

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

Decision Criterion of Default Knowledge Based on Characterization of Knowledge Acquisition in Ripple Down Rules Method

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.
takuya53@sanken.osaka-u.ac.jp

堀内 匡
Tadashi Horiuchi

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

元田 浩
Hiroshi Motoda

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

鷲尾 隆
Takashi Washio

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

Keywords: knowledge acquisition, ripple down rules method, default knowledge, minimum description length.

Summary

“Ripple Down Rules (RDR)” Method is one of the promising approaches to directly acquire and encode knowledge from human experts. It requires data to be supplied incrementally to the knowledge-base being constructed and new piece of knowledge is added as an exception to the existing knowledge-base. Because of this patching principle, the knowledge acquired strongly depends on what is given as the default knowledge, which is used as an implicit outcome when no inference outcome is obtained from the knowledge-base. Therefore, it is important to choose good default knowledge for constructing a knowledge-base which is accurate and compact. Further, many actual data are often noisy and we want the RDR noise resistant. This paper reports experimental results about the effect of the selection of default knowledge and the amount of noise in data on the performance of RDR using a simulated expert instead of a human expert. The best default knowledge is characterized as the class knowledge that maximizes the minimum description length to encode rules and misclassified cases. We confirmed by extensive experimentation that this criterion is indeed valid and useful in constructing an accurate and compact knowledge-base. We also confirmed that the same criterion holds when the data are noisy.

1. はじめに

知識ベースシステムの開発の前提として、人間の専門家が持っている知識は貴重であり、正しくかつ不変であり、時間とコストを費やしても専門家から知識を獲得することは非常に価値があると考えられてきた。しかし、技術革新の激しい今日では、既存の知識は急速にすたれ、頻繁に知識を更新する必要性が生じている。つまり、現実の世界においては専門家の知識は不変的なものではなく動的に変わりうるものと考えべきである。さらに、インターネットに代表される世界的ネットワークの進展に

より、コンピュータの使われ方が劇的に変わりつつある。数多くのユーザや専門家がネットワークを通して一つの知識ベースシステムにアクセスし、さらにそのネットワークを介して複数の専門家が継続的に新しい知識を補給することが日常的になる日も近い [Kang 97]。

このような状況のもとで、問題領域の継続的な変化に対応できる知識ベースシステムを構築する新しい手法が求められている。Ripple Down Rules法 [Compton 89, Compton 91] (以下 RDR と呼ぶ) は動的な問題環境に対応できる知識ベースシステムを構築するための有望な手法の一つであり、人間の専門家が持っている知識

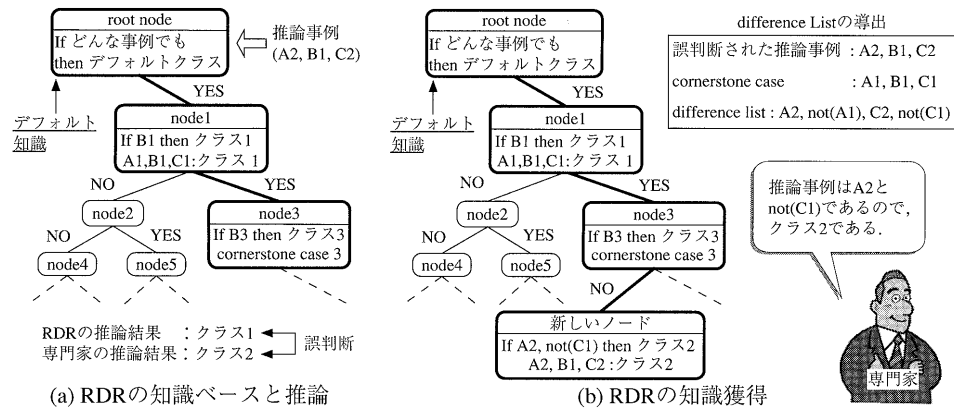


図 1 RDR の知識構造と知識獲得アルゴリズム

を直接的に獲得しルール形式への言語化を行う。RDR は知識獲得を行う際に、概念的な知識モデルを必要としない手法であると共に、逐次的な知識獲得が可能であり、さらに知識の保守・管理の点でも非常に効果的である。この手法は分類問題を対象としており、分類や診断の分野において良好な成果が得られている [Kang 96]。

RDR では人間の専門家から知識を獲得するので、多くの機械学習技術では不可避である大量のデータを必要とする統計情報を用いる必要のない点がこの手法の利点の一つとして挙げられる。しかし、データの統計情報を使うことを許せば、RDR の性能が向上する可能性がある。RDR では推論に使うルールから結論が出ないのためにデフォルト知識が準備されており、知識ベースの性能はデフォルト知識に強く依存する。どのような知識をデフォルトにおけばよいかは事前には自明ではないが、統計情報を使い特徴づけることは可能である。本論文では、デフォルト知識に関する特性を調べ、得られた知見をもとに性能の良い知識ベースを構築するためのデフォルト知識を決定する規範を提案する。さらに現実の問題環境のデータには多くの場合、ノイズが含まれているので、提案する規範がデータにノイズが含まれる場合にも適用できることを確認する。ただし「性能が良い」とは、正答率が高く（エラー率が小さく）サイズが小さいという意味である。

以下、2 章では RDR のアルゴリズムについて説明し、3 章では RDR におけるデフォルト知識の違いに着目する。4 章においてはデフォルト知識に関する特性を実験的に調べ、得られた知見をもとに性能の良い知識ベースを構築するためのデフォルト知識を決定する規範を提案する。最後に 5 章で結論を示す。

2. Ripple Down Rules 法のアルゴリズム

RDR 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しない。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ける長年の経験から [Compton 89]、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 90, Compton 92] ことがわかっている。すなわち、専門家の持っているあいまいな知識は、状況が明確になってはじめて具体化される。RDR 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。RDR ではシステムが誤って判断した場合、専門家によって知識ベースの誤りを正す知識が整合性を保ったまま追加される [Kang 92]。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。これは Schank により導入された“failure-driven memory” [Schank 82] の原理と似ている。また事例ベース推論の研究においても、専門家の知識が理路整然と整理されていない場合でも、事例に対する結論を判断することは専門家にとって容易なことがわかっている [Kolodner 85]。従って、ある事例が RDR システムによって誤って推論された場合、知識獲得プロセスにおいて専門家がその事例と別の事例をどのように区別して判断しているかを認識することが必要になる。これは物事の違いをもとに人間は事物を認識するとする Personal Construct Psychology [Kelly 55] の基本原理と似ている [Gaines 90]。事例間の差異を考え逐次的に知識ベースを修正するという考え方は人間のエピソード的記憶に対する有用なモデルである。RDR では、専門家はシステムが事例に対して推論を行い誤判断した場合のみ、誤事例と以前に正しく分類された事例との間の異なる特徴を判断するだけでよい。

RDR の知識ベースは、図 1(a) に示す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、3 章で説明する。

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 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ルールの帰結部とする。図 1(a) 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 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う必要がある。図 1(a) 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 の知識獲得のステップを述べる。図 1(b) に知識獲得の様子を示す。

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 (図 1(b) で difference list は $\{A2, \text{not}(A1), C2, \text{not}(C1)\}$ 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 (図 1(b) で専門家は A2 と $\text{not}(C1)$ 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 (図 1(b) で新しいルールは $\text{If } A2, \text{not}(C) \text{ then}$ 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 (図 1(b) では 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 の知識獲得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。つまりあるノードの YES branch 以下のノードは全てそのノードの例外ノードとなる。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. Ripple Down Rules 法におけるデフォルト知識

一般的にデフォルト知識とは、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定義することができる。RDR においてはデフォルト知識を実現するために、二分木で表現される知識ベースにおいてルートノードのもつルールにおいて条件部がないもの (常にどのような事例に対しても満たされるという意味である) を考える。そのようなルールの帰結部がデフォルトクラスと呼ばれる特別な結論となる。他の全てのノードが YES branch と NO branch の両方を持つ可能性があるのに対して、ルートノードは YES branch だけしか持たない (図 1 参照)。ルートノード以下の全てのルールはルートノードに対する例外ルールである。

逐次的な知識獲得により RDR では、デフォルト知識に関する知識は積極的に獲得せず、それ以外の知識を積極的に獲得する。例えば結論 A がデフォルトクラスとして設定され、結論 A を持つ事例がシステムに与えられたとしよう。知識ベース上にその事例に対して発火するルールが全くない場合でも、システムはデフォルトクラス A を用いて正しく分類できる。この場合、分類自体は正しいが、結論 A を特徴づける明確な知識は獲得されない。端的に言えば、矛盾する知識が推論パス上に存在しないので結論は A であるはずだ、ということである。従って同じ問題領域でも、デフォルトクラスの設定が異なると、異なる性能 (分類精度、学習速度やサイズ) をもつ異なる知識ベースが構築される。

問題領域において取りうるクラスの数と同じだけ、異なるデフォルトクラスが存在するが、従来の RDR の研究では、問題領域の中で最も事例の数が多きクラス (最頻事例クラス) がデフォルトクラスとして選ばれることが多かった [Kang 96]。RDR ではデフォルトクラスに対する例外ルールやさらに洗練ルールとして、デフォルトクラスを含む他のクラスに対する知識しか獲得する必要がほとんどないので、最頻事例クラスをデフォルトに設定することは、知識ベースのサイズが小規模になることも含め、自然なことのように思われる。しかし最頻事例クラスはそのクラスを記述したり特徴づけることが簡単であるということの意味するものではない。同じことは非最頻事例クラスについても言える。

従って、それぞれ異なるデフォルトクラスを設定した場合、各々の知識ベースが精度やサイズなどの点においてどのように異なるのかを調べることは重要である。本研究では人間の専門家の代わりに計算機プログラムにより構築された知識ベースを有する代替専門家 [Compton95] からの知識獲得を行い、デフォルトクラスが異なる場合の知識ベースの性能への影響を調べる。そして得られた結果と代替専門家の持っている各クラスの特性をともに、

表 1 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	1728	4	Nom.* 6	Iris	150	3	Num. 4
Tic-Tac-Toe	958	2	Nom. 9	Page-block	5473	5	Num. 10
Nursery	12960	5	Nom. 8	Optdigits	5620	10	Num. 64
Connect-4	16889	3	Nom. 42	Yeast	1484	10	Num. 8
Mushroom	8124	2	Nom. 22	Waveform	5000	3	Num. 21
Monk	1711	2	Nom. 6	Image	2310	7	Num. 19
Titanic	2201	2	Nom. 3	Cmc	1473	3	Mixed 7/2
Ann-thyroid	7200	3	Mixed*** 15/6	German	1000	2	Mixed 13/7
Pendigits	10992	10	Num.** 16	Shuttle	14500	7	Num. 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/連続値

構築される知識ベースが、なぜまたはどのようにして、異なる性能をもつのかという理由を議論する。そして性能の高い知識ベースを構築するためのデフォルトクラスを決定するための規範を提案する。

これまでの RDR 研究で調べられていなかったもう一つの重要な問題点はデータに含まれるノイズに対する扱いである。RDR では専門家はいつでも利用できる存在であり、データの質を正しく判断することができるという仮定があった。しかし実世界の問題でノイズが存在するのは当然のことであり、専門家でない我々にはそれを取り除くことができない。従って、知識源である人間の専門家の誤判断を許容しなければならない。本論文で提案するデフォルトクラスの規範がノイズデータに対しても保たれるかどうかを確認する必要がある。

4. 実験

4.1 データセット

デフォルトクラスの違いによる RDR の知識ベースの性能を調べるために、California 大学 Irvine 校の機械学習データライブラリ [Blake 98] から 17 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 [Delve 98] から 1 データセットを選んだ (表 1 参照)。うち 7 データセットは全部の属性が離散値、8 データセットは全部の属性が連続値、そして残り 3 データセットは離散値属性と連続値属性の両方を含む。

4.2 実験方法

4.1 節で示したデータセットに対し、代替専門家 [Compton 95] を用いて異なるデフォルトクラスに対する RDR の知識ベースの特性評価を行う。具体的には各デフォルトクラスにおいて、訓練データの割合を変化させた場合、またはノイズレベルを変化させた場合の知識ベースの精度、学習速度そしてサイズについて評価する。また RDR 法との比較対象のために別に用意した機械学習法との性能比較も行う。

元来、RDR は専門家からの知識獲得を目指しているが、実験のため個別のデータセットに対して専門家を準備するのは物理的に困難であるうえ、人間の場合は生理

的变化などにより実験の再現性が保証されない。計算機に基づく代替専門家の場合、実験の再現性は保証され、かつ本研究では知識ベースの性能を定性的に調べる必要があるため、代替専門家を用いることとする。ただし機械学習法 C4.5 を、代替専門家の知識であるインダクションルールを獲得する手法として、および RDR 法による知識獲得と比較するための機械学習の手法として、2 通りで用いる。

1 機械学習法 C4.5

RDR のデフォルト知識に関する特性評価とは別に、RDR の知識ベースの特性を確認するために、一般的な機械学習法として用いられる C4.5 [Quinlan 93] の知識ベースと比較する。C4.5 は ID3 [Quinlan 86] をもとに改良が加えられた手法であり、属性と値のペアの集合で一つの事例データを表現し、事例データのクラスを特定するためにはどのような属性がどのような値を持っていればよいかを分類する決定木を学習する。C4.5 による学習は RDR と異なり逐次的ではないので、RDR の知識ベースの性能を評価する場合と同じ訓練データを用いて C4.5 を実行する。その際、生成された決定木から導出されるインダクションルール群を RDR と比較するための C4.5 の知識ベースとした。このため知識獲得の初期段階において、RDR は訓練データが少なくとも問題領域の全事例を用いて構築された代替専門家からの知識獲得が行えるのに対して、C4.5 は少ないデータで学習しなければならないということになる。テストデータにも、RDR のテストで用いるもの (データセットの 25%) と同じものをを用いた。

2 代替専門家

具体的には各データセットの各ノイズレベルに対応した全データを使って C4.5 を実行し、生成された決定木から導き出されるインダクションルール群を使用する。これは与えられたデータセットから導き出せる最高の性能を持つ知識と見なすことができ、これを代替専門家の持つ知識とする。人間の専門家の場合、RDR システムの出力を随時チェックしてその結果が間違っていれば、difference list をもとに判断して新しいノードの条件部を決定するのと

表 2 自動車の評価データセット (Car)

クラス名	<i>unacc</i>	<i>acc</i>	<i>good</i>	<i>vgood</i>
事例数	1210	384	69	65
代替専門家の持つインダクションルール数	11	53	15	15
代替専門家の持つ誤事例数	33	56	33	15
最小記述長 (ビット数)	264.4	727.7	277.2	237.5
知識ベースの精度 (順位 *)	2	1	3	3
学習速度 (順位 *)	2	1	2	3
知識ベースのサイズ (順位 *)	3	1	2	4

* 小さい方が良い。精度とサイズに関しては訓練データの割合が 75%の場合で評価した。学習速度は訓練データの割合が 20%の場合で評価した。著者が公平な視点から見てクラス間の差がほとんどない場合は、それらのクラスに対する順位を同じとした。

同様、代替専門家をを用いる場合も、RDR の出力をチェックし代替専門家の判断と異なれば、代替専門家から知識獲得を行う。ただし本研究では新しいノードの条件として、推論事例に対して正しい結論を与えた C4.5[Quinlan 93] のインダクションルールの条件部と difference list の共通部分の全ての条件を選ぶものとする。インダクションルールの条件部 (専門家の知識と想定している) と difference list の共通部分のうち一つだけ選んでも RDR の知識ベースは構築できるが、共通部分のうち全てを使う方が知識獲得の面から見て効率が良いからである。つまり、専門家が持っている知識を全部使うという方針を採用した。

計算機に基づく代替専門家であるので、実際の専門家より分類精度が良いということは期待できない。ノイズが含まれている場合、代替専門家は幾つかの事例を誤判断してしまうかもしれないが、専門家はこのような事例を代替専門家よりも正しく判断する可能性は高いと思われる。また C4.5 では訓練事例に特化した分類決定木からルールを作る際に一般化するので、データにノイズが全く含まれない場合でも、代替専門家の分類は僅かなエラーを含んでいる。しかし C4.5 ではデータから獲得された知識である決定木を簡単な操作で If-Then ルール群に変換することができ、If-Then ルールの表現は代替専門家の知識として扱いやすい。また従来の RDR 研究では C4.5 などの機械学習プログラムが代替専門家として用いられてきた [Compton 95]。そこで、本研究でもそれに倣うことにする。

3 ノイズデータの生成

ノイズデータの生成方法は離散値属性と連続値属性とで異なる。離散値属性の場合、データの各属性に対してある一定の確率でノイズを加えるかどうかを決定する。そしてノイズを加えることが決まった属性に対しては、真の属性値を等確率で他の属性値に代える。連続値属性の場合は、各属性値が取りうる区間のある一定の割合 (ノイズレベル) に対応した区間と等しい大きさの標準偏差を持つガウスノイズを真の属性値に加える。ただしノイズをデータの属性値にだけ加え、クラス情報には加えないものとした。我々は元のデータセットに割合が 0% (つまりノイズなし)、10%、20% のノイズを加えた三つの

データセットを用意した。

4 データ準備

RDR では逐次的に知識獲得を行うので、訓練データの順序は知識ベースの性能に影響する。本研究ではこの順序の影響を取り除くため、代替専門家を構築するために使用するものと同じ各ノイズレベルに対応したデータセットの全データの順序をランダムサンプリングにより並び替え、データの順序が異なる 10 個のデータセットをそれぞれ用意した。この 10 個のデータセットに対して、データセットの先頭から 75% の事例を訓練データのために、残りの 25% をテストデータのために用いる。RDR がどのくらい速く、そしてどのくらい正しい知識を獲得するかということを評価するため、データセットの先頭から 1%、2%、4%、5%、7%、10%、15%、20%、30%、45%、50%、60%、75% のデータセットをそれぞれ訓練データとし、構築した知識ベースの予測エラー率を残りの 25% のテストデータを用いて求めた。

5 分類精度

テストデータに対する予測エラー率を RDR で構築される知識ベースの精度の指標として用いる。ただしこの値はそれぞれ異なる順序の 10 個のデータセットの残り 25% に対してのエラー率を別々に求め、平均化したものである。

6 知識ベースのサイズ

RDR の場合は二分木のノードの合計数を、C4.5 の場合はインダクションルール数を、それぞれの知識ベースのサイズとして定義した。これは RDR の二分木の各ノードにはインダクションルールと同じ形式である If-Then ルールが一つずつ含まれているからである。分類精度の場合と同じく、10 個のデータセットに対する平均をとる。

4.3 実験結果

紙面のスペースの関係上、18 データセットのうち詳細な結果はデータセット “Car” についてのみ、残りは全部を要約した結果のみを示す。

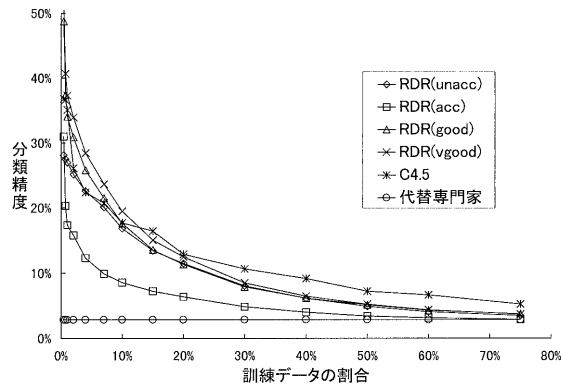


図 2 データセット Car における訓練データの割合とエラー率の関係 (例えば RDR(*unacc*) はデフォルトクラスが *unacc* であることを示す.)

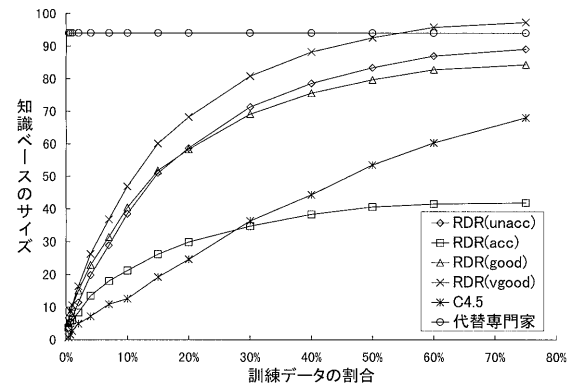


図 3 データセット Car における訓練データの割合と知識ベースのサイズの関係

表 2 はデータセット “Car” における、デフォルトクラス別の実験結果と各クラスの情報を示したものである。このデータセットでは六つの離散値属性と “unacc(unacceptable)”, “acc(acceptable)”, “good”, “vgood(very good)” の四つの分類クラスがある。この表の 2 行目は各クラスの事例の数 (全 1728 事例で、クラス unacc が最頻事例クラスである) を示している。3 行目は代替専門家の知識であるインダクションルール群のうち、各クラスを帰結部に持つインダクションルールの数を示しており、そして 4 行目はその各クラスのインダクションルール群が全事例に対して誤判断する事例の数である。

図 2 と図 3 はデータセット “Car” において、4・2 節の “データ準備” で示した様々な割合の訓練データを用いて構築した場合の、機械学習法 C4.5 とそれぞれデフォルトクラスが異なる 4 種類の RDR の結果を示したグラフである。これらのグラフにより訓練データの割合が増加するほど、知識ベースのサイズは増加し、それとともに精度 (エラー率) が改善されているのがわかる。図 2, 図 3 において精度およびサイズに関しては、訓練データの割合が 100% のポイントにおける値をプロットすると代替専門家の値と C4.5 の値は完全に一致する。また、C4.5 の値の変化は 4 種類の RDR の値の変化よりも緩やかであり、C4.5 よりも 4 種類の RDR の学習速度の方が速いのがわかる。これは RDR が問題領域の全事例を用いて構築した代替専門家から知識を獲得しているからであり、この特性は [Mansuri 91] で示された結果と一致する。つまり、人間の専門家からは、データのみからの知識獲得である機械学習に比べて、少ないデータで一般的な知識を獲得することができることを例証している。

4 種類の RDR に関しては、デフォルトクラスがクラス acc である RDR(acc) の学習速度が一番速く、RDR(acc) の知識ベースのサイズが最も小さい。このことにより、デフォルトクラスとして最頻事例クラス (この場合はクラス unacc) が設定されている RDR が必ずしも一番良い性能であるとは限らないということがわかる。これは表 2 でも示されている。表 2 の 6 行目, 7 行目そして 8 行目は、

図 2, 図 3 の結果を要約したものであり、各デフォルトクラスに対する知識ベースの精度、学習速度そしてサイズに関する順位を示している。この表で特徴的なことは、どの順位でも 1 位になっているクラス acc では、代替専門家の持つ誤事例数が最も多く、さらに代替専門家の持つインダクションルール数も一番多くなっていることである。3 章で述べたように、RDR ではデフォルトクラスに関する知識が積極的に獲得されないため、RDR(acc) はクラス acc に関する知識を積極的に獲得せず、他のクラスに関する知識、より正確にはクラス acc とその他のクラスとを区別できる知識を獲得する。従ってこの実験結果により、RDR(acc) の知識獲得が最も速いのは、クラス acc よりも少ない誤事例と少ないルール数を持つその他のクラスに関する知識を積極的に獲得するからであると考えられる。またクラス acc 以外の三つのデフォルトクラス (unacc, good, vgood) を比較すると、総合的に判断して RDR(vgood) の性能が僅かに悪くなっている。表 2 を見ると、デフォルトクラス vgood は代替専門家の誤事例数が最も少なく、ルール数は 2 番目に少なくなっている。このような実験結果は他のデータセットに対する実験からも数多く見られた。

4・4 デフォルト知識の決定規範

4・3 節の結果を総合すると、デフォルトクラスを表現するのに必要なインダクションルールがより多く、そのクラスに対する代替専門家の誤事例がより多い場合、RDR の学習速度はより速くなり、RDR の知識ベースのサイズはより小さくなると推測できる。つまり、インダクションルールを多く必要とするクラスを選べば、知識獲得過程において後々多くのルールを追加しなければならない状況を避けることができるということである。また同じように、代替専門家の持つ誤事例が多いクラスを選べば、RDR はそのクラス以外の誤事例の少ないクラスに関する知識を積極的に獲得することになり、誤事例が cornerstone case として RDR の知識ベースに記録されることを避けられるはずである。

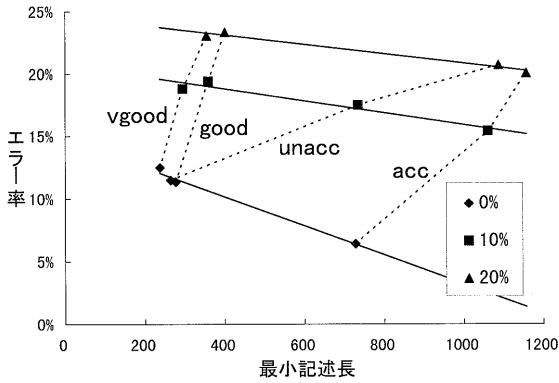


図 4 データセット Car における三つのノイズ率 (0%,10%,20%) に対する 4 種類の RDR のエラー率と最小記述長の関係 (ただし訓練データの割合は 20%である)

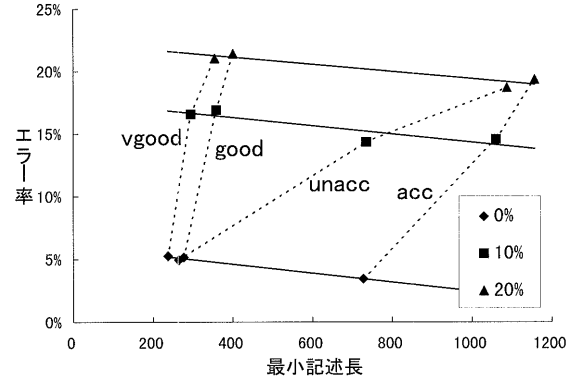


図 5 データセット Car における三つのノイズ率 (0%,10%,20%) に対する 4 種類の RDR のエラー率と最小記述長の関係 (ただし訓練データの割合は 50%である)

ここで、上述のような二つの要因を一つの指標で表現するために、各クラスに対する代替専門家のルール数と誤事例数の両方を考慮した新たな指標である最小記述長 (Minimum Description Length:MDL) を導入する (実際の値は表 2 の 5 行目に示している)。

この指標は各クラスのインダクションルール群とそのルール群で誤ってしまう誤事例との符号化コストの合計で計算される。

あるクラスのインダクションルール群を符号化するのに必要なビット数は

$$\sum_{i=1}^N \left\{ \log_2(AttNum) + \log_2 \left(\frac{AttNum}{M_i} \right) + \sum_{j=1}^{M_i} ValueBits(i,j) \right\} \quad (1)$$

である。ただし、 N はインダクションルール数、 $AttNum$ はそのデータセットが取りうる属性の数、 M_i は i 番目のインダクションルールの条件部における条件の数、 $ValueBits(i,j)$ は i 番目のインダクションルールの条件部における j 番目の条件を符号化するために必要なビット数である。 $\log_2(AttNum)$ の項は i 番目のインダクションルールの条件部における条件の数 M_i を記述するのに必要なビット数を表しており、属性数 $AttNum$ と第 1 項で得られる条件の数 M_i を用いて第 2 項で計算される記述長をもつ符号列を復号化すると、条件部に用いられている複数の属性を特定することができる。 $ValueBits(i,j)$ は以下のように計算される。

$$ValueBits(i,j) = \begin{cases} \log_2(ValueNum(i,j)) & \text{(離散値属性の場合)} \\ \log_2(2) + \log_2(CutOff(i,j)) & \text{(連続値属性の場合)} \end{cases}$$

ただし $ValueNum(i,j)$, $CutOff(i,j)$ はそれぞれ i 番目のインダクションルールにおける j 番目の条件の離散値属性が取りうる属性値の数、連続値属性の取りうる閾値の数である。連続値属性の場合だけ、条件に用いられ

る不等号 ($C4.5$ の決定木から導出されるインダクションルール群には “ \leq ”, “ $>$ ” の 2 種類しかない) を記述するのに必要なビット数を表す項 $\log_2(2)$ が必要である。

誤事例を符号化するのに必要なビット数は

$$\log_2 \left(\frac{r}{2} + 1 \right) + \log_2 \left(\binom{r}{fp} \right) + \log_2 \left(\frac{n-r}{2} + 1 \right) + \log_2 \left(\binom{n-r}{fn} \right) \quad (2)$$

である。ただしそのルール群は、 fp 個を false positive, そして fn 個を false negative とし、全事例 n 個のうち r 個をカバーするものとする。第 1 項は fp を記述するのに必要なビット数、第 2 項はルールに包含される事例群の中の false positive を指定する文字列を記述するためのビット数、第 3 項は fn を記述するためのビット数、第 4 項は包含されない事例群の中の false negative を指定する文字列を記述するためのビット数である。

各クラスの最小記述長は、代替専門家のインダクションルールを構築する際に計算されるが、上式で定義されたルールの符号長に関しては過大評価であることが知られているので、前者の値に 0.5 の重み因子を用いて二つの値を合計したものを最小記述長とする [Quinlan 93]。

代替専門家 ($C4.5$) はルールと誤事例の全符号長が最小になるように、各クラスにおいてインダクションルール群を構築する。表 2 の結果より、最小記述長が最大であるクラスをデフォルトクラスに設定した場合、RDR が “知識ベースの精度”, “知識獲得の速さ”, “知識ベースのサイズ” の 3 点の総合で最も良い性能を示しているのがわかる。また最小記述長の値が最小のクラスを設定した場合、RDR は精度とサイズの点に関して最も悪い順位になっている。以下に詳しく示すが、この関係は問題領域にノイズが含まれる場合でも見られる。

図 4 と図 5 はそれぞれデータセット “Car” における訓練データの割合が 20% と 50% において、異なるノイズレベル (0%,10%,20%) に対する 4 種類の RDR の最小記述長と知識ベースの精度との間の関係を示している。こ

表3 評価結果(ノイズ0%)

データセット名	知識ベースの精度		知識獲得の速度		知識ベースのサイズ		総合
	MMDL	30%以内	MMDL	30%以内	MMDL	30%以内	
Car	Yes		Yes		Yes		Yes
Tic-Tac-Toe	Yes		Yes		Yes		Yes
Nursery	Yes		Yes		Yes		Yes
Connect-4	Yes		Yes		Yes		Yes
Mushroom	Yes		Yes		Yes		Yes
Monk	No	No	No	No	Yes*		No
Titanic	Yes		Yes		No	No	No
Ann-thyroid	Yes*		Yes*		Yes		Yes
Pendigits	No	No	No	No	No	Yes	No
Iris	Yes		Yes		Yes		Yes
Page-block	Yes		Yes		Yes		Yes
Optdigits	Yes		Yes		Yes		Yes
Yeast	Yes		Yes		Yes		Yes
Waveform	No	No	No	No	No	No	No
Image	Yes		Yes		No	No	No
Cmc	Yes		Yes		Yes		Yes
German	No	No	Yes*		No	No	No
Shuttle	Yes*		Yes*		Yes		Yes
	14/18	14/18	15/18	15/18	13/18	14/18	12/18

“MMDL”：MMDLのクラスをデフォルトクラスとした知識ベースの性能について示す。MMDLのクラスが最も良い場合はYes、著者が公平な視点から見て、MMDLのクラスと性能が最も良いクラスとの差がほとんどない場合はYes*、それ以外はNoとする。“30%以内”：“MMDL”がNoであるデータセットに関して、MMDLのクラスの性能がほぼ上位30%以内の順位に入るか否かを示す。入る場合はYes、入らない場合はNoとする。“総合”：精度、速度、サイズの3点を総合的に判断して、MMDLのクラスが良い性能であるか否かを示す。良い性能である場合はYes、そうでない場合はNoとする。

れより、どのノイズレベルそしてどのデフォルトクラスにおいても、訓練データの割合が増加すれば(20%から50%になれば)、エラー率が小さくなるのがわかる。そしてデータセットにノイズが多く含まれるにつれて、4種類のRDRのエラー率のどれかが高くなるのがわかる。しかし、4種類のRDRのうち最小記述長が最大であるRDRが相対的にほぼ一番良い性能を示している。さらに最小記述長が大きいクラスほどエラー率は低いという相関が高いことがわかる。同様の結果が他のデータセットに関しても得られた。従って、データにノイズが含まれる場合にも、確かに最小記述長が最大であるデフォルトクラスを持つRDRが良い性能を示すと考えられる。

表3と表4は、それぞれ問題領域にノイズがない場合(ノイズ0%)とある場合(ノイズ10%)に対して、最小記述長が最大である(Maximum Minimum Description Length:MMDL)クラスがデフォルトクラスに適しているというヒューリスティックな規範が実際に全18データセットで成り立っているかどうかをまとめた表である。ただし“知識ベースの精度”と“知識ベースのサイズ”に関しては、訓練データの割合が75%のポイントで得られた分類精度と知識ベースのサイズをもとに評価しており、“知識獲得の速度”に関しては、訓練データの割合が20%のポイントで得られた分類精度をもとに評価している。表3からノイズがない場合、精度に関しては18データセット中14データセット(Yes*のクラスも含む)で、知識獲得の速さに関しては18データセット中15セットで、サイズに関しては18データセット中13セットで、MMDLであるクラスが一番であることがわかる。もしMMDLに

基づく規範を上位30%にまで緩和すれば、この指標により全ての性能(精度、速さ、サイズ)に関して18データセット中12データセットで、性能の良い知識ベースが構築されることがわかる。表4からノイズがある場合でも、MMDLに基づく規範を上位30%にまで緩和すれば、この指標により全ての性能(精度、速さ、サイズ)に関して18データセット中12データセットで、性能の良い知識ベースが構築されることがわかる。従って、MMDLに基づく規範はノイズデータに関しても成立すると結論づけられる。

問題領域の全事例を用いて計算されるMDLをもとにすれば、適切なデフォルトクラスが決定できることがわかったが、MDLを計算するのに全データの蓄積を待たなければならぬのでは意味がない。そこで、全データの10%だけを使って計算されるMDLを用いた評価も行った。その結果、全事例のうちの10%の事例群を選んで計算されるMDLの値が最も大きいクラスは、全事例から計算されるMDL値が大きい上位のクラスのうちのどれかと一致する、ということが18データセット中15データセットにおいて確認できた。つまりRDR法を用いて知識獲得を始める前に、ある程度少量のデータが集まるのを待てば、MDL値を計算することができ、それに基づき性能の高い知識ベースが構築できる可能性が高い。

本章の結果は少なくとも、ある程度事例を貯めてから最少記述長を計算し、最もMDL値の大きいクラスをデフォルトクラスとして用いれば、その後構築される知識ベースの性能はその他のクラスをデフォルトクラスとして用いる場合よりも良くなることを示している。またそ

表 4 評価結果 (ノイズ 10%)

データセット名	知識ベースの精度		知識獲得の速度		知識ベースのサイズ		総合
	MMDL	30%以内	MMDL	30%以内	MMDL	30%以内	
Car	No	No	Yes		Yes		No
Tic-Tac-Toe	Yes		Yes		Yes		Yes
Nursery	Yes		Yes		Yes		Yes
Connect-4	Yes		Yes		Yes		Yes
Mushroom	Yes		Yes		Yes		Yes
Monk	Yes*		Yes*		No	No	No
Titanic	Yes		Yes		Yes		Yes
Ann-thyroid	Yes		Yes		Yes		Yes
Pendigits	No	Yes	Yes		No	Yes	Yes
Iris	No	No	Yes*		Yes		No
Page-block	Yes		Yes		Yes		Yes
Optdigits	Yes		Yes*		Yes		Yes
Yeast	Yes		Yes		Yes		Yes
Waveform	Yes		Yes		No	No	No
Image	Yes		Yes		No	Yes	Yes
Cmc	No	No	No	No	No	No	No
German	No	No	Yes*		No	No	No
Shuttle	Yes*		Yes*		No	Yes	Yes
	13/18	14/18	17/18	17/18	11/18	14/18	12/18

のようなデフォルトクラスは必ずしも最頻事例クラスのことではないことも示された。提案した MMDL に基づく指標は RDR の知識ベースの定性的な分析に基づいたものであり、ヒューリスティックではあるが、これがデフォルトクラスの良さを決定するのに有効な指標であることがわかった。

RDR は専門家があいまいで整理されていない知識しか持っていないような問題領域も対象にしており、そのような問題領域では人間の専門家が前もって各クラスのエラー率や分類知識を正確に知っていることは考えにくい。しかしながら専門家が自分自身の知識に対して主観的に持っている確からしさや自信というものはクラス毎に違う可能性がある。例えば専門家があるクラスに対して持っている知識が複雑であり記述量が多くなると考えている場合、そのクラスの MDL 値はその他のクラスより大きいと考えることができ、我々の提案する規範が適用できると考えられる。

5. おわりに

本論文では、デフォルト知識の選択の違いが、知識エンジニアのインタビューや事前分析を必要としない知識獲得技術である RDR の性能に対し与える効果に関して、実験的結果をもとに示した。例外知識を逐次的に追加する RDR の知識獲得アルゴリズムにより、知識獲得がデフォルト知識に非常に依存することがわかった。

さらに、デフォルトクラスとしてインダクションルールと誤事例との最小記述長が最大であるクラスを選ぶと、精度・サイズ・速度の点で良い性能を有する知識ベースを構築できることを明らかにした。これは RDR のデフォルトクラスとしてそのクラスを記述するルール数が最大のものと同様専門家の誤事例が最も多いクラスを選ぶこ

とになる。この特徴はデータにノイズが含まれる場合でも成立することが示された。

RDR は、一つの事例に対して必ず一つの結論が存在する問題領域を対象としているが、今後の課題としては、一つの事例に対して複数の結論を持つような問題領域への適用が考えられる。実際、そのような問題領域を対象として、RDR の概念を取り入れた MCRDR (Multiple Classification RDR) [Kang 95, Kang 96] と呼ばれる手法が研究されている。MCRDR の最近の進展として、設計問題への適用 [Ramadan 97] が挙げられるが、分類問題以外への適用は未解決の部分が多い。

また、本研究で得られた結論は RDR 法による知識獲得に限ったものであるが、最小記述長のような知識情報を記述するための記述長という指標をもとに、デフォルト知識を決定する手法は、他の知識獲得手法や機械学習法においてデフォルト知識を決める場合にも、役立つ可能性があると考えられる。

最後になるが、多数のユーザや専門家が同一の計算機ネットワークにアクセスするネットワーク環境上で動的な問題環境に対しても強く、性能の高い知識ベースシステムを構築するのに、RDR の特性や本研究で得られた結果などは役に立つと考えられる。ネットワーク環境における多数の専門家からの知識の獲得や修正は新しいチャレンジである。

謝 辞

熱心な御議論および御助言を頂いた湖西大学 (韓国) の Byeong Ho Kang 博士に心より深謝申し上げます。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., Keogh E. and Merz C. J.: UCI Repository of Machine Learning Databases.
http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.
- [Compton 89] Compton P., Horn K., Quinlan J. R. and Lazarus L.: Maintaining an Expert System. In *Application of Expert Systems*, ed. J. R. Quinlan, pp. 366-385, Addison Wesley, 1989.
- [Compton 90] Compton P. and Jansen R.: A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition*, Vol. 2, pp. 241-257, 1990.
- [Compton 91] Compton P., Edwards G., Kang B. H., et al: Ripple Down Rules: Possibilities and Limitations. *Proc. of the 5th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1991.
- [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.
- [Compton 95] Compton P., Preston P. and Kang B. H.: The Use of Simulated Experts in Evaluating Knowledge Acquisition. *Proc. of the 9th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1995.
- [Delve 98] The University of Toronto.: Data for Evaluating Learning Valid Experiments.
http://www.cs.utoronto.ca/~delve/data/datasets.html, 1998.
- [Gaines 90] Gaines B. and Shaw M.: Cognitive and Logical Foundations of Knowledge Acquisition. *Proc. of the 5th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1990.
- [Kang 92] Kang B. H. and Compton P.: Knowledge Acquisition in Context: Multiple Classification Problem. *Proc. of PRICAI'92*, Vol. 2, pp. 847-853, 1992.
- [Kang 95] Kang B. H., Compton P. and Preston P.: Multiple Classification Ripple Down Rules: Evaluation and Possibilities. *Proc. of the 9th Knowledge Acquisition for Knowledge-Based Systems Workshop*, 1995.
- [Kang 96] Kang B. H.: Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules. *Ph.D. Thesis, University of New South Wales*, 1996.
- [Kang 97] Kang B. H., Yosida K., Motoda H. and Compton P.: Help Desk System with Intelligent Interface. *Applied Artificial Intelligence*, Vol. 11, pp. 611-631, 1997.
- [Kelly 55] Kelly G. A.: *The Psychology of Personal Constructs*, New York, Norton, 1955.
- [Kolodner 85] Kolodner J. L.: A Process Model of Case-Based Reasoning in Problem Solving. *Proc. of IJCAI'85*, Vol. 1, pp. 284-290, 1985.
- [Mansuri 91] Mansuri Y., Kim J. G., Compton P. and Sammut C.: An Evaluation of Ripple Down Rules. *Proc. of AKAW'91*, 1991.
- [Quinlan 86] Quinlan J. R.: Induction of decision trees. *Machine Learning*, Vol. 1, pp. 81-106, 1986.
- [Quinlan 89] Quinlan J. R. and Rivest R. L.: Inferring Decision Trees Using the Minimum Description Length Principle. *Information and Computation*, Vol. 80, pp. 227-248, 1989.
- [Quinlan 93] Quinlan J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [Ramadan 97] Ramadan Z., Compton P., Preston P., Le-Gia T., Chellen V., Mulholland M., Hibbert D. B., Haddad P. R. and Kang B.: From Multiple Classification RDR to Configuration RDR. *Proc. of the 10th Knowledge Acquisition for Knowledge-Based System Workshop*, 1997.
- [Schank 82] Schank R. C.: *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press, 1982.
- [Wallace 93] Wallace C. S. and Patrick J. D.: Coding Decision Trees. *Machine Learning*, Vol. 11, pp. 7-22, 1993.

〔担当委員：石塚 満〕

1999年4月9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。



堀内 匡(正会員)

1992年京都大学工学部精密工学科卒業。1994年同大学院工学研究科精密工学専攻修士課程修了。1997年同博士後期課程研究指導認定退学。現在、大阪大学産業科学研究所助手(知能システム科学研究部門)。主に機械学習、知識獲得、知識発見、データマイニングなどの研究に従事。博士(工学)、計測自動制御学会、日本ファジィ学会、IEEE Computer Society、各会員。



元田 浩(正会員)

1965年東京大学工学部原子力工学科卒業。1967年同大学院原子力工学専攻修士課程修了。同年、(株)日立製作所に入社。同社中央研究所、原子力研究所、エネルギー研究所、基礎研究所を経て1995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。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年生まれ。1983年東北大学工学部原子核工学科卒業。1988年同大学院原子核工学専攻修士課程修了。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、昭見堂)など。AAAI、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。