

解説



逆計算：計算の理論における逆問題

4. 論理の逆計算

—計算機科学の立場から†—

磯崎 秀 樹[†]

1. はじめに

演繹推論つまり「論理の順計算」では、理論から演繹により結論を導く。人工知能では、この演繹推論を逆向きにし、結論からそれを導ける理論を構成する学習やアブダクションの研究が行われている。これらの分野の工学的実現方法については他の解説記事^{22), 28), 27), 24)}を参照していただくこととして、ここでは「論理の逆計算」について少し抽象的に考えてみよう。

理想的な知的行為者（エージェント）がすでに自分のもっている理論をもとにして、新情報を反映した新しい理論を作り出すことを**理論変更 (theory change)**あるいは**信念修正 (belief revision)**という。この**信念 (belief)**という言葉は、事実かどうかに関わらず単にある人が「信じていること」を指すのに使われる専門用語であり、日常用語とは異なる。これは事実であるという保証のある**知識 (knowledge)**と区別するためである。この知識と信念の違いは、しばしば様相論理⁴⁾の公理系の差として表される⁵⁾。また、修正という言葉も厳密には分類され、世界が変化したことに追従するほうを**更新 (update)**、思い込みを直すほうを**修正 (revision)**という。(Judea Pearl¹⁵⁾は少し違った区別をしている。彼によれば、エージェントが観測に基づいて主観確率の分布を計算し直すのが更新であり、その確率分布で最大の確率をもつ値の組合せ、つまりもっともらしい世界の記述の変化が修正である。)

理論変更にはさまざまな種類・程度がある。一番簡単なのは、あらかじめ決まっている特定の事

実を追加あるいは削除するだけ、というものである。これはプランニング¹⁾の分野でよく使われる手法であり、STRIPSの追加リスト/削除リストが代表的である。これはある動作をする前の世界を記述する理論が与えられているときに、その理論から削除リストで指定された事実を取り除き、追加リストで指定された事実を追加することにより、動作後の世界を記述する理論を作り出す。しかし、複雑な世界では各行為が及ぼす影響を書き切れないという波及問題 (ramification problem)¹⁾を抱えている。

もう少し複雑なものとしてデータベースの**ビュー更新 (view update)**^{21), 3)}がある。演繹データベースでは、ルールで定義された内包述語とファクトで定義された外延述語をわけ、内包述語の変更が要求されたときは、内包述語自身の定義を変更せず、外延述語を変更することで見かけ上内包述語を更新する。このビュー更新は、人工知能のアブダクションと密接な関係がある。アブダクションは、自分の信念だけから観測結果が導けない場合に、何か単純な仮定を加えてうまく説明しようとする推論である。

さらに複雑なものとして、エージェントが多数相互作用するマルチエージェント環境²⁾における各エージェントの制御の問題がある。エージェントの行動をそのエージェントの知識や信念に基づいて決定しようとする**エージェント指向プログラミング**²⁰⁾や**知識指向プログラミング**¹¹⁾では、通信や観測による知識や信念の時間的変化が大きな問題である。各エージェントがある程度独立して活動しているので、自分の知らないところで世界がどんどん変化していく。したがって以前には正しかったことが現在では間違っていることもまれではない。

このような「理論変更」の観点から「論理の逆

† Reverse Computation for Logic, From the Viewpoint of Computer Science by Hideki ISOZAKI (Information Science Research Laboratory, Basic Research Laboratories, Nippon Telegraph and Telephone Corporation).

†† 日本電信電話株式会社基礎研究所情報科学研究部

計算」をとらえてみよう。すると次のような問題を考えなければならない。

1. 理論変更が満たすべき理想的性質。
2. 結論から理論の候補を作り出す方法。
3. 膨大な数の理論の候補の中から最適な理論を探し出す方法。

1. は逆計算の指針となるべきものであり、2. が具体的な逆計算に相当する。基本的には演繹推論を逆にすればよい。そこで2. で1. に関する議論を説明し、3. で具体例から一般的規則を得る方法を説明し、4. で規則を整理する方法を説明する。3. は2. を適切に制御して1. を満たすようにする方法の問題である。この観点からの成果はまだ特にないようである。これまでに別の観点から提案されている制御方法を4. の終りで簡単に説明する。

2. 理論変更の理想的性質

新しい情報を取り込むときにそれまでの理論の何を残して何を捨てるかの判断が必要になる。もちろん、新しい理論が矛盾しないように注意しなければならない。では、次のような場合どのように考えるのが自然であろうか²³⁾。

ある会社で鈴木さんが経理部長であった。次に田中さんが経理部長になった。そして今度はその田中さんが総務部長になった。ここで「鈴木さんは経理部長ですね」と尋ねられたら、どう答えるべきか。

実際にこのような状況におかれると、多くの人は「昔はそうだったが」というように昔の話を持ち出して説明しようとする傾向がある。これは、信念の変更に関する次の二つの大きな考え方の食い違いが反映されていると思われる。

- 正当な根拠のあるものだけを信じるべきであるという考え方。 **foundations approach** と呼ぶ。Jon Doyle の **TMS**²³⁾ や de Kleer の **ATMS**²³⁾ は foundations approach の代表的な例である。この考え方によると、ある命題を取り除くと、その命題を根拠としていた他の命題も取り除かれる。先ほどの例では、田中が経理部長であることが鈴木が経理部長でないことの根拠となっていたが、田中が総務部長になるとその根拠が失われるため、鈴木が経理部長でないという信念は取り消さなければならない。

- 新しい情報をもたらす矛盾を避けるために

必要最小限の変更を加えるだけで、信念の変更なるべく少なくするべきであるという考え方。 **coherence approach** と呼ぶ。あとで説明する **AGM の公準 (AGM postulates)** が代表的である。

先ほどの例では、田中が総務部長になっても鈴木が経理部長でないことと矛盾しないので後者を取り消さなくてよい。

この両者の関係については Jon Doyle²³⁾ や A. S. Rao ら¹⁷⁾ が研究を行っている。

理論の修正

Gärdenfors ら⁶⁾ は理論の修正について、修正前の理論 T と修正後の理論 T' の望ましい関係を定性的に記述した。これを AGM の公準という。ここでは勝野らの記法に従い、元の理論（と等価な命題論理式：以下同様）を ϕ 、新しい情報 μ で修正した理論を $\phi \circ \mu$ で表す。すると AGM の公準は、1) $\phi \circ \mu$ が μ を含意する、2) $\phi \wedge \mu$ が充足可能なら $\phi \circ \mu \equiv \phi \wedge \mu$ 、3) μ が充足可能なら $\phi \circ \mu$ も充足可能、4) $\phi_1 \equiv \phi_2, \mu_1 \equiv \mu_2$ なら $\phi_1 \circ \mu_1 \equiv \phi_2 \circ \mu_2$ 、などの条件からなる。勝野ら¹⁰⁾ は AGM の公準のどの条件が満たされているかという観点から、これまでに他の研究者によって提案されているさまざまな修正方法を分類している。

修正は故障診断のような応用を考えれば分かりやすい。正常な装置の仕組みを説明する理論を ϕ で表す。すると故障した装置の仕組みは、 ϕ を観測情報 μ で修正した理論 $\phi \circ \mu$ で表される。故障診断では、全体のうちのどこ一部が故障しているだけで、ほとんどのところは正常であると仮定する。つまり、故障を説明できるモデルのうちで正常に最も近いものが新理論のモデルとなる。もう少し正確に言えば、ある解釈の ϕ に関する異常さを定義したときに、 μ のモデルの中で異常さが最小のものが $\phi \circ \mu$ のモデルとなる。一方、 ϕ で成り立っていた特定の命題 μ を成り立たなくする操作を **縮小 (contraction)** といい、縮小した理論を $\phi \bullet \mu$ で表す。縮小についても修正と似た公準を作れる。修正と縮小の間には相補的關係が成り立つ。

なお、自分の信念から導かれないものはすべて偽であるとする閉世界仮説 (closed world assumption) を採り入れると、縮小 $\phi \bullet \mu$ から否定 $\neg \mu$ が導けることになり、修正 $\phi \circ \neg \mu$ と実質的に変

わらなくなってしまう。

理論の更新

AGM の公準は変化のない世界を記述した理論の修正には適しているが、理論が対象世界の変化に合わせて変更される場合、つまり、理論の更新には必ずしも適当ではない。更新は今までの理論が間違っていたことを主張しているわけではないので、今までの理論で複数の可能性（モデル）が考えられれば、更新はそれぞれのモデルについて考慮しなければならない。この点が修正と大きく違う。そこで勝野ら⁷⁾は理論の更新 (update) が満たすべき性質 (KM の公準) を考えた。

修正に対して縮小という相補的概念があったように、更新に対しては削除 (erasure) という概念があり、勝野らの公準を満たす任意の更新と削除は AGM の公準の修正と縮小の間に成り立つのとまったく同じ相補的關係を満たす。また、理論の修正は、対応する心的状態の更新で表せること⁹⁾が示されている。

3. 一般化の方法

述語論理の場合、「すべての...について...が成り立つ」というように具体例を列挙しきれないような一般的関係を簡単に述べることができる。しかし、外界から得られる情報は多くの場合限られた数の具体例だけである。そのため、理論の修正に際しては、いくつかの具体例から、その背後に想定される一般的関係を推定することになる。つまり、述語論理の逆計算を実現する場合、一般化 (generalization) が重要な役割を演じる。なお、一般の複雑な構造の論理式を扱うより、節形式に変換しておいたほうが扱いやすく、また逆計算を考える場合にもレゾリューションや単一化の知識が前提となるので、これらについて簡単に説明しておく。

3.1 節形式

一階述語論理では証明に Robinson のレゾリューション (resolution) がよく使われる。この場合、論理式を節 (clause) という特定の形式に変換して証明を行う。ここでは DEC-10 Prolog の記法にしたがい、変数を大文字で始まる同様の文字列で表す。たとえば、*A*, *Book*, *H_2O* などは変数であり、*put*, *is_list* などは定数である。日本語も定数として使えるものとしよう。これらの定

数は、個体・関数記号・述語記号のどれを表すのにも使えるものとする。また「 ϕ ならば ψ 」を $\phi \rightarrow \psi$ で表す。

原子式の肯定または否定をリテラルという。節はリテラルの論理和で、たとえば

$$\neg \text{子}(Ch 1, P) \vee \neg \text{子}(Ch 2, P)$$

$$\vee \text{同一}(Ch 1, Ch 2) \vee \text{兄弟}(Ch 1, Ch 2)$$

は節である。この節は *Ch 1* と *Ch 2* がともに *P* の子であれば、*Ch 1* と *Ch 2* は同一人物か兄弟であるということを表している。これからは節を次のように集合で表す。

$$\{\neg \text{子}(Ch 1, P), \neg \text{子}(Ch 2, P),$$

$$\text{同一}(Ch 1, Ch 2), \text{兄弟}(Ch 1, Ch 2)\}$$

Prolog のプログラムは \neg の付いていない原子式 (正リテラル) がちょうど一つのもので確定節 (definite clause) という。正リテラルの部分で頭部、それ以外を本体という。正リテラルが一つ以下の節を Horn 節という。

算数で $(p-q)+(q+r)$ から $p+r$ が導けるように、命題論理でも $p \vee \neg q$ と $q \vee r$ から $p \vee r$ が導ける。これがレゾリューションの基本的アイデアで、得られた結果をレゾルベント (resolvent) という。述語論理の場合は、変数部分を任意の項で置き換えられるので、

$$\{\text{父親}(P, \text{健}), \text{母親}(P, \text{健}), \neg \text{親}(P, \text{健})\}$$

$$\{\neg \text{父親}(\text{仁}, Ch 2), \neg \text{女}(\text{仁})\}$$

という二つの節 (親節) から次の結論が得られる。

$$\{\text{母親}(\text{仁}, \text{健}), \neg \text{親}(\text{仁}, \text{健}), \neg \text{女}(\text{仁})\}$$

一般に節 C_1 と節 C_2 のレゾルベントを $C_1 \cdot C_2$ で表す。

この例では「父親(*P*, 健)」と「父親(仁, *Ch 2*)」がマッチするという事実を用いている。このように二つの表現が与えられたときに、その共通の例であるような表現を求める操作を単一化 (unification) という。項 $f(X, b)$ と項 $f(a, Y)$ の共通の例である項は $f(a, b)$ しかない。したがってこの二つの項を単一化すると $f(a, b)$ が得られる。また $f(a, X, X)$ と $f(Y, Z, g(V, Y))$ の共通の例で最も一般的なのは $f(a, g(V, a), g(V, a))$ である。このときの代入 $\theta = \{X \leftarrow g(V, a), Y \leftarrow a, Z \leftarrow g(V, a)\}$ を最汎単一化子 (most general unifier) と呼ぶ、 α に代入 θ を行った結果を $\alpha\theta$ と書く。なお、 $f(a)$ と $f(b)$ のように共通の例が存在しない場合は単一化できない。

節 C_1 とそれになんらかの代入 θ を施した $C_1\theta$ を考えると, $C_1\theta$ は C_1 の例にすぎない. したがって, 理論中に C_1 があれば $C_1\theta$ をもっている必要はない. また, $C_2 \subseteq C_3$ のとき, C_2 が成り立てば C_3 も成り立つ. たとえば

{同一($Ch 1, Ch 2$), 兄弟($Ch 1, Ch 2$),
 \neg 子($Ch 1, P$), \neg 子($Ch 2, P$)}

が成り立てば,

{同一($Ch 1, Ch 2$), 兄弟($Ch 1, Ch 2$), 学生($Ch 1$),

\neg 子($Ch 1, P$), \neg 子($Ch 2, P$), \neg 会社員(P)}

が成り立つ. したがって理論中に C_2 があれば C_3 はなくてよい. そこで, $C_1\theta \subseteq C_3$ のとき, C_1 が C_3 を θ -包摂 (subsume) するという. このとき C_1 は C_3 より一般的, 逆に C_3 は C_1 より特殊である. なお代入によって複数のリテラルが等しくなる可能性があるので, $|C_1| \leq |C_3|$ とは限らない.

3.2 理論修正とプログラムのデバッグ

変更の対象となる理論として pure Prolog のプログラムを考えると, 理論の修正は Prolog プログラムのデバッグとみなせる. このような考え方に基づく学習システムとして, E. Y. Shapiro の MIS (model inference system)^{18), 19)} が有名である. MIS はおおまかにいえば Prolog の対話的デバッグ PDS にプログラムを自動的に修正する機能を付加したものである.

PDS は 1) 正しい例つまり正の例 (positive example) が受け入れられない場合, 2) 正しくない例つまり負の例 (negative example) が受け入れられてしまう場合, 3) 無限ループになってしまう場合のそれぞれに対して, 効率よく対話的に原因を調べていく.

MIS は教師からさまざまな例を受けとり, それを説明する理論つまりプログラムを作り出す. そしてそれまでの理論が間違っていると分かると, PDS によってどの節が間違っていたのかを判定し, その節に代わる節を精密化オペレータ (refinement operator) で探す. 精密化オペレータは節どうしが一般→特殊の関係で結ばれた有向無閉路グラフ (精密化グラフという) を, 一般的なほうから順に調べていく. なお, 引数の型や補助述語などのヒントをあらかじめ与えておかないと, 探索空間が広過ぎて手におえない. このような情

報を与えることは面倒であり, 後述する逆レゾリューションが注目されている.

さて, 理論修正の視点で MIS を見直してみよう. 理論 ψ を Prolog プログラム P の帰結全体に対応づけて考えられる. しかし, AGM 公準の条件 2) は一般化を認めていないので改良が必要である. また, 条件 1) は, 過去の情報より最新の情報を優先するが, MIS による学習は過去の情報をすべて説明できるプログラムを作り出す.

このように, 論理型言語のプログラムを例からの帰納的推論によって学習させる方法の研究分野を, 帰納論理プログラミング (inductive logic programming)²⁵⁾ と呼ぶ.

3.3 一般化の基本概念

個々の例から一般的な理論を作り出すには, 次のような概念¹⁶⁾が重要である.

反単一化

学習の分野では, 単一化と双対的な概念である反単一化 (antiunification) がよく用いられる. 反単一化は二つの項が与えられたときに, その両方を例としてもつ項を求める操作である. たとえば, $f(a, b, g(b, a))$ と $f(p, h(q), g(h(q), p))$ を反単一化すると, $f(A, B, g(B, A))$ が得られる. 反単一化はあきらかに一種の一般化 (generalization) である.

学習の分野では, さまざまな例が一つの概念で説明できることが望ましいが, 一般化しすぎて違うものまでその概念に取り込む不健全な規則を生成することは避けたい. このため, 一般化といってもなるべく特殊なものを考えるべきである. 先ほどの例では, $f(A, B, C)$ より $f(A, B, g(B, A))$ のほうが特殊である.

最小汎化

最も特殊な一般化を最小汎化 (least general generalization, LGG) という. 二つの項の最小汎化は反単一化で実現できる. 二つの原子式の最小汎化は述語が同じでなければならないという条件がある以外は項の最小汎化と同じである. 二つのリテラルの最小汎化は, さらに符号が共通であるという条件が加わる.

θ -包摂を使えば, 二つの節の最小汎化も考えられる. 二つの節 C_1, C_2 の最小汎化は, それぞれに属するリテラルの直積で得られる対の最小汎化 (できないものは無視) の集合から冗長なりテラル

を除いたもの¹⁶⁾である。たとえば、

{子(浩, 仁), 一親(仁, 浩), 一男(仁)}

{子(浩, 恵), 一親(恵, 浩), 一女(恵)}

の最小汎化は

{子(浩, P), 一親(P, 浩)}

である。

4. 規則を整理する方法

理論を作るには、個々の例を一般化するだけでなく、見かけの違う規則を整理し、その背景にある本質的規則を明らかにしなければならない。そこでレゾリューションを逆方向に適用して、背景にある本質的な規則を求めることが考えられる。

これが Muggleton (ILP¹⁴⁾ 12章) の逆レゾリューション (inverse resolution) の基本的なアイデアである。アイデアは単純であるが、探索空間が膨大なため、効率よく適切な理論を得ることは難しく、発展途上の分野である。

Muggleton は V オペレータと W オペレータを提案した。レゾリューションをかけ算 $a \times b = c$ に例えてみると、V オペレータは割り算 $c/a = b$ に相当する。つまり、レゾルベント $C_1 \cdot C_2$ と一方の親 C_1 から他方の親 C_2 を割り出す操作である。これに対して W オペレータは公約数の計算に似ている。 m と n_1, \dots, n_k が整数であることが分かかっていて $m \times n_1, \dots, m \times n_k$ の値だけ教えられて m を推定しようとする場合、これらの値の公約数の中に m は必ず含まれているはずである。そして $\{n_i\}$ が増えれば、次第に m の値を絞り込んでいける。このように、一つの親節を共通にもつ複数のレゾルベントがあるときに、その共通の親節を探し出すのが W オペレータである。

いきなり代入のからむ一般の逆レゾリューションを紹介しても分かりづらいので、命題論理を中心に各オペレータを紹介し、あとで代入の扱いを説明しよう。

4.1 V オペレータ

レゾリューションの各ステップは、二つの節(親)からレゾルベントを導き出す V 字型の図(図-1)で表される。そこで、レゾルベントと一方の親から他方の親を導き出す操作を V オペレータという。レゾリューションで消滅するリテラルは、それぞれの親節の中で反対の符号(正か負か)をもっており、父親と母親のような違いがあ

る。ここでは、一度に複数のリテラルが消滅してしまうことはないものとして、消滅したリテラルがもともと肯定形で入っていた節を正親、否定形で入っていた節を負親ということにしよう。消滅したリテラルは、レゾルベントにはないが既知の親節に現れているのでこれがヒントになる。

負親を推定するほうを吸収オペレータ (absorption operator) という。命題 Horn 節論理では、レゾルベント $h1 :- a \wedge b \wedge c \wedge d$ と正親 $h2 :- c \wedge d$ から負親 $h1 :- h2 \wedge a \wedge b$ が求められる。正親を推定するほうを同定オペレータ (identification operator) という。レゾルベント $h1 :- a \wedge b \wedge c \wedge d \wedge e$ と負親 $h1 :- a \wedge b \wedge k \wedge c$ から正親 $k :- d \wedge e$ が求められる。

4.2 W オペレータ

二つのレゾリューションのステップがどちらかの親を共有していると、その様子は W 字型の図(図-2)で表される。そこで、複数のレゾルベントから共通の親を推定する操作を W オペレータという。特に共通の親節が正親の場合を述語間構成オペレータ (inter-construction operator), また負親の場合を述語内構成オペレータ (intra-construction operator) という。以下の例からも分かるように、補助述語つまり理論名辞 (theoretical term) が生成される。

述語間構成オペレータは $x :- b \wedge c \wedge d \wedge e$ と $y :- a \wedge b \wedge d \wedge f$ をまとめて $z :- b \wedge d$ という共通の正親(新しい述語の定義)を作り出す。同時に $x :- c \wedge e \wedge z$ と $y :- a \wedge b \wedge z$ がそれぞれに対応する負親として得られる。述語内構成オペレータは $x :- b \wedge c \wedge d \wedge e$ と $x :- a \wedge b \wedge d \wedge f$ をまとめて

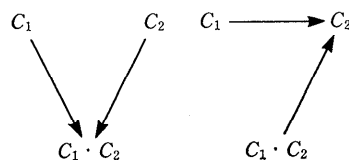


図-1 レゾリューションとVオペレータ

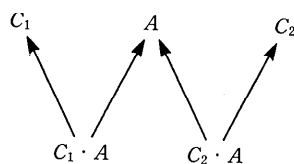


図-2 Wオペレータ

$x :- b \wedge d \wedge z$ という共通の負親を作り出す。同時に $z :- c \wedge e$ と $z :- a \wedge f$ がそれぞれに対応する正親 (新しい述語の定義) として得られる。

4.3 代入の処理

命題論理の場合は、上記のオペレータを簡単に表現できるが、述語論理に拡張すると、代入をとるため処理が面倒になる。W オペレータは単一化代入の行われたレゾルベントどうしを比較するので、レゾルベントの最小汎化をベースとして共通の親節を推定できる (ILP¹⁴) 1章)。図-2 の C_1, C_2 を原子式一つだけの節 (単位節) と仮定し、これらと打ち消し合う A のリテラルを $\neg l$ としたとき、 $A = \text{lbg} (A \cdot C_1, A \cdot C_2) \cup \{\neg l\}$ と表せる。つまり、 $\text{lbg} (A \cdot C_1, A \cdot C_2)$ の本体に C_1, C_2 とマッチする条件が一つ追加される。たとえば、

$A \cdot C_1 : \{\text{姪}(\text{長女}(B), C), \neg \text{兄弟}(B, C)\}$

$A \cdot C_2 : \{\text{姪}(\text{次女}(D), E), \neg \text{兄弟}(D, E)\}$

の最小汎化は

$\{\text{姪}(F, G), \neg \text{兄弟}(H, G)\}$

であり、

$A : \{\text{姪}(F, G), \neg \text{兄弟}(H, G), \neg p(F, H)\}$

$C_1 : \{p(\text{長女}(B), B)\}, C_2 : \{p(\text{次女}(D), D)\}$

となる。ここで $p(X, Y)$ は X が Y の娘であることを示す新述語である。

一方、V オペレータの場合、単一化代入後のレゾルベントと代入前の既知の親から代入を推定し、代入前の未知の親を推定しなければならない。このためレゾルベントに代入の逆操作を行う。ただし、代入とその結果の項が分かっても、 $f(a, a) = f(a, X) \{X \leftarrow a\} = f(X, X) \{X \leftarrow a\}$ のように元の項が確定しない場合もある。

また、正親 $p(X, a) :- r(d)$ と負親 $q(c) :- p(b, Y)$ からレゾルベント $q(c) :- r(d)$ が得られるが、このレゾルベントと正親から負親を推定しようとしても、レゾルベントにも正親にも b がいないので、負親を完全に再現できない。しかし、正親が $p(X, a) :- r(d, X)$ なら、レゾルベントは $q(c) :- r(d, b)$ になるので、 b が残る。そこで、Muggletonらは頭部に含まれる変数が必ず本体にも含まれている節を弱生成的 (weakly generative) と呼び、すべての変数が複数のリテラルに現れる節を強生成的 (strongly generative) と呼んで (ILP¹⁴) 1章) その性質を利用している。

また、別のアプローチとして、関数記号を含むプログラムをある特定の形式に変換する方法 (ILP¹⁴) 6章) も研究されている。たとえば Rouveirol の flattening (ILP¹⁴) 3章) は、「長女(隆)」という項が節に含まれていると、この項を変数 X に、「隆」を Y に対応させ、「長女- $p(Y, X) \wedge$ 隆- $p(Y)$ 」を本体に加える。「隆- $p(Y)$ 」を条件から外すと X は「長女(Y)」に対応し、逆代入を行ったのと同じ効果が得られる。

4.4 新理論の探索方法

帰納論理プログラミングでは、V オペレータや W オペレータ以外にもいくつかのオペレータが考えられているが、これらの適用方法が問題である。命題 Horn 節の場合には高速のアルゴリズム¹³⁾が知られているが、述語論理の場合はいたるところに任意性があるため探索空間が膨大なため、制御が難しい。AGM のような理論変更の理想的性質まで考慮した研究はまだないようである。

CIGOL (ILP¹⁴) 12章) というシステムでは、節中の記号の数で節の複雑さを定義し、なるべく単純な節を選ぶという素朴な方法が使われた。情報量的指標に基づいて仮説の価値を判断するシステムもある (ILP¹⁴) 17章)。いくつかの重要な述語の満たす性質に着目して、それらの定義が効率よく逆計算できるように制限を加えること (ILP¹⁴) 13章) も行われている。さまざまな抽象度の仮説が可能なきときは、抽象度の低い仮説から調べていくのが一つの手である。Muggleton (ILP¹⁴) 1章) の最も特殊な V オペレータ (most specific V operators) は、可能な親節の中で最も特殊な親節を作り出す。

計算量を下げる別の方法として、論理的な厳密さを追求せず、PAC 学習²⁶⁾ (ILP¹⁴) 5章) のように確率的に割りきることや、非単調な規則を書けるようにして、作ったルールをなるべく生かすこと (ILP¹⁴) 7章 21章) も行われている。

5. おわりに

本稿で述べたように、理論変更の満たすべき理想的性質を考え、その実現方法として論理の逆計算を位置づけるという視点から、人工知能やデータベースの分野でこれまで別々に行われてきた研究を比較・分類できる。しかし、両者を統合する体系ができるまでにはいくつかの課題が残ってお

り、今後の研究が待たれる。

なお、帰納論理プログラミングの各オペレータは匿名 ftp で入手できる。comp. lang. prolog に定期的に投稿されるリソース・ガイドを参照されたい。反単一化は Quintus Prolog のライブラリにもある。

最後にコメントをくださった勝野裕文グループリーダーに感謝する。

参考文献

- 1) Allen, J., Hendler, J. and Tate, A. eds.: *Readings in Planning*, Morgan Kaufmann (1990).
- 2) Bond, A.H. and Gasser, L. eds.: *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann (1988).
- 3) Bonner, A.J. and Kifer, M.: *Transaction Logic Programming*, Tech. Report CSRI-270. Univ. of Toronto (1992).
- 4) Chellas, B.F.: *Modal Logic, an Introduction*, Cambridge Univ. Press (1980).
- 5) Fagin, R., Halpern, J.Y., Moses, Y. and Vardi, M.Y.: *Reasoning About Knowledge*, The MIT Press (1994).
- 6) Gärdenfors, P.: *Knowledge in Flux*, The MIT Press (1988).
- 7) Gärdenfors, P. ed.: *Belief Revision*, Cambridge Univ. Press (1992).
- 8) Ginsberg, M.L. ed.: *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann (1987).
- 9) Grahne, G., Mendelzon, A.O. and Reiter, R.: On the Semantics of Belief Revision Systems, in Proc. of the Fourth Conference on Theoretical Aspects of Reasoning About Knowledge, Morgan Kaufmann, pp. 132-142 (1992).
- 10) Katsuno, H. and Mendelzon, A.O.: Propositional Knowledge Base Revision and Minimal Change, *Artificial Intelligence* 52, pp. 263-294 (1991).
- 11) Moses, Y. and Kislev, O.: Knowledge-Oriented Programming, in Proc. of ACM Symposium on Principles of Distributed Computing (1993).
- 12) Muggleton, S.: Duce, an Oracle Based Approach to Constructive Induction, in Proc. of IJCAI, pp. 287-292 (1987).
- 13) Muggleton, S.: Inverting the Resolution Principle, in *Machine Intelligence* 12, Oxford Univ. Press (1991).
- 14) Muggleton, S. ed.: *Inductive Logic Programming*, Academic Press (1992).
- 15) Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann (1988).
- 16) Plotkin, G.D.: A Note on Inductive Generalisation, in *Machine Intelligence* 5, Elsevier North-Holland (1970).
- 17) Rao, A.S. and Foo, N.Y.: Formal Theories of Belief Revision, in Proc. of the First International Conference on Knowledge Representation and Reasoning, Morgan Kaufmann, pp. 369-380 (1989).
- 18) Shapiro, E.Y.: *Algorithmic Program Debugging*, The MIT Press (1983).
- 19) Shapiro, E.Y.: 知識の帰納的推論, 共立出版 (1986).
- 20) Shoham, Y.: Agent-Oriented Programming, *Artificial Intelligence* 60, pp. 51-92 (1993).
- 21) Winslett, M.: *Updating Logical Databases*, Cambridge Univ. Press (1990).
- 22) 有川節夫, 西野哲朗: 学習における計算論的アプローチ, 情報処理, Vol. 32, No. 3, pp. 217-225 (Mar. 1991).
- 23) 磯崎秀樹: 信念時間地図に基づく相互推論機構, マルチエージェントと協調計算 II, 近代科学社 (1993).
- 24) 井上克巳: アブダクションの原理, 人工知能学会誌 7, 1, pp. 48-59 (1992).
- 25) 川村 正: 帰納論理プログラミング, コンピュータ・ソフトウェア 10, 5, pp. 3-15 (1993).
- 26) 篠原 歩, 宮野 悟: PAC 学習—確率的で近似的に正しい学習—, 情報処理, Vol. 32, No. 3, pp. 257-263 (Mar. 1991).
- 27) 沼尾正行: 機械学習の主なパラダイムと現状, 人工知能学会誌 7, 1, pp. 6-9 (1992).
- 28) 古川康一: 論理と応用(3), 人工知能学会誌 7, 6, pp. 955-963 (1992).

(平成5年9月2日受付)



磯崎 秀樹 (正会員)

1960年生。1983年東京大学工学部計数工学科卒業。1986年同大学院修士課程修了。同年 NTT 入社。

1990年～1991年スタンフォード大学客員研究員。現在, NTT 基礎研究所情報科学研究部主任研究員。人工知能, 特にエージェントモデルや時間推論の研究に従事。共訳書「メンタルモデル」(産業図書), 著書「LaTeX 自由自在」(サイエンス社)。電子情報通信学会, 人工知能学会, ソフトウェア学会各会員。