

Ripple Down Rules 法における知識獲得と帰納学習の 統合化の試み

和田 卓也* , 堀内 匡 , 元田 浩 , 鷲尾 隆

大阪大学産業科学研究所

* wada@ar.sanken.osaka-u.ac.jp

1 概要

“知識獲得”という言葉は従来、知識ベースシステムの構築の際に専門家などの知識源から問題解決に必要な情報や知識を抽出・構造化することを意味しているが、ここでは知識獲得をより広い観点から捉えてみたい。すなわち、広義の“知識獲得”を大きく分けると、1)人間の専門家からインタビューなどにより知識を獲得する知識獲得(狭義)、2)事例データをもとに計算機プログラムにより知識を抽出する機械学習(帰納学習)、3)大規模データベースから埋もれている有用な知識を発掘する知識発見(KDD)などを考えることができる。

本研究ではまず、狭義の知識獲得手法として Ripple Down Rules 法 [Compton 92] (以下 RDR 法と呼ぶ)と呼ばれる興味深いアプローチに注目する。技術進歩の激しい近年において、得られた知識が不変的であると仮定することは非常に非現実的であり、一部の知識は急速にすたれ新しい知識の供給が必要となり、知識の獲得と保守は不可分になってきている状況の中で、RDR 法が知識獲得手法として有望であると考えられる。RDR 法では、知識獲得を既存の知識の洗練と捉え、知識の誤りや不足があれば専門家の助言をもとに例外知識を追加することにより継続的に知識ベースを維持・管理する。さらに、新しい知識獲得時には知識の整合性も維持することができ、専門家と計算機とのインタラクションの中で知識を獲得できる。

しかし専門家といえども神様ではないので、逐次的に知識を追加していく中で、適切な知識を常に提供できるわけではない。また現実には(特に企業などにおいては)大量データが既にデータベース上に蓄積されており、そのような蓄積データを積極的に活用することが望ましい。

そこで本研究では、狭義の知識獲得手法である RDR 法に対して、事例データから計算機プログラムにより知識を抽出する帰納学習の機能を取り入れた統合化手法の可能性を探る。我々はすでに蓄積データから求められる記述長: Description Length (以下 DL と呼ぶ)と呼ばれる評価基準をもとに RDR 法におけるデフォルト知識を決定する規範を提案している [和田 2000]。本研究ではその発展として、最小記述長原理: Minimum Description Length 原理 [山西 92] (以下 MDL 原理と呼ぶ)を用いることで、統合化を図ることを提案する。

本研究における提案手法により、蓄積されているデータから帰納学習により知識ベースを構築し、それ以降は新たな知識の追加が必要になれば人間の専門家から知識を獲得し、知識ベースを更新するというような柔軟なシステム構築が可能になると考えられる。

2 RDR 法のアルゴリズム

RDR 法は知識エンジニアによるインタビューや事前分析を必要とせず、問題領域の専門家から直接知識を獲得することで、今まで知識ベースシステムの開発で問題になっていた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．知識ベースシステム開発において，専門家はある結論に至るために必要十分な条件をすべて示すことは苦手であるが，個別の事例に対する結論の良否を判断することは容易である [Compton 92]，ことがわかっている．RDR法の基本原理はこのような知見に基づいており，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．具体的には，ある事例が誤判断された場合，知識ベースの整合性を維持するためにその事例と知識ベースの誤りを正すために専門家が持っている知識を，同時に知識ベースに追加する．

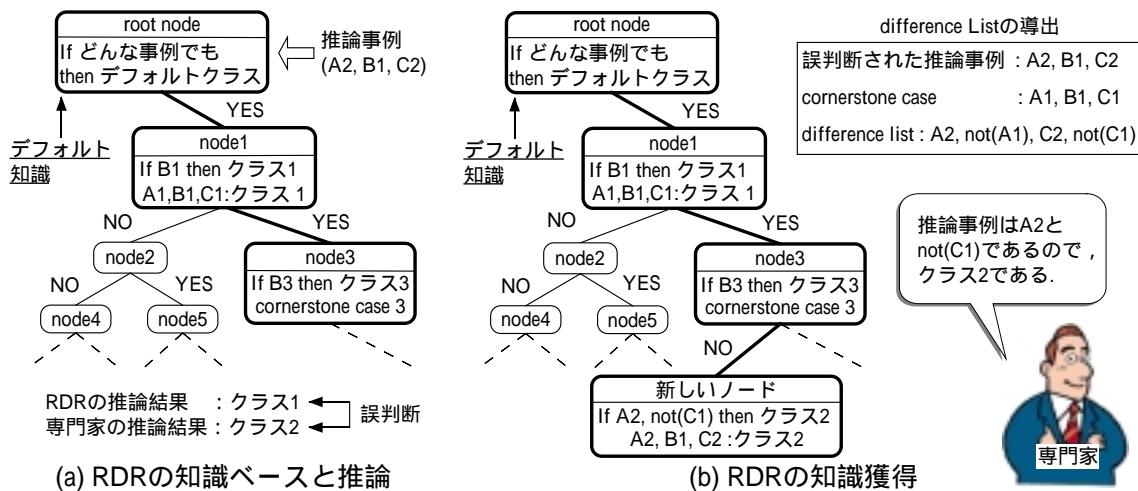


図 1: RDR 法の知識構造と知識獲得アルゴリズム

RDR法の知識ベースは図1.(a)の例のように二分木により実現することができる．各ノードはルール (If-Thenルール)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 branchとNO branchを持つ．ただしルートノードは特別であり，デフォルト知識と呼ばれる条件部が空集合のルールしか持たない．知識ベースを構築する前にこのルールの帰結部のクラス (デフォルトクラス)を設定する必要がある．

ある事例が与えられて，その事例に対する結論を推論することを考える．RDR法の推論過程は二分木のルートノードから始まる．もしその事例がノードの持つルールの条件部を満足すれば，YES branchのノードに，条件部を満足しない場合は，NO branchのノードに，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ノード (last satisfied node)の帰結部とする．例では推論事例に対して node1の条件部が最後に満足されたので，RDR法の推論結果はクラス1となっている．

もし事例に対する推論結果が間違っていた場合，知識ベースの誤りを訂正するため，専門家から知識獲得を行なう必要がある．例では専門家の推論結果がクラス2であり，知識獲得が必要である．以下にRDR法の知識獲得のステップを述べる．

- Step 1: 専門家が誤判断された事例に対する正しい結論を与える (図1.(b)ではクラス2である)．
- Step 2: RDRシステムは誤った結論を与えたノード (最後に条件部が満たされたノード)が持っている 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 (difference listは {A2,not(A1),C2,not(C1)}である)．
- Step 3: 専門家はこのリストから誤判断された事例に対して正しい結論を正当化するための条件を選ぶ (専門家は A2と not(C1)を選んだ)．
- Step 4: RDRシステムは，専門家が選んだ条件と新しい結論を，新たに追加するノードの 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 (新しいルールは If A2,not(C) then クラス2となり，cornerstone caseは A2,B1,C2:クラス2である)．

Step 5: RDRシステムはそのノードを知識ベースである二分木に追加する。追加する位置は、推論パスの一番最後のノード (end node) が一番最後に条件部が満足されたノードであれば end node の YES branch の下、そうでなければ end node の NO branch の下とする (node3 の NO branch の下に追加されている。新しい知識ベースでは事例 A2, B1, C2 の推論結果はクラス 2 になるが、事例 A1, B1, C1 の推論結果は今まで通りクラス 1 のままである)。

以上のように 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ノードの例外ノードとして end node の下に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合と同じ文脈で用いられることになる。

3 MDL 原理の説明と DL の計算方法

表 1: トレーニングデータの例

番号	属性			クラス
	Swim	Breath	Legs	
1	can	lung	2	Dog
2	can	lung	4	Penguin
3	can	skin	2	Monkey
4	can	skin	4	Dog
5	can_not	lung	2	Dog
6	can_not	lung	4	Monkey
7	can_not	gill	2	Penguin
8	can_not	gill	4	Dog
9	can_not	skin	2	Dog
10	can_not	skin	4	Monkey

送信者 A と受信者 B がそれぞれ表 1 の事例集合のコピーを持っている状況を想定する。ただし、B のコピーのクラス欄は空白であり、A が通信路を通してこのクラス欄の情報をなるべく少ない DL を使って B に教えたいとする。最も簡単な方法はクラス情報を直接送ることである。しかし、ある属性とクラスとの間に強い関係があることがわかった場合、その依存関係を相手に教えてやりさえすれば、クラスの欄の一つ一つがどうなっているかを伝えるのに必要な DL を劇的に少なくできる。そのために次の方法をとる。

- Step 1: 事例集合をその属性値に従ってある分割方法 (知識モデル) に基づき、部分集合に分割する。
- Step 2: この知識モデルの記述を符号化して送る。
- Step 3: 各集合に対応する代表クラスの記述を符号化して送る。
- 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、正しいクラスの記述を符号化して送る。

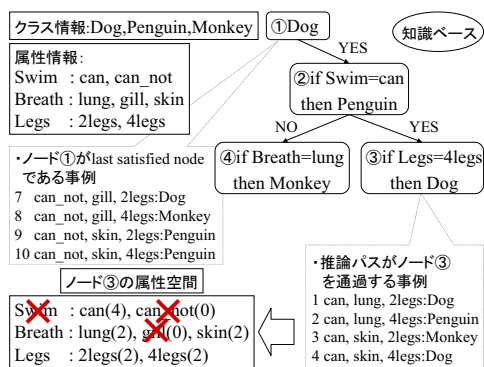


図 2: DL の計算方法

Step 1 における知識モデルの情報を符号化した後の DL と、Step 4 において例外事例のクラスの情報を符号化した後の DL はトレードオフの関係にあるが、全体の DL を最少にするような符号化を定めてやることで、十分に精密だが細かすぎない最適な知識モデルを得ることができる。そして「そのような符号化の下での全体の DL が小さい知識モデルは未知事例に対する予測精度が高い」という原理が MDL 原理である。次にこの MDL 原理に沿って、図 2 で与えられた RDR 法の知識ベースと表 1 の事例群に対する DL の具体的な計算の一部を我々の考えた符号化方法とともに説明する。ただし以下に示す符号化が MDL 原理に沿う唯一のものであるわけではない。

3.1 知識ベースの DL の計算

ここでは二分木のノード ③ の情報を一意に認定するのに必要な DL を計算する。表 1 の全事例をこの知識ベースを用いて推論した場合、その推論パスがノード ③ を通過する事例は事例 {1, 2, 3, 4} の 4 個であり、これらの事例による属性空間は {属性 Breath: lung, skin, 属性 Legs: 2legs, 4legs} (図 2 の左下部) となる。ノード ③ において符号化すべき情報は「If Legs=4legs then Dog」と「Yes branch は無く、No branch も無い」である。

まず前者の情報を {条件部に用いる属性の数: 1}, {その属性: Legs}, {その値: 4Legs}, {帰結部: Dog} という四つの情報に分ける。図左下部の属性空間により条件部に用いる可能性のある属性数の候補は {1, 2} の二

つであるので、 $\log_2 2C_1$ ビットの信号で { 条件部に用いる属性の数:1 } という情報を A から B に送ることができる。また条件部に使用される属性の候補は {Breath,Legs} であるので、情報 { その属性:Legs } を送るのに $\log_2 2C_1$ ビット必要である。同様の考え方で、情報 { その値:4Legs } を送るのに $\log_2 2C_1$ ビット必要である。情報 { 帰結部:Dog } は、帰結部の候補が {Dog,Monkey} であるので、 $\log_2 2C_1$ ビット必要である。

後者の情報「Yes branchは無く、No branchも無い」については、ノード③は {Yes(無)No(無),Yes(無)No(有),Yes(有)No(無),Yes(有)No(有)} という四つのいずれかの状態をとるので、 $\log_2 4C_1$ ビット必要である。

結局ノード③の情報を送るのに必要なDLは全部で $4\log_2 2C_1 + \log_2 4C_1$ ビットとなる。紙面のスペースの関係上、その他のノードにおけるDLの計算は省略するが、ここで説明したような符号化をAの側でルートノードから深さ優先または幅優先で行い、符号化信号をBに送れば、Bは受け取った信号を復号化して図2の知識ベースを知ることができる。

3.2 誤事例のDLの計算

ここでは推論過程で二分木のノード①が last satisfied node である事例の集合 {7,8,9,10} において、ノード①の帰結部Dogと異なるクラスを持つ誤事例 {8,9,10} に対する正しいクラス情報を一意に認定するのに必要なDLを計算する。まず情報「この四つの事例の中に帰結部とは別のクラスが2個ある」を送る必要がある。この情報に対する候補は {0個,1個,2個} が考えられるので、 $\log_2 3C_1$ ビットの信号でこの情報を送ることができる。そして情報「事例数が一番目に多いクラス Penguin の情報を送る」を伝えるのに、候補として {Penguin,Monkey} があるので $\log_2 2C_1$ ビット必要である。そして情報「4個中2個の事例がクラスPenguinを持つ」を送るのに、クラスPenguinの事例数は一番多く、候補として {4個中2個,4個中3個} が考えられるので $\log_2 2C_1$ ビット必要である。次に情報「クラスPenguinを持つ2個の事例は {7,8,9,10} のうち {9,10} である」を送るのに、{{9,10},{8,10},{8,9},{7,10},{7,9},{7,8}} の6個の候補が考えられるので $\log_2 6C_2$ ビット必要である。次に情報「事例 {7,8} のうち8がクラスMonkeyである」を送るのに、クラスPenguinの場合と同じ方法で符号化すると、 $\log_2 2C_1 + 2\log_2 2C_1$ ビット必要である。

全体として $\log_2 3C_1 + 4\log_2 2C_1 + \log_2 6C_2 + \log_2 2C_1$ ビット分の情報を受け取ることで、Bは帰結部がDogであるノード①に分類される4個の事例のうち「事例9と事例10はクラスPenguin、事例8はクラスMonkeyである」ことを知る。このような誤事例の情報をルートノードから順に送ることで、Bは各事例集合において帰結部と異なるクラスを持つ事例の正しいクラスを知る。

結局、3.1節と3.2節で触れたような符号化で得られる全部の信号をAが送ることで、Bは手元に持っている表のクラス欄の空白を正しく埋めることができる。MDL原理とは、このようにしてAからBに送られる可能性のある知識モデルの中で、送られる全信号のDLが最少な知識モデルが未知事例に対する予測精度が高い知識モデルであるという原理である。ただし、ここで説明したものを含めて符号化の手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。従って一般的にMDL原理に基づく知識モデルの構築においては(全体のDL) = (誤事例のクラス情報のDL) + W × (知識モデルのDL) という式を用いる。ここでWは1以下の係数であり、本研究では経験則によりW = 0.1としている。

4 MDL原理による新しい条件の探索

3章では、MDL原理と与えられた知識ベースと事例集合に対するDLの計算方法について説明した。この章では、MDL原理に基づく新しい知識獲得の方法論を図3を例に説明する。図の上部の表は、事例 {a2,b1,c2} に対する推論を行い、RDR法の結論は事例 {a1,b1,c1} を cornerstone case として持つノードの帰結部であるクラスNであるが、正しい結論はクラスPであることを示している。従って、既存の知識ベースに帰結部

がクラスPで cornerstone caseが $\{a2, b1, c2\}$ である新しいノードを追加する必要がある。今までの RDR 法では、専門家が difference list $\{a2, \text{not}(a1), c2, \text{not}(c1)\}$ から $a2$ と $\text{not}(c1)$ を選んだ場合、その条件をそのまま新しいノードの条件部として採用していた。しかし我々は3章で説明した MDL 原理に基づき、difference list から選ばれうる条件のうち、仮に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。従って新しい条件として専門家が選んだ $\{a2, \text{not}(c1)\}$ を探索開始点とする。この条件を仮採用した場合の全体の DL は 120 ビットである。次に $\{a2, \text{not}(c1)\}$ の一つ違いの条件 $\{a2\}$ および条件 $\{\text{not}(c1)\}$ を仮採用した時の全 DL は、それぞれ 75 ビット、100 ビットである。 $\{a2\}$ を仮採用した場合、 $\{a2, \text{not}(c1)\}$

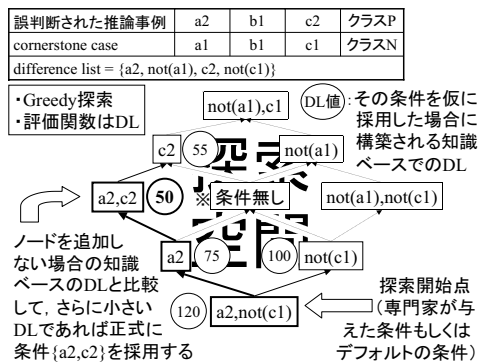


図 3: MDL 原理による新しい条件の探索

および $\{\text{not}(c1)\}$ よりも全 DL が小さいので探索点を $\{a2\}$ に移す。探索点が $\{a2, \text{not}(c1)\}$ の場合と同様に、 $\{a2\}$ の一つ違いの条件 $\{a2, c2\}$ による全 DL を計算する。この過程を繰り返し、条件 $\{c2\}$ による全 DL が条件 $\{a2, c2\}$ よりも大きいので、条件 $\{a2, c2\}$ の点で探索を終了する。最後に新しいノードを追加しない場合の全 DL よりも、条件 $\{a2, c2\}$ による全 DL の方が小さい場合、条件部 $\{a2, c2\}$ 、帰結部クラス P、cornerstone case $\{a2, b1, c2\}$ を持つノードを正式に知識ベースに追加する。

あらかじめ大量データが存在する場合、この方法により専門家の選んだ不完全な条件を探索開始点とし、その条件

にある程度近く、DL が少なくとも極小になる条件を探索することができる。また、difference list における推論事例から導かれる条件 (例では $\{a2, c2\}$) をデフォルトの探索開始点とすれば、専門家から知識獲得をしなくても、大量データから帰納的に RDR 法の知識ベースを構築することもできる。

5 実験

この章では、5.2 節で RDR 法の知識ベースに対して MDL 原理が成り立つのかどうか実験的に確認する。次に 4 章で説明した探索機能を組み込んだ RDR 法 (以下探索 RDR 法と呼ぶ) を用いて C4.5 による計算機上の代替専門家から獲得される知識のデータによる補完機能の可能性を 5.3 節で検証する。また 5.4 節では、データからの知識獲得である一般的な機械学習手法 C4.5 と同程度の予測精度を持つ知識ベースを探索 RDR 法を用いて帰納的に構築できるのかも検証する。

5.1 データセット

California 大学 Irvine 校の機械学習データライブラリから 24 個のデータセットを、Toronto 大学の機械学習評価用データライブラリから 1 個のデータセットを選んだ。うち 13 個は全属性が離散値、11 個は全属性が連続値、そして残り 1 個は離散値属性と連続値属性の両方を含んでいる。ただし探索 RDR 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。またデータセットの事例群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。

5.2 実験 0

訓練事例群と機械学習法 C4.5 を用いて構築した決定木からインダクションルールセットを導き、これを人間の専門家の代わりである代替専門家の知識源とした。そして代替専門家と訓練事例群を用いて、探索 RDR 法の知識ベースを構築する過程で、新しいノードが追加される度に、全体の DL とテスト事例群に対する誤事例数による点をグラフにプロットした。ただし RDR 法の性能は訓練事例群の推論を行う順番に

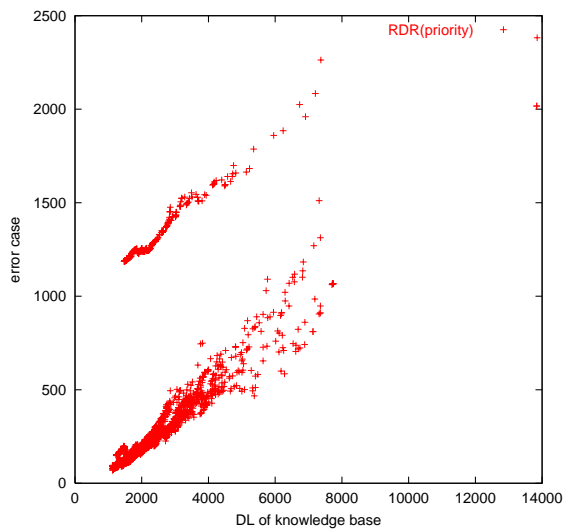


図 4: 知識獲得過程での全 DL とテスト事例群に対する誤事例数

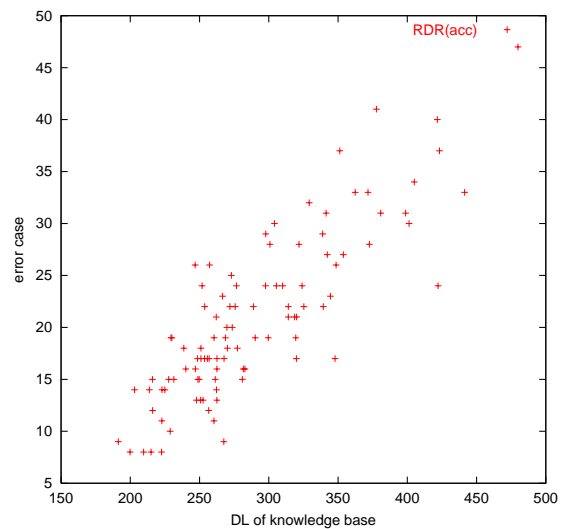


図 5: 知識獲得最終段階の全 DL とテスト事例に対する誤事例数

よって異なるので、その順番をランダムに変えて 10 回の試行分の結果をプロットした。また、探索 RDR 法による知識ベースを全体の DL が小さくなり続けるまで構築し、最終的に構築できる知識ベースに対する全 DL とテスト事例群に対する誤事例数による点も別のグラフにプロットした。訓練事例群の推論に対する順番を毎回ランダムに変えて、この試行を 100 回行い、全部で 100 点をプロットした。

5.2.1 結果と考察

代表的な結果を図 4 と 5 に挙げる。図 4 は nursery という名のデータセットにおける結果である。この図より、ノードが追加されて全体の DL が減少する程、テスト事例群に対する誤事例数が減少することがわかる。このような傾向の結果はその他のデータセットに対しても見られた。ただしこの図では、全体の DL の減少で誤事例数が減少するという傾向は同じであるが、その誤事例数の絶対値がその他の試行の場合と比べて非常に高い試行が 1 回ある。例えばあるノードの帰結部がクラス A で、このノードに分類される 4 個の事例群の持つクラス分布が { クラス A:3, クラス B:1 } である場合と { クラス A:1, クラス B:3 } である場合、3.2 節で説明した誤事例のクラス情報の符号化方法に基づけば、このノードにおける誤事例のクラス情報の符号化後の DL は同じになる。つまり 3.2 節の符号化方法では、あまりに誤事例が多い場合でも逆に DL の値が小さくなる可能性がある。誤事例数の絶対値が結果的に非常に大きくなる知識ベースを調べ、分類される訓練事例群のうち帰結部と異なるクラスを持つ事例群が極端に多いノードが知識ベースに多数存在することを確認した。

この問題の解決のため、探索 RDR 法で最終的に構築される知識ベースに対し、各ノードに分類される訓練事例群のクラス情報から、一番事例の多いクラスをそのノードの帰結部に変更するという処理を行い、テスト事例に対する誤事例数が格段に改善されることを確認した。従って以後紹介する探索 RDR 法のテスト事例に対する誤事例数およびエラー率の結果は、全てノードの帰結部変更処理を行った後の結果である。

図 5 はデータセット名 car に対して、全体の DL が減少するまで知識獲得を行って得られた知識ベースに対する全体の DL とテスト事例に対する誤事例数を調べた結果である。この図からも、MDL 原理が RDR 法の知識ベースにある程度有効であることを確認できる。例えば与えられた訓練事例群から独立に複数の知識ベースを構築した場合、複数の知識ベースの中で全体の DL が最小であるものが、それらの中で一番高い予測精度を持つものであると期待できる。ただし図 4 の傾向ほど明瞭には図 5 の傾向を他のデータセットで確認できなかった。MDL 原理と言えども、未知事例に対する予測には限界があるのであろう。

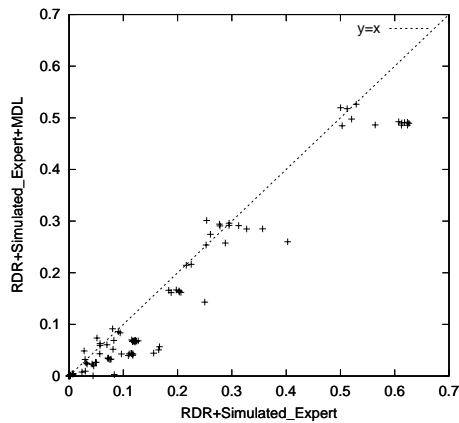


図 6: 探索 RDR 法と従来の RDR 法 (訓練事例群の 2/3 の代替専門家)

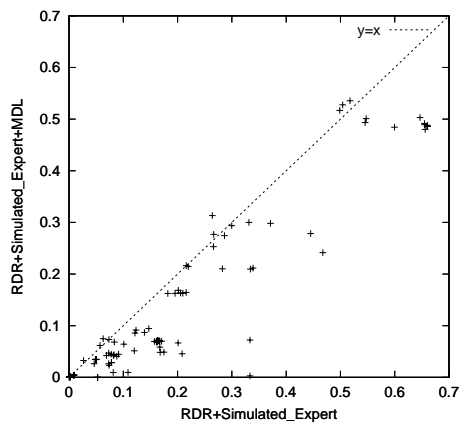


図 7: 探索 RDR 法と従来の RDR 法 (訓練事例群の 1/3 の代替専門家)

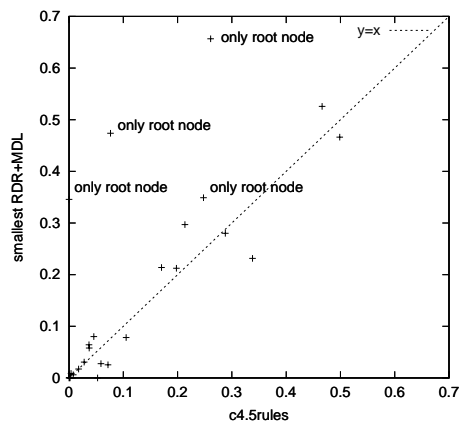


図 8: 探索 RDR 法とインダクションルールセット (デフォルトの探索開始点)

この知識ベースの予測精度の程度を知るため、同じ訓練事例群から導かれる C4.5 のインダクションルールセット (以下 c4.5rules と呼ぶ) のテスト事例に対するエラー率を x 軸に、選ばれた知識ベースのテスト事例に対するエラー率を y 軸に設定し、グラフにデータセット数だけの点をプロットした。

5.3 実験 1

訓練事例群の 2/3 および 1/3 のデータを用い、C4.5 のインダクションルールセットによる代替専門家を用意した。そしてそれぞれの代替専門家と訓練事例群を用い、探索 RDR 法の知識ベースを全体の DL が減少しつづけるまで構築した知識ベースのテスト事例群に対するエラー率と、同じ代替専門家と訓練事例群を用いて従来の RDR 法でノードが追加できるまで構築した知識ベースのテスト事例に対するエラー率を調べた。 x 軸を従来の RDR 法のエラー率、 y 軸を探索 RDR 法のエラー率として、各データセット、各デフォルトクラス毎に得られる結果を一つの図にプロットした。

5.3.1 結果と考察

結果を図 6 と 7 に示す。両方とも、25 個のデータセットにおいて、それぞれ前もって設定されるデフォルトクラスが異なる 94 種類の知識ベースに対する結果である。図 6 より、探索 RDR 法が従来の RDR 法と同じ、またはより高い予測精度を持つ知識ベースを構築できたのは 94 点中 79 点である。さらに図 7 では、そのような点が 94 点中 85 点もあった。

この結果より、頼みの専門家が経験不足でも大量のデータがあれば、その専門家から獲得した知識を MDL 原理に基づいて補完することで、専門家から直接知識を獲得するよりも予測精度が高い知識ベースを構築できる。

5.4 実験 2

5.2 節と 5.3 節では代替専門家と訓練事例群を用いて RDR 法の知識ベースを構築することを考えたが、ここでは訓練事例群から導かれる一番予測精度の高い知識ベースを自動的に構築することを考える。

訓練事例群だけを用意し、探索 RDR 法における条件の探索をデフォルトの探索開始点から行い、知識ベースを全体の DL が減少しつづけるまで構築した。RDR 法の性能は訓練事例に対する推論の順番やデフォルト知識に依存するので、各デフォルトクラスに対して、それぞれ独立に知識ベース構築の試行を 10 回行い、データセットのクラス数 \times 10 個の知識ベースを用意した。そして、その複数の知識ベースの中から、全体の DL が最小の知識ベースを選び、それを探索 RDR 法に基づき訓練事例群から導くことができた最も予測精度が高いと期待される知識ベースとした。

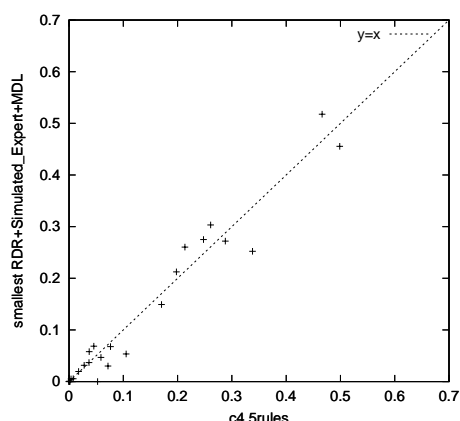


図 9: 探索 RDR 法とインダクションルールセット (代替専門家による探索開始点)

探索開始点を変えた図 9 では、どのデータセットにおいても c4.5rules と同程度の予測精度を持つ知識ベースを構築するので、図 8 の結果より即、探索 RDR 法を用いても c4.5rules と同程度の知識ベースを構築できないということにはならない。つまり 4 節で説明したものと別の探索方法を RDR 法に実装すれば、代替専門家が提示する条件に依らなくても、c4.5rules と同程度の知識ベースを帰納的に構築できる可能性はある。

6 むすび

本研究では専門家からの知識獲得手法である RDR 法に対して、データから知識を抽出する帰納学習の機能を取り入れた統合化手法の可能性を探った。そして MDL 原理に基づいて統合化を図ることを提案した。その結果、提案手法を組み込んだ RDR 法とデータを用いることで、代替専門家から獲得する知識を補完できることがわかった。また提案手法を用いても、同じデータを用いて構築できる C4.5 のインダクションルールと予測精度が同程度の RDR 法の知識ベースを帰納的に構築できないデータセットがあることもわかったが、追加実験により新しく追加するノードの条件部の探索方法を変えることで、そのような知識ベースが構築できる可能性も残っている。これらの結果から、完璧な神様ではない専門家からの知識獲得に対する大量データによる知識補完や、大量データからの一括知識獲得と人間からの逐次的な知識獲得の融合というような柔軟性を持つ知識ベースシステムの実現に一步近づいた。

ただし、更なる実験やプログラムの改良も必要であるが、特に訓練データのサイズによる提案手法の限界を示す必要がある。なぜなら訓練データが大量にある場合、提案手法を組み込んだ RDR 法は専門家から獲得される知識をより正しい方向に補完することが分かったが、訓練データが少数しかない場合は逆に獲得された知識を間違った方向に補完してしまう恐れがあるからである。

参考文献

- [Compton 92] Compton P., Edwards G., et al: Ripple Down Rules: Turning Knowledge Acquisition into Knowledge Maintenance. *Artificial Intelligence in Medicine*, Vol. 4, pp. 47-59, 1992.
- [山西 92] 山西 健司: MDL 入門: 計算論的学習理論の立場から. 人工知能学会誌, Vol. 7, No. 3, pp. 435-442, 1992.
- [和田 2000] 和田 卓也, 堀内 匡, 元田 浩, 鷲尾 隆: 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.

5.4.1 結果と考察

結果を図 8 に示す。c4.5rules と同じエラー率または探索 RDR 法の方が低いエラー率の知識ベースを構築できたのは、25 点 12 点であった。ただし探索 RDR 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が数点あった。特に“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。

しかし追加実験である、訓練事例群を用いて構築した代替専門家が提示する条件を探索開始点とする探索 RDR 法の結果 (図 9) によると、c4.5rules と同じエラー率またはより低いエラー率の知識ベースを構築できたのは 25 点中 15 点であり、極端に大きいエラー率を持つ知識ベースも構築されなかった。