特集論文

最小記述長原理を用いた帰納学習のRipple Down Rules法への統合化

Integrating Inductive Learning to the Ripple Down Rules Method with the Minimum Description Length Principle

和田 卓也 Takuya Wada
大阪大学産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University.
wada@ar.sanken.osaka-u.ac.jp

元田 浩 Hiroshi Motoda
（同上）
motoda@ar.sanken.osaka-u.ac.jp

鷲尾 隆 Takashi Washio
（同上）
washio@ar.sanken.osaka-u.ac.jp

keywords: integration, inductive learning, knowledge acquisition, ripple down rules method, minimum description length principle

Summary

A Knowledge Acquisition method “Ripple Down Rules” can directly acquire and encode knowledge from human experts. It is an incremental acquisition method and each new piece of knowledge is added as an exception to the existing knowledge base. This knowledge base takes the form of a binary tree. There is another type of knowledge acquisition method that learns directly from data. Induction of decision tree is one such representative example. Noting that more data are stored in the database in this digital era, use of both expertise of humans and these stored data becomes even more important. In this paper, we attempt to integrate inductive learning and knowledge acquisition. We show that using the minimum description length principle, the knowledge base of Ripple Down Rules is automatically and incrementally constructed from data and thus, making it possible to switch between manual acquisition by a human expert and automatic induction from data at any point of knowledge acquisition. Experiments are carefully designed and tested to verify that the proposed method indeed works for many data sets having different natures.

1. はじめに

技術革新の激しい今日，既存の知識は急速に価値を失うことも多く，最新の知識を常に保持するためには，頻繁な知識の更新が必要となることもある [Morik 93, Wrobel 94]．そのような状況で実用的な知識ベースシステムを構築する枠組みとして，Ripple Down Rules 法 [Compton 89]（以下 RDR 法と呼ぶ）と呼ばれる人間の専門家からの知識獲得を目的とした手法が有望と我々は考えている．RDR 法では知識獲得を既存の知識の逐次的な洗練と捉え，システムに入力される事例が誤って判断されるたびに，今まで正しく判断を下してきた知識の価値を下げることなく，新しい知識を既存の知識の例外知識として追加する．知識獲得は専門家とインタラクティブに行われ，知識獲得段階とシステム保守段階の間に明確な境界はない．

RDR 法は元来，人間の専門家から知識を獲得する手法であるので，人間の専門家の判断力に強く依存する．専門家は具体的な事例を前にすれば，比較的容易になぜこの事例がシステムによって誤判断されたのかを説明でき，どのような対処法が必要なのかも判断できる．しかし専門家といえども決して全能者ではなく，時には間違いも犯す．一方，機械学習の最近の進展（例えば C4.5 [Quinlan 93] による決定木や，そのような分類器を集団で用いるブースティング手法 [Freund 99]）により，多量のデータから効率的に分類器を帰納的な方法で構築できるようになった．国や企業のデータベースには既に大量のデータが蓄積されている．

本論文では，帰納学習手法（データから RDR 法の知識ベースを構築する機械学習手法）と従来の RDR 法（人間の専門家から逐次的に専門知識を獲得する知識獲得手法）を統合する可能性を調べる．この統合化のための基本

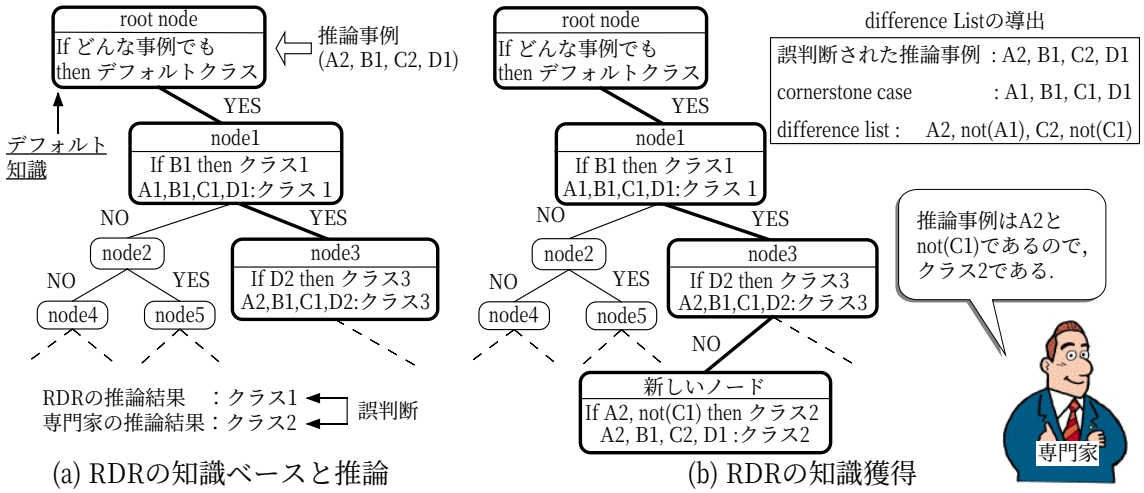

図 1 RDR の知識構造と知識獲得アルゴリズム

原理として，最少記述長原理 [Rissanen 78]（Minimum Description Length 原理:MDL 原理）を用いることを提案する．データから RDR 法の知識ベースを帰納的に構築する方法は既に Gaines と Compton によって研究されている [Gaines 92, Gaines 95]．彼らは，ある与えられた結論を偶然によって予測する可能性が最も少ない前提を探索する Induct アルゴリズム [Gaines 89, Gaines 91] を用いている．この方法でも RDR 法の知識ベースは生成されるが，知識獲得アプローチと機械学習アプローチを統合化する共通の戦略がない．また，Induct はバッチ処理を前提にしている．

統合化の実現により期待されることとして，利用できるデータがまだ多く蓄積されていない開発初期段階で人間の専門家が知識ベースを構築し，十分データが蓄積されている段階になれば帰納法的な技術によって構築した知識ベースを洗練する，また逆に開発段階において，既に蓄積されている大量データにより知識ベースを構築し，実用段階で利用環境に適するよう，専門家が知識ベースを洗練するというような柔軟な知識ベースシステムの構築方法が挙げられる．そのためには，少なくとも従来の RDR 法と同じ利点（知識ベース上の整合性や逐次的な知識獲得）を持つ知識ベースを提案手法は構築できなければならない．

本論文では 21 種類のデータセットを用いて，提案手法を用いた評価実験等を行い，MDL 原理が統合化を実現するために有効であることを確認する．

2. Ripple Down Rules 法のアルゴリズム

RDR 法は知識エンジニアによる分析やインタビューなどを必要とせず，専門家が直接知識ベースシステムを開発することによって，知識獲得ボトルネックを解消しようとする知識獲得技術である．RDR 法では知識ベースの整合性維持と事例に対する推論が基本である．RDR システムでは，ある事例が誤って推論された場合，知識獲得（保守）段階において，なぜその事例に対する推論が知識ベースに蓄積されている過去に正しく推論された事例に対する推論と異なるのかを専門家に判断させる必要がある．

RDR 法の知識ベースは，図 1.(a) に示すように二分木により表現することができる．各ノードは If-Then ルール（知識そのもの）とそのノード自身が追加される原因となった事例（cornerstone case）を記憶している．また二つの枝として YES 枝と NO 枝を持つ．

ある事例が与えられた場合，その事例に対する結論がどのように推論されるのかを以下に説明する．RDR 法における推論過程は二分木のルートノードから始まる．与えられた事例がルールの条件部を満足すれば YES 枝に，そうでなければ NO 枝に，その推論過程を移行する．そしてこれ以上進むべきノードが存在しなくなるまでこの過程を繰り返す．最終的に，事例に対する結論は推論パス上で一番最後に条件部が満足されたルール（last satisfied rule）の帰結部となる．

もし事例に対する推論結果が専門家の判断する結果と異なる場合，専門家から知識（新しいルール）を獲得し，既存の二分木に追加する．図 1.(b) に RDR 法における知識獲得の様子を示す．まず新しく知識ベースに追加するためのルールの条件部を決める．そのために RDR システムは誤った結論を与えたルール（last satisfied rule）を持つノード（last satisfied node）が記憶している cornerstone case と誤判断された事例の差として difference list を専門家に提示する．そして誤判断された事例に対する正しい結論を正当化するための条件をこのリストから専門家に選んでもらう．専門家が選んだ条件と新しい結論を，新しく追加するノードの If-Then ルールとし，誤判断された事例をこのノードの cornerstone case とする．

新しいノードの追加位置は，推論パスの一番最後のノード（end node）が last satisfied node であれば end node の YES 枝の下，そうでなければ end node の NO 枝の下とする．RDR 法では知識ベースを再編することなく，新しいノードを最後に条件部が満足されたルールを持つノードの例外ノードとして，cornerstone case とともに

表 1 データの例

ID No.	属性 Swim	属性 Breath	属性 Legs	クラス
1	can	lung	2legs	Dog
2	can	lung	4legs	Penguin
3	can	skin	2legs	Monkey
4	can	skin	4legs	Dog
5	can_not	lung	2legs	Cat
6	can_not	lung	4legs	Cat
7	can_not	gill	2legs	Dog
8	can_not	gill	4legs	Monkey
9	can_not	skin	2legs	Penguin
10	can_not	skin	4legs	Penguin

追加するだけである．従って知識ベース上の各知識（ルール）の整合性を保つことができ，各知識は常にそれが追加された場面と同じ文脈で用いられることになる．

ただし上述した RDR 法では，構築される知識ベースの性能（予測精度やノード数）に影響を与える二つの要因がある．一つはデフォルト知識と呼ばれるものであり，この知識のクラス（デフォルトクラス）は，ある事例に対して知識ベースから結論が導かれなかった際に，その事例の推論結果として選ばれる暗黙の結論となる．二分木のルートノードにおける If-Then ルールの条件部を空集合にした場合の帰結部がデフォルトクラスとなる．デフォルトクラスが異なれば，同じ問題領域でも性能の違う知識ベースが構築される [和田 00]．もう一つの要因は入力される推論事例の順番であり，デフォルト知識の場合と同じく，順番が異なれば異なる知識ベースが構築される．

3. MDL 原理の概念

Occam's razor の概念や MDL 原理は与えられた観測データをもとに，複数の選択肢の中から最良であろうと思われる確率モデルを選ぶ際の一般的な選択規範である．Occam's razor とは，データを説明するのに最もシンプルな仮説を選択すべきとの規範である．仮説の簡潔さは Rissanen [Rissanen 78] によって初めて提案された “記述長（Description Length:DL）”によって求めることが出来る．DL は与えられた仮説の複雑さを表しており，仮説が分類器の場合，その値はその仮説を符号化したときの符号長とその仮説によって誤分類されるデータを符号化したときに必要な符号長の合計で得られる．MDL 原理は，例えば決定木 [Quinlan 89]，ニューラルネット [Gary 93] そしてベイジアンネットワーク [Suzuki 93] などの機械学習の分野において，良好なモデルを選ぶための規範として用いられている．

ある通信問題を例に，MDL 原理の概念を説明する．送信者 A と受信者 B が表 1 に示したような事例群のリストを持つ状況を想定する．これは動物 {Cat, Dog, Monkey, Penguin} の分類問題であり，各事例は 3 種類の属性-属性値のペアとクラスからなる．ただし B のリストのクラス欄は空白であり，A が通信路を通してクラス欄の情報をなるべく少ない DL を使って B に伝えることを考える．

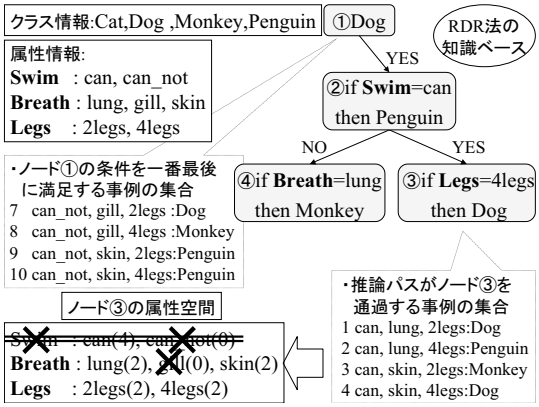

図 2 DL の計算を例証するための二分木の例

最も簡単な方法はクラス情報を直接送ることである（一番目の事例は Dog，二番目の事例は Penguin，...）が，ある属性とクラスとの間に強い関係がある場合，その依存関係を相手に教えれば，伝えるのに必要なビット数（DL）を劇的に小さくできる．そのために次の 4 ステップによる方法をとる．

Step 1： リストの事例群をその属性値に従ってある分割方法に基づき，部分集合に分割する．

Step 2： この分割方法の情報をある符号化方法に従い符号化して得られたビット列を送る．

Step 3： 各部分集合に対応する代表クラスの情報を符号化して送る．

Step 4： 代表クラスと異なるクラスを持つ例外的な事例に対し，正しいクラスの情報を符号化して送る．

1) 「(**Step 2**) において分割方法の情報を符号化して得られる DL と (**Step 3**) における代表クラスの DL」と 2) 「(**Step 4**) における誤事例のクラス情報の DL」の間にはトレードオフの関係がある．C4.5 における決定木や RDR 法における二分木のような知識ベースモデルは，分割方法とそれによって分割された部分集合に対する代表クラスから構成されていると見なすことができる．従ってある適切な符号化方法の下で，全体の DL は知識モデルに対する DL と誤事例に対する DL の和で計算される．MDL 原理に従えば，選択すべきモデルは全体の DL が最も小さくなるモデルである．RDR 法の場合，全体の DL が最も小さくなる二分木を選択すればよい．4 章で，具体的な DL の計算方法を説明する

4. 記述長の計算方法

この章では MDL 原理に沿って，我々の考えた符号化方法をもとに DL の計算方法を説明する．ただしここで示す方法が MDL 原理に沿う唯一のものであるわけではないが，6 章で得られる実験結果は我々の符号化方法が適当であることを示している．

4･1 節と 4･2 節では，図 2 に示した二分木と表 1 の事

例群を例に挙げながら，二分木に対する DL とその二分木によって誤判断される事例に対する DL の計算方法をそれぞれ述べ，4･3 節にて全体の DL について説明する．

4･1 二分木に対する DL

与えられた二分木の DL を計算する前に，各ノードにおける属性空間を調べるため，全事例に対してこの二分木で推論を行う．その結果，推論パスがあるノード（ノード C）を通過する事例が全部で k 個あったと仮定しよう（図 2 では，ノード③に対するそのような事例は {1,2,3,4} の四つである.）．これら k 個の事例から，ノード C における各属性値の頻度分布を調べる．そして少なくとも 1 事例の頻度がある属性値を，属性空間を構成する候補とし，そのような属性値が 2 個以上ある属性（すなわち，属性値が一義的に決まらない属性）の集合から，ノード C の属性空間を得る．仮に，その属性空間を構成する n 個の属性を $\{A_i|i=1,...,n\}$，そして属性 A_i を構成する m_i 個の属性値を $\{v_{i,j}|j=1,....,m_i\}$[*1]とする（ノード③の属性空間は { 属性 Breath:lung,skin 属性 Legs:2legs,4legs} (図の左下部）となる.）．

ノード C において符号化するべき情報は 2 種類：枝情報と If-Then ルール情報である（ノード③の場合，それぞれ {YES 枝:no NO 枝:no} と {If Legs=4legs then Dog} である.）．

ルートノードを除いた全てのノードに対する枝情報の候補は次の 4 種類:{YES 枝:yes NO 枝:yes},{YES 枝:yes NO 枝:no},{YES 枝:no NO 枝:yes} そして {YES 枝:no NO 枝:no} である．従って，ノード C における枝情報 {YES 枝:no NO 枝:no} を符号化するのに $\log_2{}_4C_1$ ビット[*2]必要である（ノード③の枝情報に対する DL も $\log_2{}_4C_1$ ビットである.）．

ルール情報に関しては，まずその情報を次の四つの情報：(**1**){ 条件部に用いる属性の数 }，(**2**){ 用いられる属性 }，(**3**){ 各属性に対する値 } そして (**4**){ 帰結部 } に分ける（ノード③の場合，(**1**){ 属性の数:1}，(**2**){ その属性:Legs}，(**3**){ 属性 Legs の値:4legs}，(**4**){ 帰結部:Dog} となる.）．

情報 (**1**) を符号化するためには，n 個の候補 {1,2,...,n} があるので，$\log_2{}_nC_1$ ビット必要である（ノード③の場合，その属性空間が 2 種類の属性から成っているので，$\log_2{}_2C_1$ ビットである.）．ここで条件部には $t(1 \leq t \leq n)$ 個の属性が用いられていると仮定すると，情報 (**2**) を符号化するのに $\log_2{}_nC_t$ ビット必要になる（ノード③では $\log_2{}_2C_1$ ビットである.）．なぜなら用いられている t 個の属性と用いられていない $n-t$ 個の属性の組み合わせは ${}_nC_t$ 通り考えられるからである．情報 (**3**) を符号化するためには，条件部で用いられている属性 A_i に対して，それぞれ $\log_2{}_{m_i}C_1+\log_2{}_2C_1$ ビット必要である（ノード③では，属性 Legs だけが用いられているので，$\log_2{}_2C_1+\log_2{}_2C_1$ ビットである.）．上式の第 2 項はその条件が肯定の条件か否定の条件を示すのに必要な DL である．最後に情報 (**4**) を符号化するためには，ルートノード以外の各ノードで帰結部として使用される可能性があるクラスの数は $class\_num-1$ であるので，$\log_2{}_{class\_num-1}C_1$ ビット必要である（ノード③では，Cat，Dog そして Monkey の三つの候補があるので，$\log_2{}_3C_1$ ビットである.）．ただし，$class\_num$ はその問題領域でとりうるクラスの数を表している．情報 (**1**) から (**4**) の合計がノード C におけるルール情報を符号化するのに必要な DL となる[*3]．

最初に説明した枝情報の DL とルール情報の DL の合計がノード C に対する DL となる（ノード③に対する DL は $4\log_2{}_2C_1+\log_2{}_3C_1+\log_2{}_4C_1$ ビットとなる.）．もし送信者 A が二分木の各ノードの情報をルートノードから順に符号化し，得られたビット列を受信者 B に送れば，B は送られてきたビット列を復号化することで，用いられた分割方法と各部分集合に対する代表クラスを知ることが出来る．

4･2 誤事例に対する DL

次に，与えられた二分木によって誤判断される事例に対する正しいクラス情報を符号化するために必要な DL の計算方法を説明する．4･1 節と同じく，この DL は二分木の各ノードにおいて計算される．

リストに載っている事例を全て推論した結果，ノード D を last satisfied node とする事例が r 個あると仮定する．つまり，これら r 個の事例の集合はノード D の帰結部を代表クラスとする部分集合である（図 2 のノード①の場合は {7,8,9,10} の 4 個である.）．さらに r 個のうち k 個の事例は代表クラスと異なるクラスを持つと仮定する（ノード①では，4 個のうち 3 個の事例が代表クラス Dog と異なるクラスを持っている.）．

まず，r 個の事例群に対して,「代表クラスと異なるクラスが何種類あるか」という情報を符号化する必要がある．r が対象問題領域におけるクラスの数よりも多い場合は，その情報に対する候補は必ず $\{0,1,...,class\_num-1\}$ の $class\_num$ 個考えられる．しかしそうでない場合，r 個の候補 $\{0,1,...,r-1\}$ を考慮するだけで十分であるので，この情報を符号化するためには，

$$\begin{cases} \log_2{}_{class\_num}C_1 (r > class\_num \text{ の場合}) \\ \log_2{}_rC_1 \qquad\qquad (\text{そうでない場合}) \end{cases} \tag{1}$$

ビットだけ必要である（ノード①の場合，$\log_2{}_4C_1$ ビットとなる.）．

---

*1 $v_{i,j}$はどれも頻度 1 事例以上の属性値である．

*2 ルートノードの場合，その候補は {YES 枝:yes NO 枝:no} と {YES 枝:no NO 枝:no} の 2 種類だけであるので，$\log_2{}_2C_1$ ビットだけ必要である．

*3 ルートノードの場合，条件部は予め空集合とわかっており，帰結部の候補は $class\_num$ 個あるので，$\log_2{}_{class\_num}C_1$ ビットだけである．

表 2 k 個の事例に対する正しいクラスを特定するために必要な DL を計算するアルゴリズム

```
function DescriptionLength: real;
variable dl: real;
         case, i, j, pre_num: integer;
begin
  dl := 0;
  case := r;    #ノード D を last satisfied node とする
                                          事例の数
  i := s;              # 代表クラスと異なるクラスの数
  j := class_num;          # 問題領域のクラスの数
  pre_num := ∞;                            # ダミー
  repeat
    dl := dl + log2(j − 1);      # i 番目のクラスを
                                          特定する DL
    if pre_num > case − i then
      begin
        dl := dl + log2(case − i);
                              # pi の値を特定する DL
      end
    else
      begin
        dl := dl + log2(pre_num);
                              # pi の値を特定する DL
      end
    dl := dl + log2(caseCpi);
                        # pi 個の事例を特定する DL
    pre_num := pi;                      # pi そのもの
    case := case − pi;                # 残った事例の数
    i := i − 1;                     # 残ったクラスの数
    j := j − 1;
  until i = 0;
  DescriptionLength= dl;
end
```

次にどの事例がどのクラスであるかを特定するのに必要な DL を計算する．ここで代表クラスと異なる i 番目のクラスを持つ事例の頻度を $p_i(i = 1, 2, ..., s)$ と表す．ただし s は代表クラスと異なるクラスの数であり，クラスの順番は $p_s \geq p_{s-1} \geq ... \geq p_2 \geq p_1$ となるように予め並べ替える（ノード①では，$p_2 = 2$(Penguin),$p_1 = 1$(Monkey) となる.）．並べ替えを行った後，表 2 で示すアルゴリズムに基づき DL を計算する（表 3 はノード①における計算方法を説明している.）．

ただしルートノードにおける DL の計算の場合は，表 2 の**if-else** 文の個所を

```
if pre_num > case − i + 1 then
  begin
    dl := dl + log2(case − i + 1);
  end
else
  begin
    dl := dl + log2(pre_num);
  end
```

のように変更する．

式 (1) によるビット列と表 2 による “*DescriptionLength*” ビットのビット列を足し合わせた信号を復号化することによって，k 個の事例に対するクラスを知ることが出来る．残り $r - k$ 個の事例は代表クラスと同じクラスである．ルートノードからこのような符号化を順に行うことによって二分木により誤判断される事例の正しいクラス情報を符号化したビット列が得られる（ただし既に 4･1 節で述べた二分木の情報を得ていなければ，そのビット列を復号化することはできない）．

表 3 ノード①における 3 個の事例に対する DL の計算方法

```
function DescriptionLength: real;
variable dl: real;
         case, i: integer;
begin
  dl := 0;
  case := 4;    # ノード①を last satisfied node とする
                                  事例 {7,8,9,10} の数
  i := 2;                  # 代表クラスと異なるクラス
                                  {Monkey,Penguin} の数
  j := 4;          # 動物の分類問題におけるクラスの数
  dl := dl + log2(3); # {Cat,Monkey,Penguin} の内,
                                  Penguin を特定する DL
  dl := dl + log2(3);          # {3,2,1} の内，p2 = 2 を
                                  特定する DL
  dl := dl + log2(4C2); # p2 個の事例を特定する DL
  case := case − 2;              # 残りの事例 {7,8} の数
  i := i − 1;              # 残りのクラス {Monkey} の数
  j := j − 1;
  dl := dl + log2(2); # {Cat,Monkey} の内，Monkey
                                  を特定する DL
  dl := dl + log2(2);            # {2,1} の内，p1 = 1 を
                                  特定する DL
  dl := dl + log2(2C1); # p1 個の事例を特定する DL
  case := case − 1;                # 残りの事例 {7} の数
  i := i − 1;                  # 残りのクラス { ∅ } の数
  j := j − 1;
  DescriptionLength= dl;
end
```

4･3 全 体 の DL

最終的に全 DL（4･1 節と 4･2 節で述べた DL の合計）の長さのビット列を送ることによって，B はリストの全事例のクラス情報を知ることが出来る．RDR 法における MDL 原理によれば，A から B に送られる最も小さい全体の DL を導く二分木は未知事例（リストには載っていない事例）に対する予測精度が高いということになる．

ただし，ここで説明したものを含めて MDL 原理に基づいて知識ベースシステムを構築するために用いられる符号化方法のほとんどは誤事例のクラス情報に比べて，知識モデルの方を符号化するのに必要な DL を多く見積もりすぎてしまうきらいがあることが知られている [Quinlan 93].

このため，一般的に全体の DL を求める場合，(全体の DL) = (誤事例のクラス情報の DL) + $W \times$ (知識モデルの DL) という式を用い，C4.5 では $W = 0.5$ を採用している．提案した符号化方法では，誤事例のクラス情報は知識モデル（二分木）の情報を用いているので，モデルに独立に符号化する場合より誤事例に対する DL はさらに小さくなる．従って本研究では，実験の結果，経験的な値として $W = 0.3$ を採用することにした．

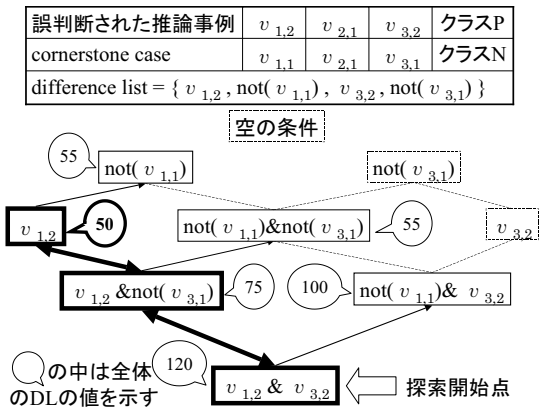

図 3 データだけを用いる場合の提案手法の例

5. MDL 原理に基づく知識獲得方法

本章では RDR 法において MDL 原理に基づいた 2 種類の知識獲得手法を提案する．データだけから二分木を帰納的に構築する方法（5･1 節）とデータと人間の専門家を用いて二分木を構築する方法（5･2 節）である．

5･1 データだけを用いた場合の方法

従来の RDR 法では，difference list から専門家に選ばせた要素の集合が新しいノードの条件部となる．しかし我々は MDL 原理に基づき，その list から選ばれうる集合のうち，新しいノードの条件として採用した場合の全体の DL が最小になる条件を探索したい．

そこで以下のような 6 ステップからなる提案手法を説明する．ただし対象問題領域の属性空間を構成する n 個の属性を $\{A_i|i=1,...,n\}$，そして属性 A_i を構成する m_i 個の属性値を $\{v_{i,j}|j=1,....,m_i\}$ とし，その集合を V_i と表す．既存の（その時点までに構築されている）二分木により誤判断された推論事例（事例 A）を $\{v_{1,a_1}, v_{2,a_2}, ..., v_{n,a_n}(v_{i,a_i} \in V_i)\}$，誤った結論を返した last satisfied node が持つ cornerstone case（事例 B）を $\{v_{1,b_1}, v_{2,b_2}, ..., v_{n,b_n}(v_{i,b_i} \in V_i)\}$ とする（図 3 は事例 A が $\{v_{1,2}, v_{2,1}, v_{3,2}\}$，事例 B が $\{v_{1,1}, v_{2,1}, v_{3,1}\}$ の場合の例を示している．ただし図 3 および図 4 における吹き出しの中の値は架空の値である.）.

Step 1 : 従来の RDR 法において専門家に提示される difference list は以下のように表現される．

$$difference\ list = \{v_{i,a_i}, not(v_{i,b_i}) | v_{i,a_i} \neq v_{i,b_i}, 1 \leq i \leq n\} \quad (2)$$

（図 3 では，$\{v_{1,2}, not(v_{1,1}), v_{3,2}, not(v_{3,1})\}$ となる.）説明の簡略化のために，difference list を改めて以下のように表現する．

$$difference\ list \equiv \begin{pmatrix} \alpha_1, \alpha_2, ..., \alpha_n \\ \beta_1, \beta_2, ..., \beta_n \end{pmatrix} \quad (3)$$

$$(\alpha_i, \beta_i) = \begin{cases} (0,0)(v_{i,a_i} = v_{i,b_i} \text{の場合}) \\ (1,1)(v_{i,a_i} \neq v_{i,b_i} \text{の場合}) \end{cases} \quad (4)$$

（例における difference list は $\binom{1,0,1}{1,0,1}$ である.）

Step 2 : difference list から選ばれる要素の集合のうち，事例 A に特化した集合（条件 $Start$）を以下のように定義する．

$$Start \equiv \begin{pmatrix} \alpha_1^S, \alpha_2^S, ..., \alpha_n^S \\ \beta_1^S, \beta_2^S, ..., \beta_n^S \end{pmatrix} \quad (5)$$

$$(\alpha_i^S, \beta_i^S) = \begin{cases} (0,0)(v_{i,a_i} = v_{i,b_i} \text{の場合}) \\ (1,0)(v_{i,a_i} \neq v_{i,b_i} \text{の場合}) \end{cases} \quad (6)$$

（例では，$\binom{1,0,1}{0,0,0}$ となる.）

Step 3 : 条件 $Start$ を新しいノードの条件部として採用した場合の全体の DL を計算する．

Step 4 : 条件 $Start$ と異なる複数の条件 $Start'$ をそれぞれ採用した場合の全体の DL を計算する．

ここで，条件 $Start'$（$= \binom{\alpha_1', \alpha_2', ..., \alpha_n'}{\beta_1', \beta_2', ..., \beta_n'}$）を以下のように定義する．

$\{\alpha_i = 1$ かつ$\beta_i = 1\}$ を満たす属性 A_i を任意に一つ選び，

$$(\alpha_i', \beta_i') = \begin{cases} (0,1)\ ((\alpha_i^S, \beta_i^S) = (1,0) \text{ の場合}) \\ (0,0)\ ((\alpha_i^S, \beta_i^S) = (0,1) \text{ の場合}) \end{cases} \quad (7)$$

または

$$(\alpha_i', \beta_i') = \begin{cases} (0,1)\ ((\alpha_i^S, \beta_i^S) = (0,0) \text{ の場合}) \\ (1,0)\ ((\alpha_i^S, \beta_i^S) = (0,1) \text{ の場合}) \end{cases} \quad (8)$$

を適用する．

選ばれた属性以外の属性に対しては，以下の式を適用する．

$$(\alpha_i', \beta_i') = (\alpha_i^S, \beta_i^S) \quad (9)$$

式 (7) または (8) と (9) から導かれうる条件が条件 $Start'$ となる．ただし条件 $\binom{0,0,...,0}{0,0,...,0}$ が生成された場合，その DL は計算しない.（例において条件 $Start$ が $\binom{1,0,1}{0,0,0}$ の場合，$\binom{0,0,1}{1,0,0}$ と $\binom{1,0,0}{0,0,1}$ に対する DL が計算される.）

Step 5 : もし **Step 3** での DL が **Step 4** で求めた DL の中で最小の DL よりも大きい場合，条件 $Start$ を最小の DL を導き出した条件 $Start'$ に更新し，**Step 3** に戻る．そうでなければ，**Step 6** に進む.（例において条件 $Start\binom{1,0,1}{0,0,0}$ の時，$\binom{1,0,1}{0,0,0}$, $\binom{1,0,0}{0,0,1}$, $\binom{0,0,1}{1,0,0}$ のそれぞれの DL が 120 ビット，75 ビットそして 100 ビットであるので $\binom{1,0,0}{0,0,1}$ に更新される．そして更新後，$\binom{1,0,0}{0,0,0}$, $\binom{0,0,0}{1,0,1}$, $\binom{1,0,1}{0,0,0}$ に対する DL は，それぞれ 50 ビット，55 ビットそして 120 ビットであるので，$\binom{1,0,0}{0,0,0}$ にさらに更新される．$\binom{0,0,0}{1,0,0}$, $\binom{1,0,0}{0,0,1}$ に対する DL はどちらも 50 ビットより大きいので，$\binom{1,0,0}{0,0,0}$ が条件 $Start$ の状態で **Step 6** に進む.）

Step 6 : もし条件 $Start$ を採用した場合の全体の DL が既存の二分木に対する DL よりも小さい場合，新しく

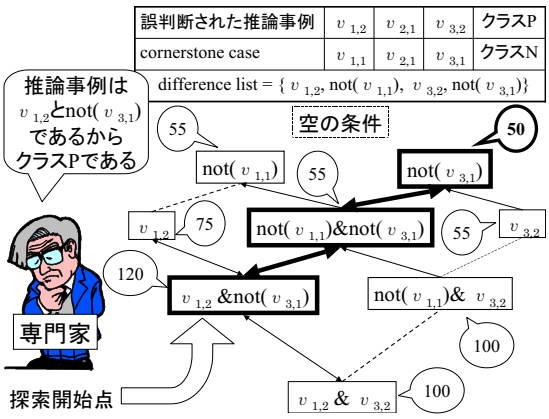

図 4 専門家とデータを用いる場合の提案手法の例

追加するノードの持つルールの条件部としてその条件を採用する．そうでなければ，新しいノードは追加しない．（例において既存の二分木に対する DL が $\binom{1,0,0}{0,0,0}$ を採用した場合の全体の DL：50 ビットより大きければ，条件部が $\{A_1 = v_{1,2}\}$ で，帰結部が { クラス P}，そして cornerstone case として $\{v_{1,2}, v_{2,1}, v_{3,2}\}$ を持つ新しいノードがその二分木に追加される．）

探索は事例 A に特化した条件から始まり，欲張り探索で全体の DL が少なくとも極小になる条件を探し出す．この方法[*4]により，人間の専門家に頼ることなくデータのみ[*5]から帰納的に RDR 法の知識ベースを構築することが可能である．全探索のほうがより予測精度の高い知識ベースを構築する可能性が高いと考えられるが，本研究では探索空間の縮小のため欲張り探索を採用した．

5･2 データと専門家を用いた場合の方法

5･1 節で述べた欲張り探索は誤判断された事例に特化した条件から探索が始まるが，それが探索開始条件として必ずしも良いとは限らない．もし従来の RDR 法と同じく専門家のアドバイスが利用できるならば，専門家の選んだ条件から探索を開始することも考えられる．

5･1 節の提案手法のアルゴリズムの **Step 2** を，

Step 2： 専門家によって difference list から選ばれた要素の集合を条件 $Start$ として，以下のように定義する．

$$Start \equiv \binom{\alpha_1^S, \alpha_2^S, ..., \alpha_n^S}{\beta_1^S, \beta_2^S, ..., \beta_n^S} \tag{10}$$

$$(\alpha_i^S, \beta_i^S) = \begin{cases} (0,0)(v_{i,a_i} = v_{i,b_i} \text{の場合}) \\ (0,0)(v_{i,a_i} \neq v_{i,b_i} \text{で，専門家が} \\ \quad v_{i,a_i} \text{も} \ not(v_{i,b_i}) \ \text{も選ばなければ}) \\ (1,0)(v_{i,a_i} \neq v_{i,b_i} \text{で，専門家が} \\ \qquad v_{i,a_i} \text{を選べば}) \\ (0,1)(v_{i,a_i} \neq v_{i,b_i} \text{で，専門家が} \\ \qquad not(v_{i,b_i}) \ \text{を選べば}) \end{cases} \tag{11}$$

とすれば，専門家の選んだ条件から探索を開始することができる（図 4 を例に挙げると，専門家が difference list$\{v_{1,2}, not(v_{1,1}), v_{3,2}, not(v_{3,1})\}$ から $v_{1,2}$ と $not(v_{3,1})$ を選んだので，条件 $Start$ は $\binom{1,0,0}{0,0,1}$ となる．）．この手法の利点は専門家が選んだ条件よりも，（MDL 原理の観点から見て）さらに良い条件を見つける可能性があることである．一般的に考えて，専門家の選ぶ条件が適切であればあるほど，全体の DL が小さい条件が探索される可能性は高い．

6. 実　　験

5 章で提案した手法の有効性を検証する前に，従来の RDR 法における二分木に対しても MDL 原理が成り立っていることを確認する必要がある．従って 6･1 節で，従来の RDR 法でも二分木の構築につれて全体の DL が小さくなるかどうかを調べる．そして 6･2 節では，5･2 節の手法を用いて構築した二分木と，従来の RDR 法で構築した二分木のエラー率を比較する．また 6･3 節では，標準的な機械学習法として選んだ C4.5 によって得られるインダクションルールセットと同程度の予測精度を持つ二分木が 5･1 節の手法により帰納的に構築できるかどうか調べる．

1　実験に用いたデータセット

California 大学 Irvine 校の機械学習データライブラリ [Blake 98] から 20 データセットを，Toronto 大学の機械学習評価用データライブラリ [DEL 98] から 1 データセットを選んだ（表 4 参照）．うち 11 個は全属性が離散値，7 個は全属性が連続値，そして残り 3 個は離散値属性と連続値属性の両方を含んでいる．ただし提案手法を計算機上に実装した我々のプログラムは離散値属性しか扱えないので，実験前に連続値属性に対して機械的な離散化を行った（C4.5 の結果を利用）．

2　代替専門家

実験の再現性と性能評価の必要性から，人間の専門家の代わりに，データ（6･1 節で説明する）を使って機械学習法 C4.5 で構築される決定木から導かれる If-Then ルール集合を代替専門家 [Compton 95]（Simulated Expert:SE）の知識源とする．このルール集合は実際には，構築された決定木のルートノードから葉ノードまでを一つのパスとし，各パスに対する If-Then ルールを導き，個々のルールに対する一般化の後，クラス毎のルール集合に対する一般化を行って，得られるものである．SE が difference

*4　紙面のスペースの関係上，事例 A や B が欠損値を持つ場合を考慮しないアルゴリズムを表記したが，実際のアルゴリズムでは欠損値を持つ場合にも対応している．

*5　ここで用いるデータにクラス情報が含まれることは暗黙の前提である．

表 4 データセットの概要

データセット名	事例数	クラス数	属性数	データセット名	事例数	クラス数	属性数
Car Evaluation	1728	4	Nom.* 6	Image	2310	7	Num.** 19
Tic-Tac-Toe	958	2	Nom. 9	Page Blocks	5473	5	Num. 10
Nursery	12960	5	Nom. 8	Iris Plant	150	3	Num. 4
Mushrooms	8124	2	Nom. 22	Pen-Digits	10992	10	Num. 16
Monk1	124(432)†	2	Nom. 6	Yeast	1484	10	Num. 8
Monk2	169(432)†	2	Nom. 6	German Credits	1000	2	Mixed.*** 13/7
Monk3	122(432)†	2	Nom. 6	Pima Indians	768	2	Num. 6
Krvkp	3196	2	Nom. 36	Cmc	1473	3	Mixed. 7/2
Titanic	2201	2	Nom. 3	Shuttle	14500	7	Num. 9
Voting Records	435	2	Nom. 16	Ann-thyroid	7200	3	Mixed. 15/6
Breast Cancer	699	2	Nom. 9				

* 離散値属性，** 連続値属性，*** 離散値属性と連続値属性が混在：離散値属性の数/連続値属性の数，† このデータベースは予め訓練事例群とテスト事例群が分けられている：訓練事例群の数 (テスト事例群の数)

list から選ぶ条件は，RDR システムにより誤判断された事例を正しく判断している SE の If-Then ルールの条件部と，difference list の共通部分とする．C4.5 による If-Then ルール集合では否定の表現（not）が扱われていないので，SE が否定の条件も difference list から選べるように属性値のバイナリー化を実施し，ルール集合に否定の条件が現れるようにした．その結果，バイナリー化を行わないで得られる If-Then ルール集合とは異なるルール集合が得られるが，未知事例に対する予測精度は同程度であることを確認している．

6·1 実験 1：従来の RDR 法における MDL 原理

データセットからランダムに選んだ 75%を，知識ベースを構築するための訓練事例群，残りを知識ベースの精度を調べるためのテスト事例群とする．そしてこの訓練事例群を使って SE を構築した（専門家もテスト事例は経験していない）．また同じ事例群を全体の DL を計算するためにも用いる*6．SE と訓練事例群を用いて，従来の RDR 法で二分木を構築していく．そして事例が入力されて，新しいノードが追加されるたびに，その時点での二分木により誤判断されるテスト事例の数と全体の DL に基づく点をプロットしていく．構築される二分木の性能は入力事例の順番によって異なるので，実際には同じ試行を入力事例の順番をランダムに変えて 10 回行い，全ての点を同じグラフにプロットした．

代表的な結果を図 5 に示す．これはデータベース “Car Evaluation” に対して得られた結果である．横軸が全体の DL の値，縦軸がテスト事例群に対する誤事例数を表している．この図から，専門家が例外ルールを追加するにつれ，全体の DL が平均的には小さくなり，巨視的に見て未知事例に対する誤判断事例数が少なくなっているのがわかる．この傾向はその他の多くのデータベースにおいても見られ，専門家の判断は合理的であり MDL 原理に沿っていると判断できる．

*6 我々は本論文で，木が逐次的に成長する間でも，75%の事例群は利用できると仮定している．

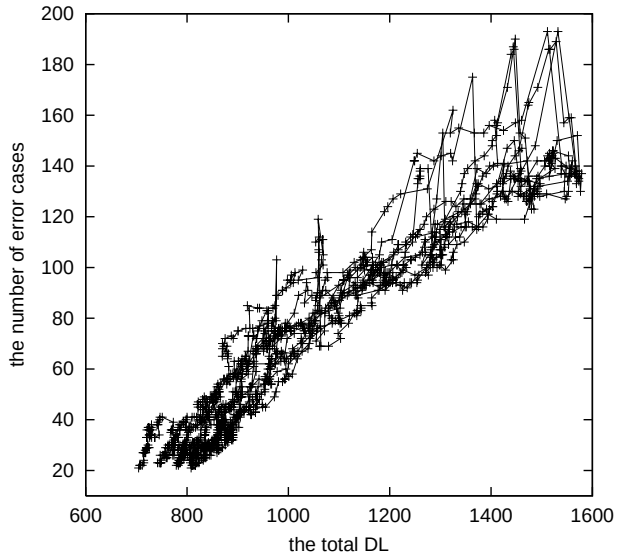

図 5 データベース “Car Evaluation” においてデフォルトクラス “unacceptable” の場合の結果

6·2 実験 2：データと専門家を用いる場合の効果（従来の RDR 法との比較）

6·1 節と同じく，データベースを訓練事例群（75%）とテスト事例群（25%）に分けるが，SE は訓練事例群の 2/3 だけを用いて構築した．対象問題領域での経験が少し不足している SE を仮想的に作り出し，その専門家からの知識獲得だけでなく，データからも知識獲得を行うことで，予測精度の高い二分木を構築できるかどうかを調べるためである．二分木を構築するため（そして全体の DL を計算するため）に用いるデータは訓練事例群の 1/3，2/3 そして 3/3 と仮定した．

SE とそれぞれのデータセット（1/3,2/3,3/3）を用いて，5·2 節の提案手法により 3 種類の二分木を構築し，テスト事例に対するエラー率を調べる．それぞれの結果は同じ SE，同じデータを用いて構築した従来の RDR 法による二分木の結果と比較する．つまり例えば，1/3 のデータを用いて提案手法により構築された二分木は，同

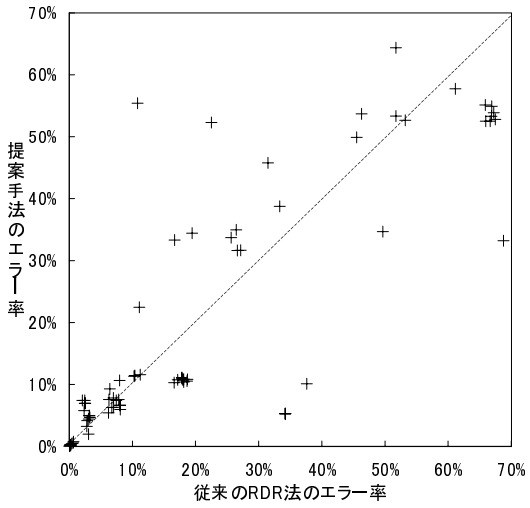

図 6 提案手法と従来の RDR 法の比較（25%データ）

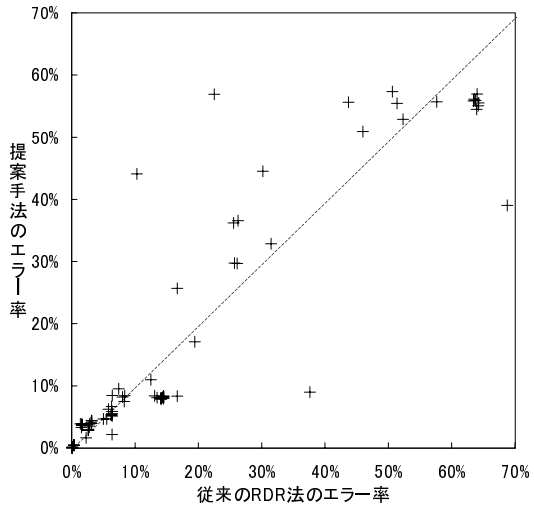

図 7 提案手法と従来の RDR 法の比較（50%データ）

じ 1/3 のデータを用いて構築された従来手法の二分木と比較する．ただし，RDR 法の性能はデフォールトクラスとデータの順番に依存するので，各データベースにおける各デフォルトクラス設定に対して，入力する事例の順番をランダムに変えて異なる二分木を 10 個構築した．これら 10 回の試行の平均エラー率を結果とし，全 21 データベースに対して，全部で 79 点をグラフにプロットする．

表 5 はサイズが異なる 3 種類のデータに対して，提案手法が従来手法よりもエラー率に関して同じであるか，下回る場合を要約したものである．どのデータサイズに対しても平均的には互角の性能が得られている．図 6，図 7 そして図 8 ではそれぞれ，データサイズが 1/3，2/3 そして 3/3 の場合に得られる結果をプロットしたものである．

どの図を見ても，提案手法により全体の記述長がなるべく小さくなるように構築したにも関わらず，従来手法

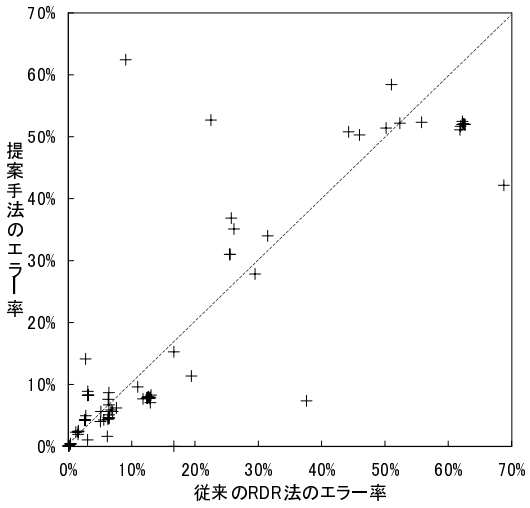

図 8 提案手法と従来の RDR 法の比較（75%データ）

表 5 提案手法と従来手法の比較

データのサイズ	79 点中の割合
75%の 1/3	38 点
75%の 2/3	40 点
75%の 3/3	40 点

と比べて非常に高いエラー率を持つ二分木が構築される場合もあるし，逆に従来法よりも非常に小さいエラー率を持つ二分木が構築される場合もあることがわかる．従来法でもエラー率が 50%を越えるものがあり，SE がテスト事例を経験していないことも影響しているものと思われる．

従来法に比べて誤差が大きな二分木を分析した結果，誤差が 50%を越えるノードが多く存在することがわかった．誤事例に対する DL の計算は誤差が 50%以下であることを前提としたものであるので，訓練事例に対して二分木の各ノードで誤差が 50%以下になる範囲で最小の DL を探索するような制約を課せば，この問題は回避できる．RDR 法の原理に従えば，十分なデータが与えられれば，このような事態は最終的には修復されるので，訓練データ数が小さく，その修復（例外ノードを追加すること）が完了する以前に構築を終了してしまったことも一因である．

表 5 の結果，それぞれ割合の異なる 3 種類のデータに対する結果が変わらないということは，提案手法はデータの量に依存せず，従来手法と同等の頑強性を持つということを示している．以上の結果，SE が経験不足でもデータを用いた MDL 原理に基づく提案手法により，従来手法と同程度の予測精度を持つ知識ベースを構築できることを実験的に確認できた．

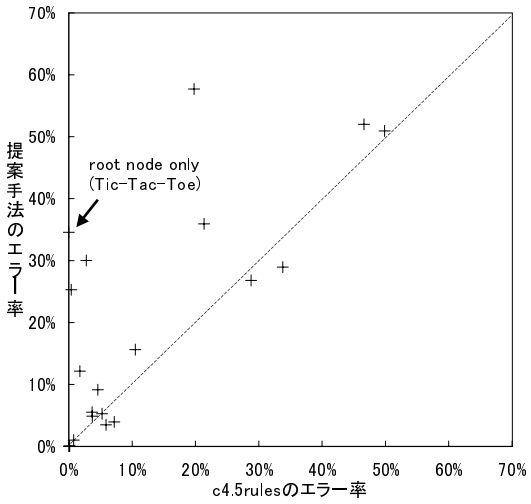

図 9 提案手法と c4.5rules の比較

6·3 実験 3：データだけを用いる場合の効果（C4.5 との比較）

この章では 6·2 節と異なり，SE に頼らずにデータだけから予測精度の高い知識ベースを帰納的に構築することを考える．75%の訓練事例群のみを用い，5·1 節の提案手法により二分木を構築する．

ただし入力事例の順番，デフォルトクラスの設定により異なる二分木が構築されるので，それぞれのパラメータを変更し，全部で (クラス数 $\times 10$) 個の知識ベースを独立に構築する．そしてその中で最も全体の DL が小さい二分木を選び，提案手法によって構築される最も予測精度が高いと考えられる知識ベースとする．

選ばれた二分木のエラー率を C4.5 によって得られるインダクションルールセット（以下，“c4.5rules” と呼ぶ）のエラー率と比較する．図 9（横軸：c4.5rules のエラー率，縦軸：選ばれた二分木のエラー率）に 21 データセットの結果を示す．

c4.5rules と同じまたは提案手法の方が低いエラー率の知識ベースを構築できたのは，21 点中 8 点である．提案手法のエラー率が c4.5rules のものと比べて極端に大きいデータセットも数点有ある．そのようなデータセットに対する二分木には，6·2 節の場合と同様に，ノードの帰結部とは別のクラスを持つ事例群を多く持つノードが多数あった．

また “only root node” と記載した点のデータセットでは知識獲得過程においてノードが一つも追加されず，ルートノードだけの知識ベースであった．ただし，これは探索開始点となる条件の選び方（5·1 節参照）並びに欲張り探索の方法に原因があり，探索が完全であれば，c4.5rules と同程度の二分木が構築できる可能性はある（6·2 節では SE が選ぶ条件から探索を開始した場合，問題なく新しいノードが追加されている．）．

21 点中 8 点という結果ではあるが，図 9 により c4.5rules と同程度の二分木が提案手法により構築できる場合も多く，RDR 法は逐次的に知識獲得を行うのに対して，C4.5 はバッチ処理であることを考慮すると，この結果は妥当であると考える．

7. お わ り に

本論文では，帰納学習（データからの知識抽出）の RDR 法（人間の専門家からの知識獲得）への統合化の可能性を探り，基本原理として MDL 原理を用いることを提案した．

実験により，従来の RDR 法による SE からの知識獲得過程において，MDL 原理が成り立っていることを確認した．また，専門家があまり経験豊かでなくても，従来手法と同程度の頑強さを持つ知識ベースが提案手法により構築できること，標準的な機械学習法と同程度の予測精度を持つ知識ベースが構築できることを示した．

今後の課題として以下の 3 点に取り組む．

1）本論文では MDL 原理を全面的に信用し，RDR システムが誤判断したときに，DL が小さくならなければ，例外ノードを追加しない方針を採用した．RDR 法の誤事例が見つかるたびに二分木を修正するという利点を損ねた面もある．MDL 原理の適用範囲を，新しい条件部の選択のみに限定した場合の性能を検討する．

2）本論文で提案した手法は離散値属性しか扱えないので，連続値属性も扱えるように拡張する．従来の RDR 法では，RDR システムによって誤判断された事例（属性 A_1 の値 α）と誤った結論を導き出したノードの持つ cornerstone case（属性 A_1 の値 β）を分ける条件として，「属性 A_1 の値が γ 以下ならば」という不等号の条件を専門家が提示することで連続値属性も扱える（$\alpha < \gamma < \beta$）．MDL 原理に基づく知識獲得では，α と β の間に存在する閾値の候補を新しいルールの条件として採用した場合の全体の DL を計算すれば，連続値属性も扱える．

3）全体の記述長を計算する際に用いる重みを経験的に 0.3 と設定しているが，MDL 原理の概念を考えると，真に適切な符号化方法を採用すれば，この重みは必要ないはずである．C4.5 において，構築された決定木から If-Then ルール集合を導く際に用いられる符号化方法を参考に，本論文では二分木によって各ノードに分割される事例集合毎に誤事例に対する DL を計算した．符号化の方法と重みの関係をより深く検討し，重みに鈍感な（あるいは重みを不要とする）符号化方法を実現する．

本論文により，1）人間の持っている知識と蓄積されたデータの異なる 2 種類の知識源を併用した知識獲得，2）知識ベースシステム開発の任意の時点における専門家からの知識獲得とデータからの帰納学習の切り替えが可能である柔軟なシステム開発の可能性が開けた．

謝　辞

本研究の一部は文部省科学研究費基礎研究 B（国 11694159，12480088）の補助による．最後に，有益なご指摘を賜りました査読者の方々に深く謝意を表します．

◇ 参 考 文 献 ◇

[Blake 98] Blake, C. and Merz, C.: UCI Repository of machine learning databases (1998), http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[Compton 89] Compton, P., Horn, K., Quinlan, J. R., and Lazarus, L.: *Maintaining an Expert System*, pp. 366–385, Addison Wesley (1989).

[Compton 95] Compton, P., Preston, P., and Kang, B.: The Use of Simulated Experts in Evaluating Knowledge Acquisition, in *Proc. of the 9th Knowledge Acquisition for Knowledge Based Systems Workshop* (1995).

[DEL 98] Data for Evaluating Learning Valid Experiments, The University of Toronto (1998), http://www.cs.utoronto.ca/∼delve/data/datasets.html.

[Freund 99] Freund, Y. and Schapire, R.: ブースティング入門, 人工知能学会誌, Vol. 14, No. 5, pp. 771–780 (1999).

[Gaines 89] Gaines, B.: An Ounce of Knowledge is Worth a Ton of Data: Quantiative Studies of the Trade-Off between Expertise and Data based on Statistically Well-Founded Empirical Induction, in *Proc. of the 6th International Workshop on Machine Learning*, pp. 156–159, San Mateo, California (1989), Morgan Kaufmann.

[Gaines 91] Gaines, B.: *The Trade-Off Between Knowledge and Data In Knowledge Acquisition*, chapter 29, pp. 491–505, MIT Press, Cambridge, Mass (1991).

[Gaines 92] Gaines, B. and Compton, P.: Induction of Ripple-Down Rules, in *Proc. of the 5th Australian Joint Conference on Artificial Intelligence*, pp. 349–354, Singapore (1992), World Scientific.

[Gaines 95] Gaines, B. and Compton, P.: Induction of Ripple-Down Rules Applied to Modeling Large Databases, *Journal of Intelligent Information Systems*, Vol. 5, No. 2, pp. 211–228 (1995).

[Gary 93] Gary, D. and Trevor, J.: Optimal Network Construction by Minimum Description Length, *Neural Computation*, pp. 210–212 (1993).

[Morik 93] Morik, K., Wrobel, S., Kietz, J., and Emde, W. eds.: *Knowledge Acquisition and Machine Learning: Theory, Methods, and Applications*, Academic Press (1993).

[Quinlan 89] Quinlan, J. and Rivest, R.: Inferring Decision Trees Using the Minimum Description Length Principle, *Information and Computation*, pp. 227–248 (1989).

[Quinlan 93] Quinlan, J. ed.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993).

[Rissanen 78] Rissanen, J.: Modeling by Shortest Data Description, *Automatica*, pp. 465–471 (1978).

[Suzuki 93] Suzuki, J.: A Construction of Bayesian Networks from Databases on an MDL Principle, in *Proc. of the 9th Conference on Uncertainty in Artificial Intelligence* (1993).

[和田 00] 和田, 堀内, 元田, 鷲尾：Ripple Down Rules 法における知識獲得の特性評価に基づくデフォルト知識の決定規範, 人工知能学会誌, Vol. 15, No. 1, pp. 177–186 (2000).

[Wrobel 94] Wrobel, S. ed.: *Concept Formation and Knowledge Revision*, Kluwer Academic Publishers (1994).

〔担当委員：新田克巳〕

2000 年 10 月 6 日　受理

著 者 紹 介

和田　卓也（学生会員）

1998 年大阪大学工学部通信工学科卒業．1999 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．人間からの知識獲得とその利用法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．