

帰納型プログラムの機械学習システム『Bくん』

3E-8

赤間 浩樹, 峯崎 俊哉, 成嶋 弘

東海大学

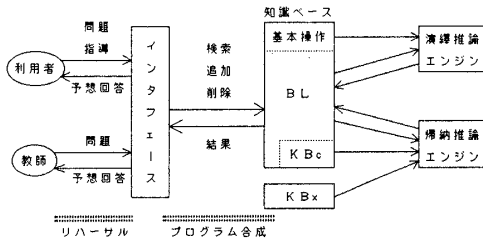
1. はじめに

帰納推論は, 古くから人間の基本的な問題解決能力として研究されてきたが, 演繹推論が Prolog やエキスパートシステムなどで広く利用されているのに比べ, 帰納推論の研究は実用化という面で非常に立ち後れている。[1]

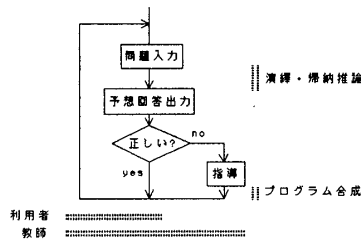
本稿では, この帰納推論を応用した, 手続きの機械学習システム『Bくん』について述べる。『Bくん』は, (1) 帰納型プログラミング言語の採用, (2) 指導の例からのプログラム合成, (3) リハーサルによるプログラミング環境, という3つの特徴を持っている。

2. システムの概要

本システムの全体構成, および流れは, おおよそ 図1, 図2 のようになっている。



【図1 システムの構成図】



【図2 システムの流れ】

3. 帰納型プログラミング言語とBL

従来のプログラミング言語が解釈系に演繹推論のみを使用したのに対し, 帰納型プログラミング言語では解釈系に帰納推論の能力も仮定する。たとえば, LISP に帰納推論エンジン ind を組み込んだ場合, 図3 のようなプログラミングが可能となる。

```
(setq E '((1 3)(2 5)(3 7)(4 9)))
(dotimes (v (ind E 10))
  (print v))
```

実行結果
0
1
2
:
20

【図3 LISP を帰納型に拡張した例】

本システムの知識表現言語BLは, この帰納型プログラミング言語の一種である。ただし, 現在のBLでは, 帰納推論エンジンの帰結範囲を正整数上に限定し, 以下のような帰納推論アルゴリズムを用いて実現している。

【帰納推論の基本アルゴリズム】

帰納推論を下式の v_m を求める過程とする。

$$\frac{e_1 \rightarrow v_1}{e_2 \rightarrow v_2} \dots \frac{e_n \rightarrow v_n}{e_m}$$

$$v_m$$

$$E \hat{=} \{(e_1, v_1), (e_2, v_2), \dots, (e_n, v_n)\}$$

KBc $\hat{=}$ 特徴抽出(数値化)関数の集合

KBx $\hat{=}$ 外挿関数生成関数の集合

このとき E, e_m, KBc, KBx より v_m を求めるアルゴリズムは,

```

for  $\forall Fc \in KBc$ 
  for  $\forall Fx \in KBx$ 
    if  $\forall (e, v) \in E : Fx(E, Fc(e)) = v$ 
      return( $Fx(E, Fc(e_m))$ )
return(FAIL)

```

となる。

4. 指導の例からのプログラム合成

一般に、例からのプログラム合成システムにおいては、操作の合成（計画）の過程において組合せ爆発を起こしてしまう。そこで、この問題を回避するため、本システムでは始めから操作の系列（以後、これを指導と呼ぶ）を例として与えている。このことで、帰納推論の適用範囲は狭まるものの、実用性と実現容易性は高まり、さらに、必要となる例の個数も少なく済むようになる。

ところで、本システムは帰納型プログラミング言語を採用しているので、本質的には単なる事例の追加がプログラム合成となる。しかし、実際には帰納推論エンジンの帰結範囲が正整数上に限定されているため、それに対応した変換が必要になる。具体的には、繰返し構造の抽出と、分岐構造の生成が主な働きとなっている。

5. リハーサルによるプログラミング

通常プログラミングという行為は、まず頭の中にモデルを創り、そのモデルを操作するコードをプログラムとして記述していくスタイルである。

これに対し、リハーサルによるプログラミング^[2]では、画面上のオブジェクトに対し操作を直接適用、またはUNDOしながら試演（Rehearsal）していく過程そのものがプログラミングになる。

本システムでは、教師が指導の例を与えていく過程に、このリハーサルによるプログラミングを採用している。

6. おわりに

本システムは、正の例のみからの、完全教師を仮定した手続きの機械学習システムであり、帰納推論の実行過程をプログラム合成時からプログラム実行時へ移したことが大きな特徴である。

このように、学習時には単に事例を内部に蓄えることだけを行い、実行時に一般化して解釈するという手法^[3]は、学習の初期段階や一般化の結果に自信がない時に有効である。逆に、自信がある時には、高速化という面からも事例の集合を一般化した規則に書き換えるという手法が有効である。

そこで、今後は2つの手法をバランスよく組合せ、より効率的な学習システムを構築していきたいと考えている。

また、プログラミング・システムとしての実用性の向上と、理論的な面からの考察も今後の課題である。

なお、本システムは著者の修士論文^[4]としてパソコン上にICLISP(ICL)で実現されている。

7. 謝辞

本システムの開発にあたって、貴重なご意見を頂いた有川節夫先生（九大）に感謝します。

8. 参考文献

- [1] Feigenbaum, E.A. et al. : The Handbook of Artificial Intelligence, Vol. 2 and 3, 1982.
- [2] Finzer, W., Gould, L. : Programming by Rehearsal, Byte, Vol. 9, No. 6, 1984.
- [3] 佐藤 理史 : Memory-Basedアプローチと規則学習, 学習のパラダイムとその応用シンポジウム論文集, 情報処理学会, 1989.
- [4] 赤間浩樹 : 東海大学大学院 理学研究科 数学専攻 1989年度修士論文