

TM-0866

提案誘導型知識獲得支援システム

滝 寛和, 寺崎 智

February, 1990

©1990. ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

(財) 新世代コンピュータ技術開発機構
第 5 研究室

滝 寛和、寺崎 智

1. まえがき

知識ベース構築に望まれる機能は、適用能力の高い一般的知識の獲得である。一般化された知識を専門家（知識源）から抽出する対話型知識獲得の機能に、事例から一般的な知識を生成する帰納推論の機能を融合させることで、高機能の知識獲得システムの開発を目指す。現在の知識獲得支援システムと帰納推論システムでは、伴に、長所と短所がある。本研究では、各の短所を補えるような知識獲得能力の高いシステム開発を目的とする。帰納推論の基本機能は、事例から一般的知識を生成することであるが、概念形成問題での短所は、帰納的に作られる概念の候補（仮説）が非常に沢山生成されることと、計算量が非常に多いことである。帰納推論の利用においては、計算量と概念の候補を減らすために、生成される概念の形式（言語）を制限する必要がある。

対話的知識獲得の問題点は、専門家が一般的な知識を意識していない場合には、非常に個別的な事例や不十分な知識しか抽出出来ないことである。知識獲得支援システム EPSILON/One [滝 87] [椿 88] [大崎 88] には、オペレーションのエバリュエータの抽出において、具体例から知識の候補を推定する必要があることが分かった [椿 89]。このエバリュエータの推定に帰納推論を適用する場合には、知識獲得支援システム EPSILON/One での知識表現である専門家モデルの特徴を旨く利用することとした。専門家モデルの構成要素であるオペレーションは、タイプを持っており、獲得すべき知識の形式を強く制限するのに利用される。この形式の制限を帰納推論で求める知識の形式の制限とすることで、効率的な概念形成が行える可能性がある。

2. 提案誘導型知識獲得の概要

知識獲得支援システムの基本機能は、知識の抽出、知識の整理／体系化、知識の不備（冗長、不足、矛盾）の検出と解消、知識表現の変換である。従来、知識の抽出に際しては、適用範囲の広い一般的な知識を専門家（知識源）から心理学的な連想方法により獲得していた。ほとんどの知識抽出方法は、専門家から得た僅かな情報を利用して、その情報を元のまま、または、異なる表現で専門家に再提示して、新たな知識を連想させるものであった。ETS [Boose 86]、AQUINAS [Boose 87] で利用されている Personal Construct Theory や知識獲得支援システム EPSILON/One のプリポスト法 [滝 87] がその代表的な例である。

専門家が、ある問題解決に知識を十分に意識（知識を整理して確認できる状態におくこと）していない状況では、その知識を表現する（述べる）ことはむずかしい。一方、専門家にとって、ある知識が自分の持つ知識と同じであるか異なるかを判別することは、前者と比べてむずかしくはない。つまり、知識獲得支援システムの知識の抽出の機能の向上させるには、専門家に対して、知識獲得支援システムが知識の候補を提案できることが望ましい。

知識の候補を提案できる機能の具体的な例としては、専門家から与えられた情報を利用して、知識獲得支援システムがその情報に含まれる以上のものを知識の仮説として生成できる能力をいう。この仮説の生成に帰納推論を用いることで、『事例の抽出』、『一般的知識の生成』、『専門家による修正』を知識抽出機能として持つ知識獲得支援システムが開発できる。

専門家と知識獲得支援システム間の情報伝達をみると、システムの提案した知識に対して、『別候補の要求』、『修正した基準でのオペレーションの実行による基準の検証』、『事例の追加』、『専門家が入力した評価基準の例による自動修正』が考えられる。この内、最後の『専門家が入力した評価基準の例による自動修正』は、専門

家が進んで、帰納されるべき言語形式を定義しているので、数値演算(変換)オペレーションの推定の可能性を与えると伴に、今後の重要な研究課題の1つと言える。

専門家と知識獲得支援システムとの対話の関係は、図1のようになる。この知識獲得の枠組は、システムが知識の候補を生成し提案することから提案主導型知識獲得支援システムと呼ぶことにする。

3. 帰納推論を用いたオペレーション推定

3.1 オペレーションタイプに対応した帰納的な知識推定アルゴリズム

知識獲得支援システム EPSILON/One の専門家モデルで用意されているオペレーションタイプは、『分類』、『選択』、『入力』、『出力』、『順序付け』、『組み合わせ』と『変換』である。

例からの帰納的一般化を行うには、『求める概念を表現する言語』、『概念についての正例/負例』、『概念の一般化空間を規定する知識(バイアス)』と『帰納アルゴリズム』が必要であるが、オペレーション推定で準備しているこれらの基本的情報を次に示す。

『求める概念を表現する言語』: オペレーションタイプ毎に決められたオペレーション実行の為の評価基準の表現。単一属性による評価基準を基本とする。

『概念についての正例/負例』: オペレーションの入出力の要素の例。

『概念の一般化空間を規定する知識(バイアス)』: オペレーションタイプの評価基準毎の評価の一般化方法。

『帰納アルゴリズム』: オペレーションタイプの評価基準に合う推定アルゴリズム。

このように、オペレーションのタイプを活用して、言語の制限、例の扱い、バイアスの設定を行い帰納的一般化から得られる候補を制限し、評価基準の推定を効率の良いものとしている[Taki-1 89]。

以下、オペレーション型知識の推定方法として、『属性値の入れ替え』(変換)オペレーション、『順序付け』オペレーション、『選択』オペレーションと『分類』オペレーションについて、論じる。ただし、『入力』と『出力』のオペレーション及び、『要素分解』(変換)オペレーションでは、専門家が、注目している属性が明確なので知識の推定は行っていない。『組み合わせ』オペレーションは、2つの要素を組み合わせる1つの要素を作るアルゴリズムを固定しており、評価基準は不要なので、知識の推定は行わない。また、『数値演算』(変換)オペレーションは、かなり数式を限定しないと推定の計算量が多いので、推定アルゴリズムは、未開発となっている。

推定対象となっているオペレーション毎に次の項目に従って説明する。

- ・オペレーションの実行アルゴリズム
- ・オペレーションの実行に必要な評価基準(獲得対象と推定対象)
- ・例の扱い
- ・評価基準の推定アルゴリズム

(1) 選択と分類知識の推定

選択オペレーションは、与えられた要素の集合から選択基準に合う要素を選ぶ。

分類オペレーションは、与えられた要素の集合を分類基準により、複数の集合に分割する。分類オペレーションは、選択オペレーションを複数個組み合わせたものと言える。

(a) 選択アルゴリズム

ある要素が与えられたとき、その要素の属性と属性値が、選択基準を満たすとき選択される。選択基準は、ある属性の値の取り得る集合（または、複数の属性についての可能な属性集合の連言(conjunction)）で表される。選択基準を満たすとは、評価基準が規定している属性に注目して、評価される要素のその属性値が、選択基準の可能な属性値の集合に含まれることをいう。

(b) 選択基準

単一属性から成る選択基準は、次のように限定する。

(属性名、属性値の集合)

複数属性から成る選択基準は、単一属性の選択基準の連言形式とする。

(属性名1、属性値の集合) \cap (属性名2、属性値の集合) \cap ...

注) EPSILON/One のインタビューでは、選択基準に属性値の集合だけでなく、属性値の区間や不等号(数値の場合)と、選言(disjunction)形式も獲得しているが知識推定では、帰納推論の効率を考慮して、知識の形式を連言に固定している。

(c) 例の扱い

例としては、複数の要素をオペレーションの入力および、出力要素グループとして与える。選択オペレーションで、選択された要素は、選択基準の正例と言え、また、選択されなかった要素は、負例と言え。要素は、オペレーションとは、独立の知識としてグローバルに定義できる。また、要素は、属性と1つの属性に対して1つの属性値の対の連言で表わされるものとする。

(例題)

入力要素グループと出力要素グループを次のように与えると

入力要素グループ = {e1, e2, ..., ei, ei+1, ..., en}

出力要素グループ = {e1, e2, ..., ei} (ただし、ej (j=1~n)は、要素を表す)

正例は、出力要素グループそのものであり、{e1, e2, ..., ei}となる。また、負例は、入力要素グループから出力要素グループの要素を除いたもので、{ei+1, ..., en}となる。

(d) 選択基準の推定アルゴリズム

Step1: まず、正例、負例を問わず、例すべてに含まれる属性aを求める。

例: {a1, a2, ..., am} (ただし、aj (j=1~m)は、属性を表す)

Step2: 正例/負例毎に、各要素に対して、上記の属性集合の各属性に対して値を求める、要素がある属性を持たない場合には、その属性値を undefined とする。

Step3: 求める選択基準が単一属性であると仮定して、属性単位で正例/負例の可能な属性値の集合を求める。これは、属性間の従属性に関する一般化である。

(例) 要素を ek(k=1~n)、属性を aj(j=1~m)、属性値を vjk(k=1~n, j=1~m)とすると正例/負例毎に要素と属性に関するマトリックスができる。

正例:	a1	a2	a3	...	am
e1 :	v11	v21	v31	...	vm1
e2 :	v12	v22	v32	...	vm2
		:			
ei :	v1i	v2i	v3i	...	vmi

負例:

$$\begin{array}{l}
e_{i+1}: v_{1i+1} v_{2i+1} v_{3i+1} \dots v_{mi+1} \\
e_{i+2}: v_{1i+2} v_{2i+2} v_{3i+2} \dots v_{mi+2} \\
\vdots \\
e_n : v_{1n} v_{2n} v_{3n} \dots v_{mn}
\end{array}$$

正例と負例について、属性ごとに属性値の集合を作る。属性aa に関する正例/負例の属性値集合をそれぞれ P(aa)、N(aa)とする。

$$\begin{array}{l}
P(a_1) = \{v_{11}, v_{12}, \dots, v_{1i}\} \\
P(a_2) = \{v_{21}, v_{22}, \dots, v_{2i}\} \\
\vdots \\
P(a_m) = \{v_{m1}, v_{m2}, \dots, v_{mi}\} \\
\\
N(a_1) = \{v_{1i+1}, v_{1i+2}, \dots, v_{1n}\} \\
N(a_2) = \{v_{2i+1}, v_{2i+2}, \dots, v_{2n}\} \\
\vdots \\
N(a_m) = \{v_{mi+1}, v_{mi+2}, \dots, v_{mn}\}
\end{array}$$

Step4: 正例と負例で積集合を持つ属性は、単一属性では選択に利用できないので除く。
if $P(a_j) \cap N(a_j) = \emptyset$ (空集合) then 属性 a_j を選択する
else 属性 a_j は選択しない, ($j = 1 \sim m$)

選ばれた属性の集合が、{aa, ab, ac}であれば、P(aa)、P(ab)または、P(ac)が選択基準となる。

注) $P(a_j) \cap N(a_j) = \emptyset$ から $\neg N(aa) \wedge \neg N(ab) \wedge \neg N(ac)$ も選択基準である。ただし、 $\neg N(aa)$ は、属性aa が、属性値として、N(aa) の要素を取らないことである。

Step5: 求められた選択基準を提示する。複数ある場合には、順に提示する。

上記の例では、P(aa)、P(ab)、P(ac)、 $\neg N(aa) \wedge \neg N(ab) \wedge \neg N(ac)$ を順に提示する。なお、この例の基準の一般性/特殊性の関係は、図 2 のVersion Graphで表される[Genesereth 86]。

(e) 推定された選択基準の提示

選択基準としては、選ばれた属性が属性aa 1つの場合には、P(aa)を専門家(知識源)に提示する。選ばれた属性が複数のときは、順に、その属性に対する正例の属性値集合を提示する。

(f) 選択基準の一般化の誘導

属性値の集合を一般化することで、選択幅を広げられることを示す。

(例) $\{1, 2, 3, 4, 5\} ==> \{>= 1, <= 5\}$
==> 整数

$\{\text{青りんご}, \text{あかいりんご}\} ==> \{\text{りんご}\}$

さらに、正例か負例か分からない場合についての説明の提示により一般化できることを専門家に示す。

例: $P(aa) = \{1, 2, 3\}$
 $N(aa) = \{5, 6, 7\}$

属性aa の属性値が、4 のとき、正例か負例か不明である。

選択基準が、P(aa)なら、この要素は、負例として扱われる。

また、選択基準が、 $\neg N(aa)$ なら、この要素は、正例として扱われる。

つまり、 $\neg N(aa)$ を選ぶ方が選択幅は広がり、より一般的な選択基準となる。

$P(a_j) \cap N(a_j) = \emptyset$ となる属性 a_j がない場合は、属性間を独立にすることは、

失敗となり、複数の属性の関係を考慮した判定基準が必要となる。

(2) 順序付け知識の推定

(a) 順序付けアルゴリズム

与えられたすべての要素について、ある属性に注目して、その属性値の全順序基準に従って、要素を並べ替える。この順序は、オペレーション（選択や出力オペレーションなど）での要素の評価順序を意味する。

(b) 順序付け基準

単一属性から成る順序付け基準は、次のように限定する。

（属性名、属性値リスト）または、（属性名、昇順）または、（属性名、降順）属性値リストの記号順序が、上位から下位への順序を表すものとする。また、昇順と降順は、属性値が数値であるものを表す。

(c) 例の扱い

例としては、複数の要素をオペレーションの入力および、出力要素グループとして与える。順序付けオペレーションでは、出力要素グループの要素の順序が順序基準の正例と言える、また、入力要素グループでの2要素の順序関係が、出力要素グループでの異なる順序となる2要素間の順序は、負例と言える。

(例題)

入力要素グループと出力要素グループを次のように与えると

入力要素グループ = {e1, e2, e3, e4}

出力要素グループ = {e1, e3, e4, e2}

a よりも b を優先する関係を \succ で表すと、正例は、 $e1 \succ e3 \succ e4 \succ e2$ である。また、負例は、 $e2 \succ e3$ と $e2 \succ e4$ となる。

(d) 順序付け基準推定アルゴリズム

- Step1: 2つの要素間で、入力要素グループと出力要素グループで、順序が異なるものを探す。
- Step2: 属性値に undefined 含まない属性を1つを選び、出力要素グループの要素の対について、その属性値の順序を決める。これを順序基準と呼ぶ。
推定対象となる属性がない場合には、推定を終了する。
- Step3: 順序の関係に、矛盾がないかを調べる。矛盾があれば、属性を変更して、Step2 を実行する。順序の関係に矛盾があるとは、 $a \succ b$ かつ $b \succ a$ の関係が同じに成立することをいう。
- Step4: 負例から導かれる順序関係を2要素間の関係について求める。
- Step5: 負例から導かれた順序関係が求めた順序基準で成立するときは、属性を変更して、Step2 を実行する。
- Step6: 属性値が、数値のときは、昇順か、降順かを調べる。その他の関係（偶数を優先するなど）であれば、数値であっても、文字と同じように扱う。
- Step7: このような条件を満足する属性が、複数ある場合には、順に提示する。

(3) 属性値の入れ替え

(a) 属性値入れ替えアルゴリズム

ある要素が与えられたとき、ある1つの属性に注目して、入れ替え基準に従って、元の属性値に対応する別の値に入れ替える。入れ替え基準は、変換テーブル形式で表され、元の値と変換後の値の対からなる集合である。

(b) 属性値入れ替え基準

単一属性から成る入れ替え基準（規則）は、次のように限定する。
（属性名、（元の属性値、変換後の属性値）の集合）

(c) 例の扱い

例としては、複数の要素をオペレーションの入力および、出力要素グループとして与える。属性値の入れ替えオペレーションで、入力と出力で値の異なる要素は正例と言える、また、値の変わらない要素は負例と見ることができる。

(d) 属性値入れ替え基準推定アルゴリズム

- Step1: 値が変更されている属性を1つ探す。これを注目属性とする。
Step2: 正例について、注目属性の元の属性値と変更後の属性値の対を作る。
Step3: 元の属性値が同じで、変更後の属性値が異なるものを探す。
この変更後の属性値の候補が複数あるものを不明確な基準とする
Step4: 負例について、注目属性の属性値を取り出す。
Step5: Step2 で得られた属性値の対の中で、元の属性値がStep4 で得られた属性値と等しい属性値の対を不正確な基準とする。
Step6: 不明確でなく、不正確でない属性値の対を属性値入れ替え基準として提示する。
Step7: さらに、不明確な基準と不正確な基準を提示する。

4. おわりに

知識獲得支援方法として、帰納推論による知識を生成し、その知識を専門家に提示することで、知識を抽出する方法を研究した。この研究は、工学的な知識獲得方法と数学的／論理的な学習方法の1つの適切な融合を示した例といえる。今後は、この方法を発展させて、設計問題のための知識獲得（制約緩和知識の提案と獲得など）に利用していく予定である。また、現在、帰納推論アルゴリズムは、より効率を向上させるために、並列論理型言語KL1 で並列化を推進中である。

謝辞

本研究に協力頂いたICOT 第5研究室 研究員、ならびに、生駒室長に感謝いたします。また、本研究の機会を与えて頂いた ICOT 淵所長に深く感謝いたします。

参考文献

- [滝 87] 滝、椿、岩下：知識獲得支援システム（EPSILON）における専門家モデル、情報処理学会、知識工学と人工知能研究会報告 52-4、1987年
[椿 88] 椿、滝、大崎：知識獲得支援システム EPSILON/One(1)、第8回知識工学シンポジウム資料、計測自動制御学会、1988
[大崎 88] 大崎、滝、椿：知識獲得支援システム EPSILON/One(2)、第8回知識工学シンポジウム資料、計測自動制御学会、1988
[Taki-1 89] H.Taki and Y.Fujii: Operation Presumption: Knowledge Acquisition by Induction, Proceedings of Third European Workshop on Knowledge Acquisition for Knowledge-Based Systems, pp 34-48, 1989
[Taki-2 89] H.Taki: Knowledge Acquisition by Abductive and Inductive Explanation, Proceedings of 4th Knowledge Acquisition for Knowledge-Based Systems Workshop, pp 34-1 - 34-19, 1989
[永井 89] 永井、滝、寺崎、横山、井上：設計問題向けツール・アーキテクチャ、

人工知能学会誌, Vol.4, No.3, pp 61-67, 1989年 5月号

- [滝 89] 滝: 知識獲得支援システム EPSILON/One, 第7回 第5世代コンピュータに関するシンポジウム 予稿集, pp 17-18, 1989
- [椿 89] 椿、滝、大崎: 知識獲得支援システム EPSILON/One, 人工知能学会全国大会 (第3回) 論文集, Vol.2, 12-5, pp 719- 722, 1989
- [Boose 86] J. H. Boose: A Knowledge Acquisition Program for Expert Systems Based on Personal Construct Psychology, J. of Man-Machine Studies, Vol. 23, pp 495-525, 1986
- [Boose 87] J. H. Boose, et al.: Expertise Transfer and Complex Problem Using AQUINAS as A Knowledge-Acquisition Workbench for Knowledge-Based Systems, J. of Man-Machine Studies, Vol.26, No.1, pp3-28, 1987
- [Genesereth 86] M. Genesereth and N. Nilsson: Logical Foundations of Artificial Intelligence, Los Altos, CA: Morgan-Kaufman