

知識獲得機能を持つ選択型問題解決ツール「DT」

1 Q-1

古関 義幸、田中 みどり、中莖 洋一郎
日本電気(株) C&C システム研究所

1 はじめに

1970年代にエキスパートシステム(以下、ES)技術が開発されて以来、多くのES構築ツールが開発されているが、当初の予想程は普及していない。これは、従来のES構築ツールでは、知識エンジニアが業務専門家からインタビューを通して知識を引出し、ルールやフレームなどの知識表現形式に整理して記述しなければならなかったからである。本報告では、業務専門家が自ら、知識エンジニアを介さずにESを構築できることを目的として、表形式の知識表現をベースにして対話型の知識獲得機能を実現した分類・選択型問題向けES構築ツール「DT」について述べる。

2 対象とする問題

本システムは次のように定義される「分類・選択型問題」を対象とする。まず、ある概念の集合 $X = \{x_1, x_2, \dots, x_m\}$ と属性の集合 $A = \{a_1, a_2, \dots, a_n\}$ を考える。ここで、各属性 $a_j (j = 1, 2, \dots, n)$ の定義域は、有限の多値集合であるとする。この概念集合に対する分類知識は、各概念ごとに、各属性のブール関数の集合として定義できるとする。

たとえば、故障診断問題の場合、 X は考えられる全ての原因の集合であり、 A は観測され得る全ての症状あるいはテスト結果の集合である。工具選択問題の場合、 X は使用可能な工具の集合であり、 A はこれらの工具を選択するときを考慮すべき条件の集合である。ここで、この「分類・選択型問題」の目的は、与えられた属性の集合に対して、これを満足する X の部分集合を求めることである。

3 知識表現

従来のシステムのほとんどがルール形式の階層的知識表現を利用しているのに対して、本システムは、概念の関数を、DNF (Disjunctive Normal Form, 積和標準形) で表す表形式の知識表現を採用している。この表現形式によれば、属性の選言として表されるような概念も含めて一般的にどのような概念も表現可能である。概念関数を構成する各選言をキューブ、また、概念関数を表す複数のキューブの集合をカバーと呼ぶ。各キューブは、パートと呼ぶ各属性に対応する部分の集合からなる。

ここで、例として、ある仮想的な工場での、グラインダの選択問題を考える。この工場において、研磨をする際の対象材料の性質と要求品質の組合せから、その研磨を行なうための最適なグラインダを選択したい。2種類のグラインダ(Grinder-HとGrinder-L)のうち、高級なGrinder-Hは、硬い材料でも軟らかい材料でも研磨できる上に、その仕上がりがよい。一方、廉価なGrinder-Lは、軟らかい材料しか研磨できない上に、その仕上がりがあまりよくない。この工場では、製品コストを最小限にするために、なるべくGrinder-Lを使用し、どうしても必要な時のみGrinder-Hを使用することになっている。例えば、硬い材料を使って、要求品質が低い製品を研磨する時には、Grinder-Hを使用する。

このグラインダの選択に関する知識は表1に示す表で表される。ここで、キューブとは各行に相当し、同一の概念を構

成するキューブの集合がカバーである。例えば、「Grinder-H」という概念のカバーは二つのキューブから構成される。ひとつのキューブにおいて、各パートは○のついている属性値の論理和と解釈される。それらのパートの論理積としてキューブは解釈される。カバーは、それを構成するキューブの論理和として解釈される。不明あるいは無関係の属性のパートにはすべて○がつけられる。○が一つも無いパートが一つでもあるキューブは空キューブ ϕ である。

	材料		要求品質	
	硬い	軟らかい	高い	低い
Grinder-H	○	×	○	○
Grinder-H	×	○	○	×
Grinder-L	×	○	×	○

表1: 例題の表形式知識表現

表形式知識表現の長所は、(1)その内容が一見して理解しやすいこと、(2)修正が容易であること、(3)質問のうちある属性が不明であったとしても、部分的にマッチする概念を見つけていることができること、(4)質問の順序などの制御的な知識を排除していることである。その短所は、表現する知識によっては、表が疎になって大きくなる可能性があることである。このような場合には、複数の表に分割して表現することで対処が可能である。

本システムでは、否定カバーと呼ぶカバーを用いて、同じ概念関数を異なる形で記述できる。否定カバーに対して、これまでの記述で単にカバーと呼んでいたものを肯定カバーと呼ぶ。肯定カバー PC と否定カバー NC を用いて、ある概念 C は、 $C = PC \wedge \overline{NC}$ と表される。否定カバーを構成するキューブを否定キューブと呼ぶ。

前述の例題で、Grinder-Hを選択する条件は、「材料が軟らかく、要求品質が低い時以外は、常にGrinder-Hを使用する」と解釈することができる。この記述は、否定カバーを使ってそのまま表2のように表現できる。ここで、記号 $\#$ は、このキューブが否定キューブであることを意味する。

	材料		要求品質	
	硬い	軟らかい	高い	低い
Grinder-H	○	○	○	○
#Grinder-H	×	○	×	○
Grinder-L	×	○	×	○

表2: 否定キューブを使った例題の表現

4 推論方式

ある概念 C が、肯定カバー PC と、否定カバー NC で表現されており、質問として与えられた属性の集合が質問キューブ QC で表せる時、この概念 C が質問キューブ QC の解答候補となるのは $(PC \wedge \overline{NC}) \wedge QC \neq \phi$ が成り立つときである。候補集合の計算方法や効果的な質問の選択方式の詳細については、[Koseki91]に述べてあるので、本報告ではその概要についてのみ述べる。本システムは、全ての概念を初期候補集合として、質問制御機構が順次選択する質問を利用者に行うことにより質問キューブ QC を狭めながら、候補となる概念を絞っていく。このサイクルを終了条件(候補がひとつに絞られたときなど)が成り立つまで続ける。

質問する属性は、その時点での候補集合に対して、それを

DT: A Classification Problem Solving Tool with Knowledge Acquisition Function

Yoshiyuki KOSEKI, Midori TANAKA, Yoichiro NAKAKUKI
C&C Systems Research Labs., NEC Corporation

分類するのに自然でかつ効果的な属性がシステムによって選択される。これを行う質問制御機構は、自然な質問の流れを実現するために、質問依存条件、質問禁止条件、質問グループ、概念階層の制約条件を優先する。そして、質問の数を最小限にするために、制約条件が無い場合には、単位質問コスト当たりの情報量の最も多い属性を選択し、質問する。

この方法はエントロピー最小化法とも呼ばれ、概念候補集合の生起確率分布を基にその状態のエントロピーをもとめ、このエントロピーをなるべく小さくするような属性を選択する方法である。エントロピーの計算方法として、決定木学習アルゴリズムであるID3 [Quinlan86]と同様のシャノンのエントロピーを利用している。

5 対話型知識獲得

対話型知識獲得は、システムの出した結論に誤りがある場合に、利用者がその誤りをメニュー形式で指摘することにより知識ベースの修正を行う機能である。誤りの指摘の仕方として、(1a) 解答に本来含まれるべきでない答えが解答されている、(1b) 本来質問されるべき質問が行われないために、含まれるべきでない答えが解答されている、(2) 含まれるべき答えが解答されない、の3種類がある。これらの誤りが指摘されると、システムは、概念の詳細化、一般化、差別化のいずれかの方法を用いて、知識ベースを修正する。例えば、(1a)の場合は、その誤りの答えの概念関数を詳細化することで、(1b)の場合は、属性を追加して差別化を行うことで、(2)の場合は、解答されるべき答えの概念関数を一般化することで、修正を行う。本システムでは、表形式の知識表現を用いているために、これらの手続きは、キューブの加減や、一部を修正したりすることにより容易に実現できる。

例えば、例題の工場において、Grinder-Lを改良して、軟らかい材料に加えて硬い材料も研磨できるようにしたとしよう。つまり、材料が硬く、要求品質が低いときにはGrinder-IIの代わりにGrinder-Lを解答するように修正がしたいとする。この時、利用者はまずシステムを通常と同様に使用し、解答がGrinder-Hと出たところで、Grinder-IIに対して(1a)を、Grinder-Lに対して(2)を選択する。確認質問の後、システムは下記に示すように知識ベースを修正する。

概念の一般化 概念の一般化は、肯定キューブを加えかつ、同じキューブを否定キューブから引くことにより行なう。この例では、Grinder-Lの概念関数を、材料が硬く要求品質が低い時を含むように一般化したい。この場合、システムは表3に示したようなキューブを作成し、知識ベースに加える。

	材料		要求品質	
	硬い	軟らかい	高い	低い
Grinder-L	○	×	×	○

表 3: 概念の一般化

概念の特殊化 概念の特殊化は、単に否定キューブを加えることによって行なう。つまり、誤った答が出力されたことを利用者が指摘した時には、システムは知識ベースにその属性の組合せからなる否定キューブを加える。この例では、材料が硬く、要求品質が低い場合はGrinder-IIは解答からはずしたいので、システムは表4に示された否定キューブを加える。

	材料		要求品質	
	硬い	軟らかい	高い	低い
#Grinder-II	○	×	×	○

表 4: 概念の特殊化

対話型知識獲得の結果、システムは、表1に表3と表4を合成した表5を作成する。この表は、論理的には正しくこのまま

でも使用できるが、冗長である。そこで、知識ベースエディタの表圧縮機能を使うことにより、これと等価な表6に変換できる。

	材料		要求品質	
	硬い	軟らかい	高い	低い
Grinder-II	○	×	○	○
Grinder-H	×	○	○	×
#Grinder-H	○	×	×	○
Grinder-L	×	○	×	○
Grinder-L	○	×	×	○

表 5: 合成された冗長な表

	材料		要求品質	
	硬い	軟らかい	高い	低い
Grinder-H	○	○	○	×
Grinder-L	○	○	×	○

表 6: 縮小された最終的な表

概念の差別化 属性を付加することにより、複数の概念を区別することができる場合には、概念の差別化を行う。この詳細については本報告では省略するが、材料の大きさという新しい属性によって、Grinder-Lを新しいグラインダであるGrinder-LLと差別化した例を表7に示す。

	材料		要求品質		材料	
	硬い	軟らかい	高い	低い	大	小
Grinder-H	○	×	○	○	○	○
Grinder-H	×	○	○	×	○	○
Grinder-L	×	○	×	○	○	×
Grinder-LL	×	○	×	○	×	○

表 7: 差別化された表

6 おわりに

本システムは、知識ベースを直接作成修正するための会話型の表エディタと、メニューを使った利用者インタフェースを持つ推論実行プログラムとからなる。知識ベースエディタは、そのエディタ機能に加えて、表圧縮機能とシミュレーション機能を持っている。システムはすべてC言語で記述されており、EWS4800およびPC9800で動作する。また、このシステムは、答えが正しかった場合にはその回数を数えて頻度を学習することにより、より少ない質問回数で答えに到達する機能も持っている。

これまでに、故障診断、ツール選択などの例題を通して評価を行なってきた。その中で、最大の知識ベースは、あるユーザにおける製品選択の知識ベースであり、全部で100以上の選択項目を持ち、キューブの数は800程度である。この評価の結果、業務専門家自身が本システムを使用することにより、知識エンジニアの手を借りずに知識ベースを容易に開発出来ることが確認された。

最後に、本研究の機会を与えて下さったC&Cシステム研究所 山本所長、同システム基礎研究部 吉村部長に深謝します。

参考文献

[Quinlan86] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning* 1:1, 1986, pp. 81-106.
 [Koseki91] Y. Koseki, et al. "DT: A Classification Problem Solver with Tabular-Knowledge Acquisition" *Proc. Third Int. Conf. on Tools for AI*, 1991, pp. 156-163.