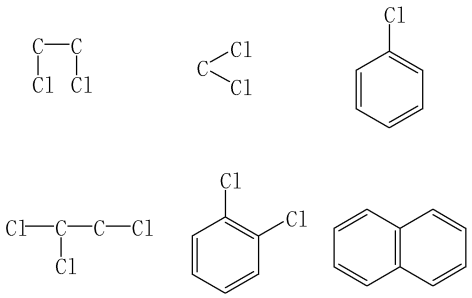

図 17 正例から抽出されたパターンの一例

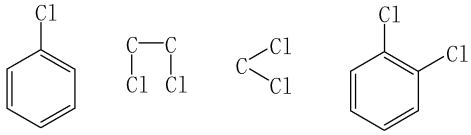

図 18 負例から抽出されたパターンの一例

ターンを有効なルールとしてみることができる（ 図 19 参照）.

GBI 法は Greedy 探索のため，すべての類型的パターンを抽出できるわけではない．そこで，類型的パターンがどれくらい抽出されているかを評価するため， AGM [Inokuchi 00] で抽出されたパターンと比較した．AGM は猪口らによって提案された *Apriori* アルゴリズムに基づいた部分グラフ抽出アルゴリズムである．AGM は連結・非連結のすべての部分グラフを抽出するアルゴリズムであるが，完全探索であるために膨大な計算時間を要する．

現プログラムで数えているペアの頻度は，図 16 に示すように，すでにチャンクされたパターンを元のパターンに戻して評価することはしないために実際の頻度よりは少なく数えられる．そのため，AGM で設定した最小支持度と同じ閾値を用いると，同様の類型的パターンを得ることはできないが，これは閾値を低く設定することで解決できる．AGM の計算時間は最小支持度に対して指数関数的に増加するため，最小支持度を 15 % に設定し AGM によるパターン抽出を行った．その結果，AGM で得られた類型的パターンは 454 種類であるが，AGM は非連結部分グラフの抽出も行うため，実際に GBI 法で得られる可能性のある類型的パターンは 18 種類であった．そのうち，GBI 法で得ることのできたパターンは 11 種類であった．さらに 18 種類の類型的パターンのうち，得ることのできなかった残りの 7 種類の類型的パターンは，得ることのできた類型的パターンの部分グラフに含まれており[*5]，そのパターンも含めると 18 種類の全パターンを得ることができていた．AGM が部分構造を抽出するのに同じマシンスペックで 17 分を要したことを考慮すると，GBI 法はそれよりもはるかに高速（約 1 秒）にすべての類型的パターンを発見することができている．

現在のプログラムはノードに番号を付けることによってグラフデータを管理しているため，ノードの番号付けによって抽出されるパターンが変化する可能性がある．そのため，ランダムにノード番号を付け替えてパターンの抽出を行い，AGM の抽出結果と比較するという試行を 10 回繰り返したが，どの試行でも結果は変わらなかった．

5·2 変異原性物質データへの適用

変異原性物質は，しばしば DNA の構造的な変化を引き起こすことが知られている．DNA の変異は進化するために重要な要素であるが，多くの場合は健康に害を与える結果となる．また，変異原性と発ガン性の間には，高い相関関係があることが知られている．しかし，化学化合物をすべて実験的な手法で評価することは，時間やコストの関係から不可能である．そのため，化学構造から変異原活性を評価することは非常に有用である．変異原性のメカニズムは非常に複雑で，ほんの一部分しか知られていない．このメカニズムについての様々なクラスの化学物質に対する兆候を見つけることができ，それにより変異原性のメカニズムを解明する鍵となるような仮説を導くことができれば，変異原性の研究において非常に重要な発見となるに違いない．

本実験で用いたデータは [Debnath 91] から取得した．このデータは，230 個の芳香族ニトロ化合物と複素環芳香族ニトロ化合物で構成されている．変異原活性は次のように 4 つに離散化し，これをクラスとして用いた．

- Inactive : $activity = -99$

*5 $a \to b \to c$ で $a \to b$ がまずチャンクされ，次に $(a,b) \to c$ とチャンクされた場合の $b \to c$ のようなパターン．

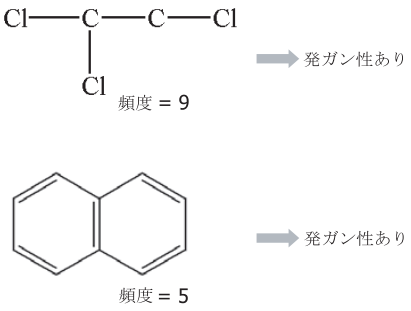

図 19 有機塩素化合物からの抽出パターン例

- Low : $-99 < activity < 0.0$
- Medium : $0.0 \leq activity < 3.0$
- High : $3.0 \leq activity$

これにより，クラス Inactive の化合物は 22 個，クラス Low の化合物は 68 個，クラス Medium の化合物は 105 個，クラス High の化合物は 35 個となった．各々の化合物にはさらに属性として，LogP 値[*6]と LUMO 値[*7]があるが，最初の実験ではまずこれらの属性を無視した．リンクの属性としては，単結合，2 重結合，3 重結合と芳香族結合を用い，ノードのラベルとしては，炭素，塩素，窒素を用い，水素は無視した．さらに，任意のラベルのノードつまり任意の原子を扱うために人工的にノードを 1 〜 5 個飛ばしてつないだリンクを生成したデータでも評価した．このようにリンクを張ることで，5 連続までの任意の原子を含んだパターンを抽出できる．パターン抽出のための評価関数には頻度を用い，人工的なリンクがないデータではペアの頻度が 2 以上，人工的なリンクがあるデータではペアの頻度が 5 以上の場合チャンクを行った．

この解析では，GBI 法は属性構築のために用いた．つまり，GBI 法により抽出されたパターンを新しい属性として用い，C4.5 により分類規則学習を行った．まず，GBI 法によりパターンの抽出を行い，入力データ中に含まれる化合物のクラス比を考慮したクラスの支持度の最大値をあらわす式 (1) を用いて得られた各パターンの評価値を算出した．

$$Max.\left\{\frac{\frac{i}{I}}{\frac{i}{I}+\frac{l}{L}+\frac{m}{M}+\frac{h}{H}}, \frac{\frac{l}{L}}{\frac{i}{I}+\frac{l}{L}+\frac{m}{M}+\frac{h}{H}}, \frac{\frac{m}{M}}{\frac{i}{I}+\frac{l}{L}+\frac{m}{M}+\frac{h}{H}}, \frac{\frac{h}{H}}{\frac{i}{I}+\frac{l}{L}+\frac{m}{M}+\frac{h}{H}}\right\} \quad (1)$$

ここで，i，l，m，h は評価するパターンを含むそれぞれのクラスごとの化合物の数で，I，L，M，H は入力データのクラスごとの化合物数である．

*6 LogP 値とは，化合物の疎水性係数の log をとったものである．

*7 LUMO 値とは，化合物の最低空分子軌道のエネルギーの値である．

この評価値による上位 n 個（n =10,20,30,40,50:人工的なリンクなし，n =10,20,30,40,50,60,70,80,90,100:人工的なリンクあり）のパターンを新しい属性として用いた．

図 20，図 21，図 22，図 23 はチャンクのステップ（図 3 の 3) 〜 7) のループ）に対するチャンクするペアの頻度の推移とパターンの評価値，そしてチャンクの大きさ（チャンク内に含まれているノード数）の推移を示している．それぞれのチャンクは異なる経歴を持っているため，評価値とチャンクの大きさは，チャンクのステップに対して単調になっていない．なお，帰納論理プログラム (ILP) の枠組みで，グラフ構造を一階述語論理で直接表現し，その中の部分構造を発見するアルゴリズムが Dehaspe らにより提案されている [Dehaspe 98]．この手法では探索効率を上げるために ILP にレベルワイズ探索を組み合わせている．しかし，それでも探索空間が大きすぎるために，発ガン性物質の解析では，現実的な時間では最大でも 6 つの節，つまり 3 原子位の大きさの部分構造を見つけることしかできない．しかし，GBI 法では 図 21 , 図 23 からわかるように 29 原子からなる部分構造を見つけることができている．なお，GBI によるパターン抽出にかかった計算時間は約 4 秒（CPU:Pentium III 600 MHz，Memory: 384 MB）であった．このように，GBI 法はグラフ構造から高速に部分構造を抽出することのできるアルゴリズムであるといえる．

表 6 は，パラメータの値を様々に変化させたときの C4.5 による予測誤差率を示している．“All” はチャンクパターンと $LogP$ 値と $LUMO$ 値を属性として用いたもので，“w/o LUMO”，“w/o LogP” はそれぞれ “All” から $LUMO$ 値あるいは $LogP$ 値を除いたもの，“patt. only” は GBI 法で抽出されたパターンのみを属性として用いたものである．また，“w/o patt” は $LogP$ 値と $LUMO$ 値を，“only LUMO”，“only LogP” はそれぞれ $LUMO$ 値のみあるいは $LogP$ 値のみを属性として用いた場合のエラー率を示している．“w/o cv （cross validation なし）” は全てのデータで分類木を構築し，それを用いて全てのデータを再分類したときのエラー率である．この表より $LogP$ 値が最も効いている属性であることがわかるが，上手くパターンを選ぶことにより，パターンのみでも $LogP$ 値と同等くらいの分類精度があることがわかる．また，全ての属性を用いることにより，エラー率は 13.9 % にまで下がることがわかる．

次に 4•1 節 で述べた方法により，抽出されたチャンクパターンのみを用いて 10-fold cross validation を行った．データは人工的なリンクがないものを用いた．この方法では，抽出された各々のパターンは式 (1) を用いて評価し，パターンを条件部，この式による評価値を最大にするクラスを結論部とすることによりルールとして用いた．そして，それらのルールを (1) の評価値の大きいものから適用した．評価値が同じ場合はサイズ（パターン内のノード数）がより大きなルールから適用した．評

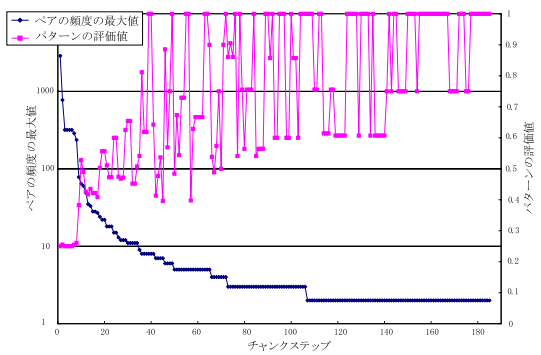

図 20 ペアの最大数，クラス予測関数の値とチャンクのステップとの関係（人工的なリンクなし）

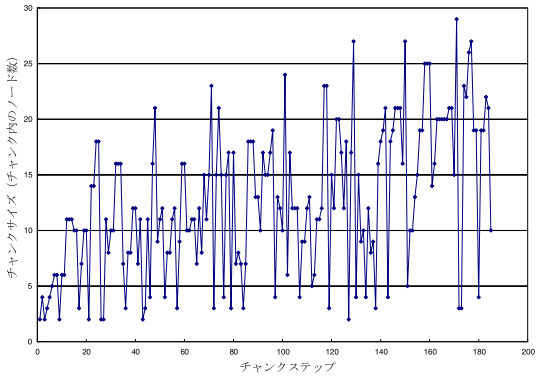

図 21 チャンクサイズ（チャンク内のノード数）とチャンクのステップとの関係（人工的なリンクなし）

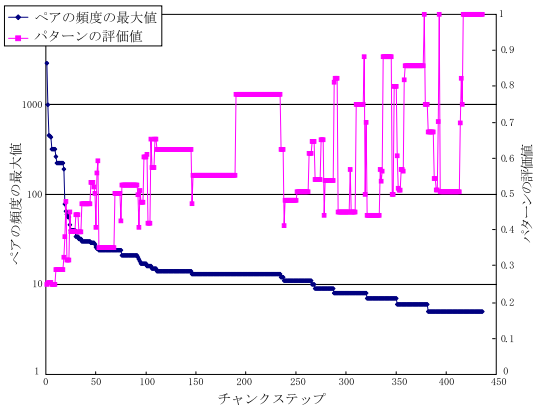

図 22 ペアの最大数，クラス予測関数の値とチャンクのステップとの関係（人工的なリンクあり）

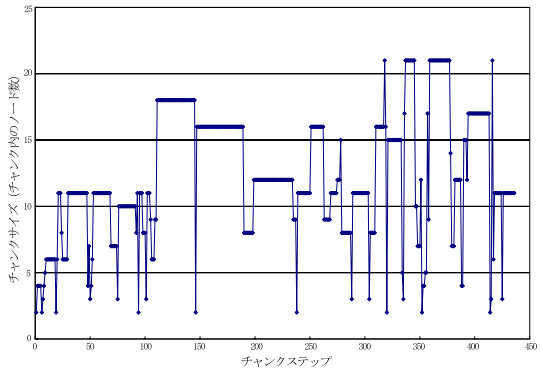

図 23 チャンクサイズ（チャンク内のノード数）とチャンクのステップとの関係（人工的なリンクあり）

価値が大きいものから適用したのは，よりクラス分解能が高いと思われるからである．また，パターンのサイズがより大きなルールから選んだのは，より具体的なパターンから適用するということを意図している．前出の方法と比較するため，適用するルールを上位 10 ， 20 ， 30 ， 40 ， 50 ， 60 ， 70 ， 80 ， 90 , 100 個にした場合と，全てのルールを用いた場合についてそれぞれ評価した．結果を 表 7 に示す．なお，この表の誤差率は *Unknown* を除いて計算している．平均的に誤差率は約 45 % 程度とあまりよくない．しかし， 表 6 のパターンのみを用いた C4.5 によるクロスバリデーションでも予測誤差率が同じくらいになっていることより，この手法は C4.5 と同程度の分類能力を持っているといえる．見かけの誤差が悪く，さらに 10-fold cross validation で予測精度が低くなっている理由としては，抽出されたパターンが，分類能力があるものというよりは記述的なものであったためだと思われる．これは，用いるルール・パターン数を増やしてもクロスバリデーションの結果があまりよくなっていないことからわかる．つまり，この手法がうまく機能していないというよりは，分類規則学習に用いた属性が分類にあまり寄与しないものであったためと思われる．つまり，我々の用いた属性のみでは分類には不十分であり，さらに 3-D 構造情報などのいくつかの属性を加える必要があることを示唆している．いくつかの抽出されたルールの例を 図 24 に示す．本実験では，任意のラベルのノード（任意の原子）が含まれている有用と思われるパターンは見つからなかったが，これは水素原子を取り扱わなかったためと思われる．

6. お わ り に

本研究では，一般グラフ構造データに対する GBI 法を実装し，実装プログラムの計算時間について理論的および実験的な観点から評価した．その結果，実験的な評価

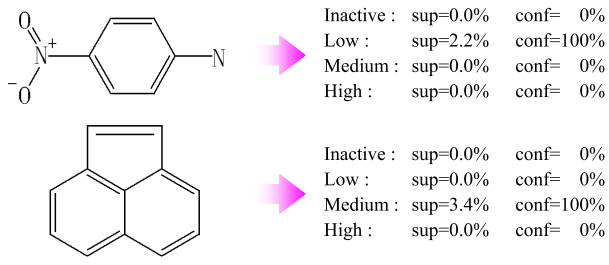

図 24 変異原性物質データから得られたルールの一例

表 6 様々な属性を用いた C4.5 による変異原性物質の予測誤差（%）：上側（人工的なリンクなし），下側（人工的なリンクあり）

属性に用いたパターン数	all		w/o LUMO		w/o LogP		patt. only	
	w/o cv	10 fcv	w/o cv	10 fcv	w/o cv	10 fcv	w/o cv	10 fcv
10	15.7	44.8	20.9	43.9	25.7	47.4	40.4	44.8
20	14.8	45.6	18.3	43.0	23.0	50.4	35.7	45.6
30	13.9	44.8	17.8	41.3	21.3	49.6	35.7	47.4
40	13.9	39.6	20.0	41.3	20.4	51.8	30.9	49.1
50	15.2	47.0	19.6	47.0	20.9	47.8	29.6	49.1
属性に用いたパターン数	all		w/o LUMO		w/o LogP		patt. only	
	w/o cv	10 fcv	w/o cv	10 fcv	w/o cv	10 fcv	w/o cv	10 fcv
10	21.3	44.8	29.1	52.2	30.9	46.5	48.3	51.3
20	17.0	41.7	23.9	49.1	25.7	47.0	38.2	55.2
30	18.3	48.7	21.3	52.6	22.2	48.7	33.9	48.7
40	17.4	49.1	22.2	52.6	23.5	50.9	31.7	50.9
50	14.3	47.4	20.4	51.7	17.8	48.7	30.4	50.9
60	13.0	47.4	19.6	53.1	16.5	46.5	28.3	52.2
70	13.5	43.9	20.4	49.6	18.3	44.8	28.3	47.4
80	13.5	50.4	18.3	45.6	17.0	50.4	26.1	46.1
90	14.8	49.1	18.7	47.8	17.8	50.0	26.1	45.2
100	14.3	46.1	18.3	46.9	17.8	46.9	24.8	45.2

w/o patt.	
w/o cv	10 fcv
27.8	40.9
only LUMO	
44.8	47.8
only LogP	
30.4	47.4

表 7 変異原性物質の予測誤差（2）

ルール数	正解	不正解	*Unknown*	エラー率 (%)	ルール数	正解	不正解	*Unknown*	エラー率 (%)
10	3	6	221	66.7	70	60	43	127	41.7
20	12	8	210	40.0	80	79	68	81	46.3
30	24	18	188	42.9	90	102	88	40	46.3
40	32	27	171	45.8	100	118	105	7	47.1
50	46	36	148	43.9	*all*	121	109	0	47.4
60	53	39	138	42.4					

においても理論的な評価と同様に，実装プログラムの計算時間はグラフのノード数に対してほぼ線形オーダーであることが示された．さらに応用例として，この GBI 法を DNA 塩基列の分類規則学習および有機塩素化合物からのパターン抽出に応用し，その有効性を検証した．また，GBI 法を分類規則学習法のための属性構築に用いることのできる可能性を示した．

今後の課題として以下のようなものがあげられる．GBI 法は基本的に Greedy 探索であるためノードの番号付けにより抽出パターンが変化する可能性があると考えられる．今回の実験では，有意な差は見られなかったが，その傾向を調査するとともに，ノードの番号付けによらずになるべく多くの重要な特徴的パターンを抽出できるようにする必要がある．また，今回は，概念を頻度により特徴付けたが，頻度以外で特徴付けることのできる概念も考えられるため，それらを取り扱えるような評価関数を考える必要がある．あるいはドメイン知識を用いてチャンクを制御することにより，Greedy 探索の範囲でもそのドメインにとって有意な類型的パターンを取り出せるようにすることが考えられる．さらに，今回は無向グラフを近似的に有向グラフに変換して扱ったが，GBI 法で無向グラフを厳密に扱えるように改良する必要がある．

謝　　辞

本研究において，化学データの提供および御指導してくださった関西学院大学情報メディア教育センターの 岡田 孝 教授に感謝いたします．本研究の一部は文部省科学研究費基礎研究 B (2)(国 11694159，12480088) の補助による．

◇ 参 考 文 献 ◇

[Agrwal 94] Agrwal, R. and Srikant, R.: First Algorithms for Mining Association Rules, in *Proc. of the 20th VLDB Cnference*, pp. 487–499 (1994).

[Blake 98] Blake, C. L., Keogh, E., and Merz, C.: UCI Repository of Machine Leaning Database (1998), http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[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).

[Breiman 89] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J.: The CN2 Induction Algorithm, *Machine Learning*, Vol. 3, pp. 261–283 (1989).

[Debnath 91] Debnath, A. K., Compadre, Lopez de R. L., Debnath, G., Shusterman, A. J., and Hansch, C.: Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Compunds. Correlation with Molecular Orbital Energies and Hydrophobicity, *J. Med. Chem.*, Vol. 34, pp. 786–797 (1991).

[Dehaspe 98] Dehaspe, L., Toivonen, H., and King, R. D.: Finding frequent substructures in chemical compound, in *Proc. the 4th International conference on Knowledge Discovery and Data Mining*, pp. 30–36 (1998).

[Inokuchi 00] Inokuchi, A., Washio, T., and Motoda, H.: An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data, in *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 13–23 (2000).

[Matsuda 00a] Matsuda, T., Horiuchi, T., Motoda, H., and Washio, T.: Extension of Graph-Based Induction for General Graph Structured Data, in *Knowledge Discovery and Data Mining: Current Issues and New Applications, Springer Verlag, LNAI 1805*, pp. 420–431 (2000).

[Matsuda 00b] Matsuda, T., Horiuchi, T., Motoda, H., and Washio, T.: Graph-Based Induction for General Graph Structured Data and Its Application to Chemical Compound Data, in *Proc. of the 3rd International conference on Discovery Science* (2000).

[松本 99] 松本, 田辺：ニューラルネットワークによる有機塩素化合物の発ガン性物質予測, *JCPE Journal*, Vol. 11, No. 1, pp. 29–34 (1999).

[Michalski 90] Michalski, R. S.: Learning Flexible Concepts: Fundamental Ideas and a Method Based on Two-Tiered Representaion, *In Machine Learning, An Artificial Intelligence Approiach*, Vol. 3, pp. 63–102 (1990).

[Muggleton 94] Muggleton, S. and Raedt, de L.: Inductive Logic Programming: Theory and Methods, *Journal of Logic Programming*, Vol. 19, No. 20, pp. 629–679 (1994).

[Quinlan 86] Quinlan, J. R.: Induction of decision trees, *Machine Learning*, Vol. 1, pp. 81–106 (1986).

[Quinlan 93] Quinlan, J. R.: *C4.5:Programs For Machine Learning*, Morgan Kaufmann Publishers (1993).

[吉田 92] 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 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).

〔担当委員：沼尾正行〕

2000 年 11 月 14 日 受理

著 者 紹 介

松田 喬(学生会員)

1999 年大阪大学工学部通信工学科卒業．2000 年同大学院博士前期課程修了．現在，同大学院博士後期課程在学中．グラフ構造データからのデータマイニング・知識発見に関する研究に興味を持つ．人工知能学会会員．

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終了．同年，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，データマイニングなどの研究に従事．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，同編集委員会委員，日本認知科学会編集委員会委員，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，International Journal of Human-Computer Studies (Academic Press) 編集委員，Knowledge and Information Systems: An International Journal (Springer-Verlag) 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．1997 年人工知能学会研究奨励賞受賞，1997, 1998 年人工知能学会全国大会優秀論文賞受賞．人工知能学会，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．

鷲尾 隆(正会員)

1960 年生．1983 年東北大学工学部原子核工学科卒業．1988 年東北大学大学院原子核工学専攻博士課程修了．工学博士．1988 年から 1990 年にかけてマセチューセッツ工科大学原子炉研究所客員研究員．1990 年（株）三菱総合研究所入社．1996 年退社．現在，大阪大学産業科学研究所助教授（知能システム科学研究部門）原子力システムの異常診断手法に関する研究，定性推論に関する研究を経て，現在は人工知能の基礎研究，特に科学的知識発見，データマイニングなどの研究に従事．1988 年 2 月計測自動制御学会学術奨励賞受賞，1995 年 8 月人工知能学会全国大会優秀論文賞受賞，他 2 件．1996 年 3 月日本原子力学会論文賞受賞，1996 年 12 月人工知能学会研究奨励賞受賞，他 1 件．著書に “*Expert Systems Applications within the Nuclear Industry*”, American Nuclear Society,「知能工学概論」: 第 2 章エージェント（共著，廣田 薫 編，昭晃堂）など．AAAI，人工知能学会，計測自動制御学会，情報処理学会，日本ファジィ学会，各会員．