

# ネットワーク環境を利用した仮説推論

山本 章博

## ■ 研究の目的

本研究は、ネットワークという高度な情報交換技術が出現した今日において、ネットワーク上に分散している知識をできる限り目標を特化しながら有効につなぎあわせて手元の問題を解くための仮説の生成と処理方式を探求することにあつた。仮説とは、現象を統一的に説明しうるように設けた仮定で検証が進んでいないものを指す。仮説を生成することは自然科学に限らず、日常生活においても意思を決定する際に必要である。情報化社会においては、判断材料として様々な情報を組み合わせて利用することになるが、必要な情報が全て得られるわけではなく、また、すべての情報に対して合理的な根拠を得られるわけもない。そのような場合には、得られた情報から仮説を生成して判断材料とすることになる。このように合理的な仮説の生成技術は情報化社会において基礎的な知識処理技術である。

本研究開始時点までは人工知能研究において仮説を生成する技術は、発想推論(推論アブダクション)と帰納推論(インダクション)とよばれ、それぞれ個別に研究されてきた。本研究では両者を仮説推論という統一的視点に置いた上で、これらの特性を充分区別した上で、以下の2点を中心に研究を進めてきた。

1. 発想的仮説推論と帰納的仮説推論をそれぞれネットワーク環境に適合させたシステム構成
2. 推論の立場からデータ・知識の共有と交換のための技術

なお、ネットワーク環境に適合させた場合には、発想推論という名前は必ずしも適切ではなくなるため、補完型仮説推論とよぶことにした。システム設計の方針としては、補完型仮説推論は利用者が直接仮説を生成するクライアント、帰納的仮説推論が、多くの利用者に仮説の生成の補助を行うサーバーとしての役割を持たせることとした。

## ■ 研究成果

本研究の成果は、ネットワーク環境に整合した補完型仮説推論と帰納的仮説推論をそれぞれ実現したことである。なお、両推論を統一的に説明しかつ区別するための仮説推論の数学的原理も並行して開発したが、その詳細は成果リストの文献[3, 7, 11, 12]を参照されたい。

### 1. 補完型仮説生成システム

#### 1.1. システム構成

本研究のもととなったアイデアである、「ネットワーク環境上に分散している知識をできる限り目標を特化した上で有効につなぎあわせて手元の問題を解く」に忠実なシステムの実現を行った。母体となる推論システムとして論理プログラム(Prolog)処理系を採用しているが、一般の節論理証明系への拡張も可能である。

もともと論理プログラムの実行は、

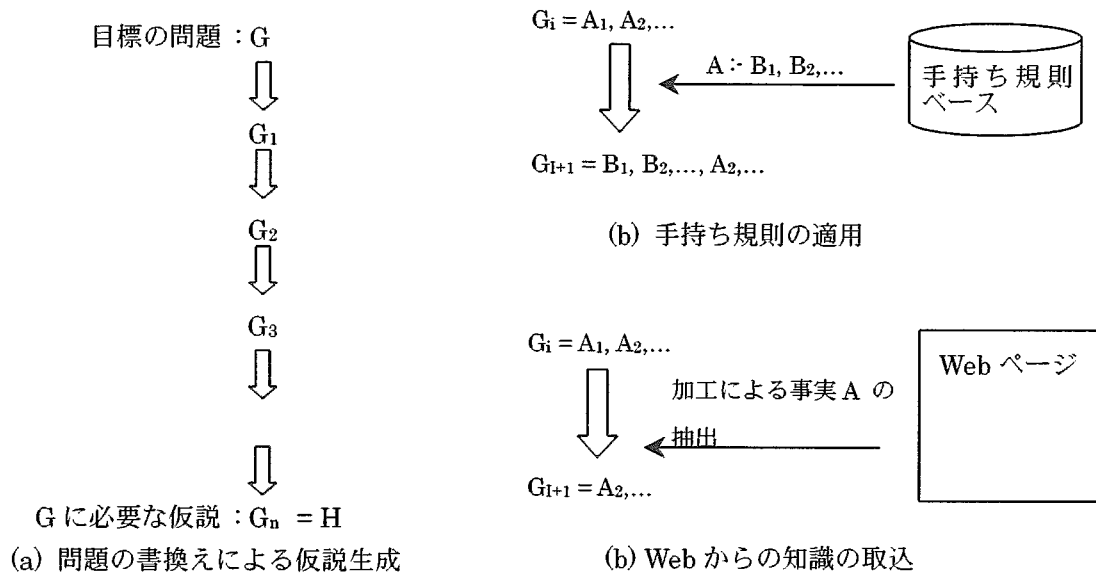


図1 補完的推論システムの動作概念

目標問題の連鎖  $G$  中にある個々の問題  $A_i$  に合致する事実  $A$  が無ければ、 $A_i$  を (できればより簡単な) 問題の連鎖に置換えることで  $G$  を新たな目標問題の連鎖  $G'$  に置換える。

この繰り返しと説明できる (図 1 (a))。したがって、問題  $A_i$  に合致する事実を検索することと、問題  $A$  を簡略された問題の連鎖  $G'$  に置換える規則を検索することが必要になる。本システムは、問題  $A_i$  を簡略化する規則の集合 (知識ベース) が手元にあるものと仮定し (図 1 (b))、 $A_i$  に合致する解が無ければ、あらかじめ決められた範囲で Web ページを検索して情報を取り出して何らかの加工を加えて  $A_i$  に合致する事実  $A$  を構成することを試みる。この 2 つの操作で解くことができない問題は、最終的に  $G$  を解くために必要な仮説として提示する。

従来の論理プログラム処理系はほとんど全て「知らない」問題は、その否定が成立することを前提にしている。しかしもともと「知らない」ことと「知らないことの否定を認めること」の間には大きなギャップがある。問題  $G$  が成立するかどうかを調べて、どうしても解けない問題が残ってしまったとき、そこに至る推論は正しかったはずである。したがって、 $G$  そのものを否定することは失う情報があまりにも多く、最も保守的であると考えられる。これに対して、発想推論 (アブダクション) は、どうしても解けない問題が残ったとき、それらを仮説として認めてしまう。本研究で作成したシステムはこの両者の中間的な立場と考えられる。

## 1.2 知識交換技術

補完型仮説推論構築には、検索した Web ページを加工して事実  $A$  を取り出す技術が必要になる。Web ページは HTML 文書や XML 文書であるから、それらを論理プログラムにおいて処理方式を開発しなければならない。

```

<TABLE>
  <TR><TD>a</TD><TD>1</TD></TR>
  <TR><TD>b</TD><TD>2</TD></TR>
  <TR><TD>c</TD><TD>3</TD></TR>
</TABLE>

```

(a)

\$1 =

```

<TABLE>
  <TR><TD>a</TD><TD>1</TD></TR>
</TABLE>

```

\$2 =

```

<TABLE>
  <TR><TD>b</TD><TD>2</TD></TR>
  <TR><TD>c</TD><TD>3</TD></TR>
</TABLE>

```

(b)

\$1 =

```

<TABLE>
  <TR><TD>a</TD><TD>1</TD></TR>
  <TR><TD>b</TD><TD>2</TD></TR>
</TABLE>

```

\$2 =

```

<TABLE>
  <TR><TD>c</TD><TD>3</TD></TR>
</TABLE>

```

(c)

図2 生垣論理プログラムに対する入力(a)と出力例(b)、(c)

論理プログラムの対象は項と呼ばれる木構造データであるから、タグ付きの文字列データである HTML/XML 文書をそのままの形で扱うことができない。考えられる方法は、次の2つである、

- 1) HTML/XML 文書を項として表現する。
- 2) HTML/XML 文書を直接扱えるように論理プログラムの処理系に変更を加える。

本研究では後者を選択した。その理由を説明するために、HTML/XML 文書のデータ構造としての側面を述べておく。

HTML/XML 文書は通常、半構造化データとよばれる。HTML/XML 文書そのものはタグを含む文字列データである。それが半構造化データとみなされるためには、少なくとも文書中の開始タグと終了タグのバランスがとれていなければならない。もし、バランスがとれていれば、HTML/XML 文書は文字列とポインタを用いたデータ構造で表現することができる。XML の世界ではこのデータ構造を DOM と呼び、HTML/XML 文書を(タグのバランスがとれていれば) DOM に変換するプログラムを XML プロセッサと呼ぶ。したがって、XML プロセッサを用いることを前提にすれば、HTML/XML 文書を論理プログラムで扱うことはすなわち DOM を扱うことになる。

DOM というデータ構造が項の一部になっていれば、論理プログラムにおける HTML/XML 文書の扱いにはほとんど問題が生じないが、残念なことに両者は全く別物である。論理プログラミングの世界では項は“木構造”であるといわれ、XML の世界ではしばしば DOM は“木構造”であるといわれるが、以下では両者を厳密に区別する。DOM は再帰的な木の列であり、木と文字列の双方の性質を兼ね備えたデー

タ構造である。列である以上、接続、分割、削除などの演算が必要である。論理プログラムにおいては、これらの演算を論理変数によって表現する。DOM に論理変数を導入したデータ構造は生垣 (hedge) または森 (forest) とよばれ、これが論理プログラムで扱う対象となる。

HTML/XML 文書すなわち生垣を論理プログラムで表す方法として、上述の 1) ではなく 2) を選択した理由は接続構造が自然に記述できるからである。項は接続構造を記述するには不向きなデータ構造であり、方針 1) を採用すると接続演算を表現する述語を用意しなければならない。しかし、方針 2) を採用すれば、接続演算がデータ構造の中で表現できることになる。生垣を直接扱う論理プログラムを生垣理プログラム (hedge logic program) とよぶ。生垣論理プログラムでは、図 2(a) に示すような一つの表として表示される文書を 2 つの表に分割するプログラムは

```
divide( <TABLE>$1$2</TABLE>, <TABLE>$1</TABLE>, <TABLE>$2</TABLE> ).
```

と記述する。ここで \$1、\$2、\$3 は論理変数である。実行結果の例として図 2(b)、(c) に示すような表の対が得られる。

## 2. 帰納的仮説生成システム

### 2.1. システム構成

帰納的推論とは、個々の具体的事実から一般的な命題ないし法則を導き出す推論を指す。帰納的推論の成果は十分な根拠を、機械学習やデータ・マイニングを含む知識発見は帰納的仮説生成となる。そこで仮説の生成と対象を関係データベースに限定した上で、帰納的仮説生成を行うサーバーとその結果を利用するクライアントを設計するための支援環境の実装を行った。クライアントの例として、複数のサーバーから得られた仮説を比較可能にするアプリケーションを採用した。このクライアントは、仮説がデータに対する“見方”でありかつ、一つのデータから得られる様々な“見方”を比較検討する必要がある、というデータ分析におけるの自然な要求から生れたものである。

システムの構成を図 3 に示す。図において矢印はデータの流れを示している。仮説生成サーバーは次の二つの機能を持つ。

- 1) クライアント設計支援環境によって定義された述語記号および事実と証拠の指定によって、データベースから得られるタプル集合をそのシステムに応じた入力フォーマットに変換する。
- 2) 変換系を用意している。さらに、そのシステムが論理プログラム以外の形で仮説を生成する場合には、その仮説を定義された述語記号を用いた論理プログラムに変換する。

クライアントは SQL 文を用いてデータベースからタプルが引き出し、さらに補足情報を用いて選択された各知識発見システムへ送る。その際データは各システムに適切な入力フォーマットに変換される。そして知識発見システムを実行し、得られた仮説を論理プログラムに変換してクライアントに返される。生成された論理プログラムはルールベースに蓄えられ、次の知識発見システムの背景知識として利用可能にする。現状で選択可能な知識発見システムは C4.5, Progol, FOIL である。

### 2.2 知識交換技術

帰納的仮説生成システムの実装においては、論理プログラムを知識のプロトコルとして扱うということを基本方針とした。すなわち、仮説生成サーバーが生成する仮説は論理式に変換されてクライアントに送られる。逆にいえば、どのような仮説生成システムであろうと、論理式に変換できさえすれば、

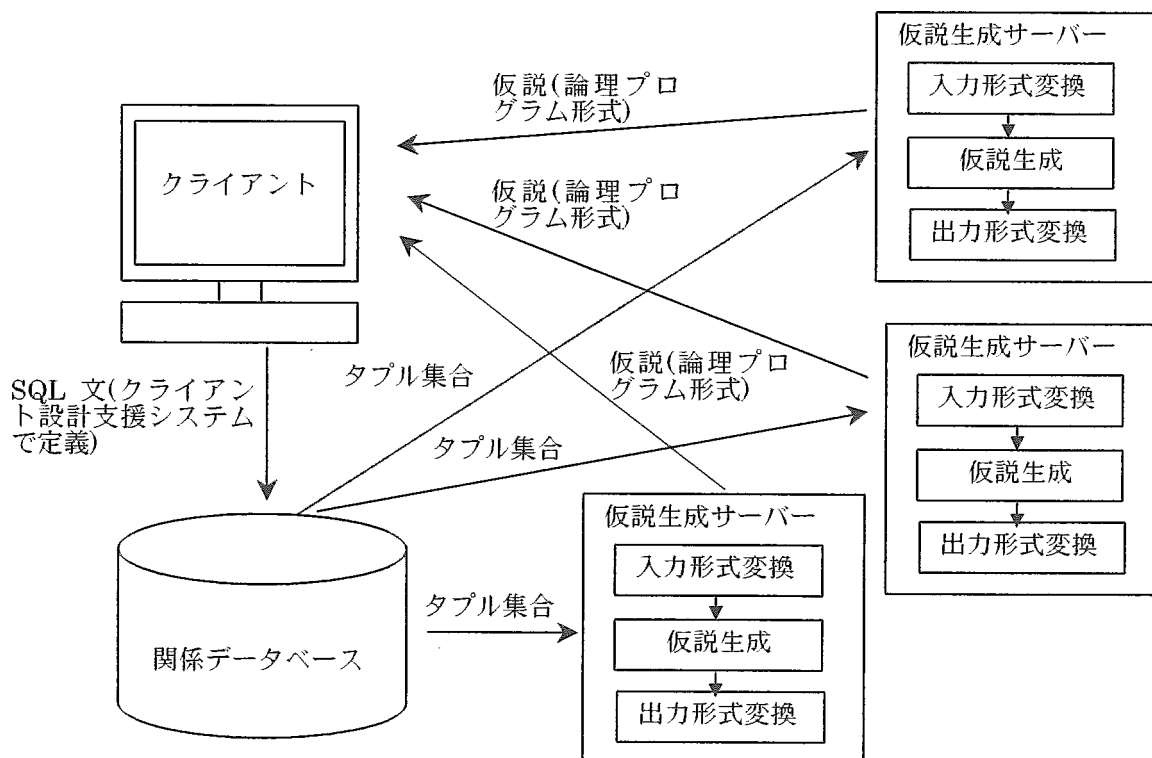


図3 帰納的仮説生成システムの概要

クライアントは仮説の利用が可能である。利用者は、もし必要ならば定理証明システムや論理プログラム処理系を利用して仮説の一般性を判定することも可能となる。関係データベースの用語を用いれば、3層スキーマ構造の中間層である概念スキーマ(論理構造)として発見の論理の枠組みを採用し、その下位層(内部スキーマ、物理構造)として個々の知識発見システムとその入力データを保持したデータベースを位置付ける、ということである。この基本方針は、仮説を比較可能にするアプリケーションを実現可能にしているという意味においてその有効性を確認することができた。

基本方針に沿うためにクライアント設計支援システムは以下の機能を提供している。

- 1) 論理プログラムにおける述語を统一的に定義するとともに、クライアント設計者がその中から必要な述語を選択する機能。
- 2) クライアントの設計者標準的な述語に別名をつける機能。
- 3) データベースからタプル集合を得るためのSQL文と発見の論理における述語記号との対応を与える機能を提供している。

ここで1)は、ネットワークを用いた分散環境における語彙統一の問題に対する一つの解である。支援システムにおいては、ウィザード形式の入力インターフェイスを使用することでパラメータ選択の履歴を用いた標準的な述語記号を決定することを可能にした。2)は標準的な述語記号の意味が不明瞭であったり、長さが非常に長くなるなど利用しづらいものになる状況に対する対処であったが、一つの機械学習・知識発見システムが属性の指定方法によって異なる結果を得ること、さらにはデータのセキュリティの観点から述語名を匿名にする必要がでてきたことへの対処でもある。

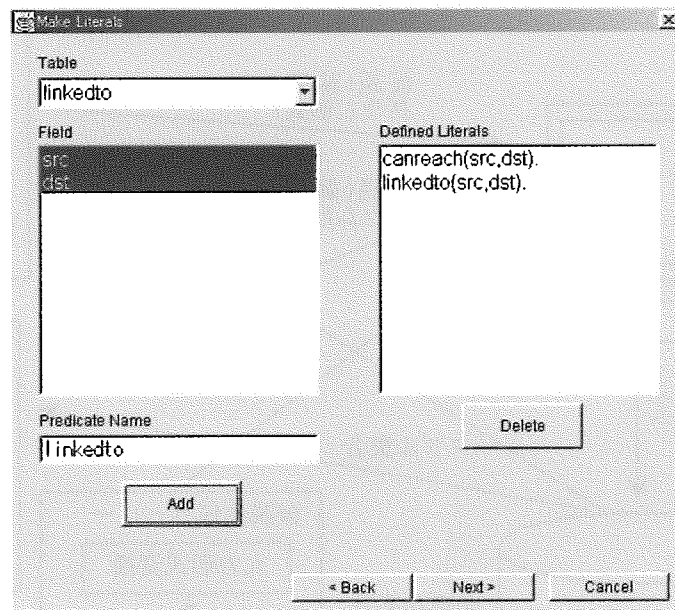


図4 クライアント設計支援システム実行の様子

クライアントの設計者はデータベースから知識発見の対象となるデータを選択する。次に仮説生成システムの特徴として属性指向、組指向のいずれかを選択する。次に仮説を生成させる知識発見システムを選択する。関係データベースのどの関係のどの属性を用いて述語を定義するのかをメニュー選択形式で決定する。属性指向を選択した場合はある制約に基づいて述語を生成する。組指向を選択した場合は利用者が任意の述語名を入力することで述語を生成する。その述語の定義に基づいて SQL 文が生成される。

図4にクライアント設計支援システムの入力インターフェースを実行している様子を示す。図は組指向を選択した場合の述語を生成する部分である。Table と書かれている下の選択枠にはデータベース内の関係の一覧が含まれており、利用者はその中から関係を選択する。関係を選択すると、その関係に含まれる属性の一覧が Field と書かれている下の枠に表示される。利用者は属性の一覧から属性を選択し、Predicate Name と書かれている下の入力枠に任意の述語名を入力して Add ボタンを押すことで述語が生成される。Defined Literals と書かれている下の枠にはには生成した述語が表示される。なお、Defined Literals の枠に表示されている canreach(src,dst)は基礎アトムではなく、標準的述語記号 canreachsrcdst を()と、を用いて見やすくしたものである。

帰納的仮説生成システムの原点は、30年前にPlotkinによって提案された発見の論理にある。当時と現在では計算機環境は全く異なっている。本システムはネットワークという現代的な環境が与えられたもとの、さらに様々な仮説生成技術が開発されているという状況のもとで、発見の論理を具現する方法を模索した結果とみなすことができる。発見の論理はCarnapらの構築した科学哲学の論理による具現を目指したものであった。今日の人工知能的な視点からは、その内容は決して新しいものではない。それでも本研究の根底に採用したのは、発見の論理の中に、ほぼ同時期に独立に発表された関係データベースに関する基本的な考え方とそこからの仮説発見の原理を見出すことができたからである。さらに、関係データベース理論において今日実体や関連とよばれる概念の芽をも知識発見と関連

した形で見出すことができる。すなわち、発見の論理は古いが非常に普遍的であり、それゆえ、本システムの基盤として採用するに足るものであった。

## ■ 今後の展開

この3年間の研究を振り返ると、ネットワーク環境は推論に知識共有という領域への接点の開拓を迫っていることを改めて感じさせるものであった。本研究では、仮説推論を具体的目標領域にしたものの、そこで開発した知識共有方法は広く推論システムが知識共有に接点を持つための基本技術であると考えている。その意味で、本研究の成果は十分あったと考えている。

研究成果である各システムを発展させる方向を述べると、まず補完型仮説生成システムについては、推論方式としては連続値データのより柔軟な扱えるように、さらに生垣論理プログラミングはページ・ラッパーの基礎技術へと展開させる。特に生垣論理プログラミングは適切な実行システムを作成すれば、DTD や Relax などの機能であるデータの定義や XSLT の機能であるデータ変換に加えて、双方向変換が単一の言語で記述可能になると考えられる。帰納的仮説生成システムについては、現在では構造化離散値データしか扱うことができないところを、連続値データの扱い、半構造化データ・文字列データの扱いを可能にし、適切かつバリエーションのあるクライアントを作成すること、さらには仮説生成システムを公開するための基礎を明らかにする。

最後に、本研究を通して得た新たな認識について述べておく。本研究を進めていくうちに、ネットワーク環境下において情報を性悪的に捉えることが研究の根底になっていることが次第に明らかになってきた。具体的に WWW を例に考えれば、確かに WWW による情報公開は速報性に優れ、検索が容易であり、新たな情報交換の場として認識されている。しかし、誰にも審査されず、客観的に評価されていなくても情報公開できることも事実である。そこで、確実な知識が極一部に限られ、そこから導出される帰結が全て仮説であるような世界で、不確実な推論の接続が全体として確実な方向へ向くような新たな推論原理が必要になる。本研究を進め、また関連研究を調査していく中で、現在では仮説推論と演繹推論の中間種として（古生物の始祖鳥になぞらえて）「始祖学習」とも呼ぶべき推論の存在に行きつき、仮説推論と演繹推論が連続しているという認識を強く持つに至った。この認識を元に、今後本研究を不確実性推論システムの分散構成へと展開し、次世代の推論の中心に位置付けたい。

## ■ 成果リスト

- [1] Yamamoto, A. : Hypothesis Construction and Network, Joint Seminar on Theories and Applications of Discovery Science, The University of New South Wales, Sydney (1999).
- [2] Yamamoto, A. : Relative Least Generalization Revisited, Second Joint Seminar on Theories and Applications of Discovery Science, The University of New South Wales, Sydney (2000).
- [3] 山本章博 : Herbrand の定理に基づく仮説生成、2000 年度人工知能学会全国大会講演論文集、287-290(2000).
- [4] Yamamoto, A. : New Conditions for the Existence of a Least Generalization under Relative Subsumption, Proceedings of the 10th International Workshop on Inductive Logic Programming , (Lecture Notes in Artificial Intelligence 1866), 253-264, Springer (2000).

- [5] 山本章博 : XML 文書を扱う論理プログラミング処理系の試作、人工知能学会知識ベース研究会資料、SIG-KBS-A003, 67-68(2000).
- [6] 山本章博、伊藤公人、石野 明 : 生垣論理プログラミングによる情報の抽出と変換、2001 年度人工知能学会全国大会講演論文集、CD-ROM(2001).
- [7] Yamamoto, A.: Hypothesis Finding based on Upward Refinement of Residue Hypotheses -extended abstract-, In Proceedings of the Workshop on Logic and Learning affiliated with LICS 2001 (2001).
- [8] 菊池敏幸、山本章博 : 様々な知識発見システムを比較可能にする環境の構築、人工知能学会知識ベース研究会資料、SIG-KBS-A101, 49-55 (2001).
- [9] Yamamoto, A., Ishino, A., Ito, K. and Arimura, H.: Modelling Semi-structured Documents with Hedges for Deduction and Induction, Proceedings of the 11th International Workshop on Inductive Logic Programming, (Lecture Notes in Artificial Intelligence 2157), 240-247, Springer (2001).
- [10] Yamamoto, A. and Fronhoefer, B. : Hypothesis Finding by Inverted Deduction of Residue Hypotheses, Technical University of Darmstadt, Darmstadt (2001).
- [11] Yamamoto, A and Fronfoefer, B : Finding Hypotheses by Generalizing Residue Hypotheses, the Work in Progress session in the 11th International Workshop on Inductive Logic Programming, Strasbourg (2001).
- [12] Yamamoto, A.: Hypothesis Finding based on Upward Refinement of Residue Hypotheses, to appear in Theoretical Computer Science.
- [13] 菊池 敏幸、山本 章博 : 様々な発見の論理を統一的に利用可能な環境の構築、人工知能学会誌 (投稿中)。