1C－3

# 再帰論理プログラムのProgolによる学習のための例の順序処理

齋田 豊　五味 エジソン 智志　石塚 満
東京大学 工学部 電気工学科
e-mail: saida , gomi, ishizuka @miv.t.u-tokyo.ac.jp

## 1 はじめに

帰納論理プログラミングの目的は、与えられた正例を満たし、負例を含まない論理プログラムを求めることである。本研究では、論理プログラムの中でも多くの有用なプログラムを含む二節からなる再帰論理プログラムを対象とし、これをProgolを用いて効率的に学習することを目的とする。

Progolは優れた学習システムであるが、再帰論理プログラムの学習に対して特別な配慮を行なっていないため、正常な学習が行なわれない場合がある。ここでは、制約付2-mmgを利用してProgolを応用できる範囲を広げるための例の前処理について述べる。

## 2 Progolの問題点

Progolは優れた学習システムであるが、再帰論理プログラムの学習においては、その動作原理上、学習が正常に行なえない場合がある。

1. 例を与える順序に問題がある場合

   ベースケース(深さ0)や、深さの浅い例の順序が後ろだと計算に時間がかかったり、学習が正常に行なわれなかったりする場合がある。

   例）再帰の深さの大きな例が初めの方で与えられる場合

   $append([1,2,3],[4,5,6],[1,2,3,4,5,6]).$

   $append([a,b,c],[d,e,f],[a,b,c,d,e,f]).$

   $append([a],[b],[a,b]).$

   ...

   Progolの出力は以下の通りとなる。

   $append([A],B,[A \mid B]).$

   $append([A \mid B],C,[A \mid D]) : -append(B,C,D).$

   $append(A,B,C) : -append(C,A,B).$

2. 例の再帰の深さに不連続がある場合

   Progolは$\theta$-包摂を用いているため、例の深さに不連続があると学習が行なえない場合がある。

   例）member(a,[b,a]).のような深さ1の例を含まず、深さ0(ベースケース)、
   および、深さ 2(ex.member(a,[b,c,a]).), 深さ3(ex.member(d,[a,b,c,d]).) の例のみを含んでいる場合、ベースケースは一般化されるが、再帰節は一般化されずに出力される。

## 3 制約付2-mmgによる例の分割

極小多重汎化(mmg)という手法が再帰論理プログラムの節の頭部を帰納するのに用いられる。極小多重汎化はPlotkinの最小汎化(lgg)のの拡張と考えることが出来る。例えば、与えられた正例の集合に対して、最小汎化のアルゴリズムは例の集合をカバーするただ一つの節を見つける。一方、mmgのアルゴリズムは例をカバーする「木パターン言語の極小結合」と呼ばれるリテラルの集合を探し出す。リテラルは木パターンによって表現することが可能である。木パターンの結合の多項式時間推論は、有村、篠原、大月によって研究されている。ここでは、例をカバーする二つの節を探す2-mmgを利用する。

2-mmgを用いることで、再帰の深さごとに正例を分割することが可能である。ただし、2-mmgの結果が必ず例を再帰の深さごとに分割するわけではないため、得られる結果から再帰的に2-mmgを利用することで探索を行なう必要がある。例としてmemberを分割した結果を以下に示す。

$$\begin{cases} 1. \\ member(A,[B,A]). \\ member(A,[B,C,D \mid E]). \rightarrow 2. \end{cases}$$

$$\begin{cases} 2. \\ member(A,[B,C,D,A]). \\ member(A,[B,C,D]). \rightarrow 3. \end{cases}$$

$$\begin{cases} 3. \\ member(A,[B,C,A]). \\ member(4,[3,4,5]). \end{cases}$$

---

A Processing of Example Order for Inducing Recursive Programs using Progol
Dept. of Information and Communication Engineering, Faculty of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113, JAPAN

## 4 例の前処理とProgolの利用

再帰論理プログラムの学習に関するProgolの不具合に対処するために、2-mmgによる例の分割を利用することが有効である。

例の数が多い場合には、2-mmgの計算に長時間を要する。そのため例の部分集合を取り、例の入れ換えを行なうことが必要となることが考えられる。

具体的にどの程度の時間がかかるかについて、appendを用いて実験した結果を以下に示す。

2-mmgはPrologで書かれたプログラムである。

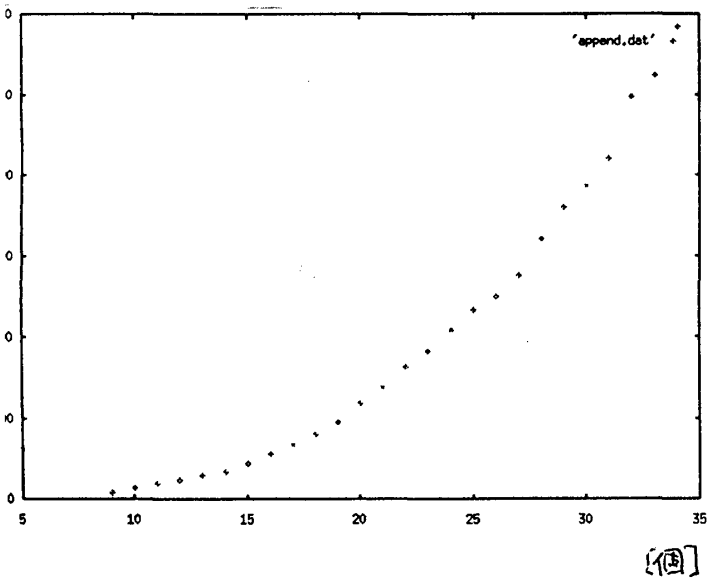


図 1: 例の数と2-mmgの計算時間

### 4.1 例の順序の変更

例の深さに不連続がなく、順序を変更することによってProgolでの学習が可能となる場合、2-mmgによって例を分割し、順序を前後してProgolに与えることで、望ましい結果を得ることができる。

例) 上記のappendの例で、正例を2-mmgによって分割すると

$$\begin{cases} 1. \\ append([A \mid B],[C],[A,D \mid E]). \\ append(A,[B,C \mid D],[E,F \mid G]). \end{cases}$$

$$\begin{cases} 2. \\ append([A,[B \mid C]],[D,E]). \\ append(A,[B \mid C],[D,E,F \mid G]). \end{cases}$$

$$\begin{cases} 3. \\ append([],[A,B \mid C],[A,B \mid C]). \\ append([A \mid B],[C \mid D],[A,E \mid F]). \end{cases}$$

上にあげる3通りの解が得られる。2,3の結果をこの順序通りにProgolに与えることで正しい結果である

$append([],A,A).$

$append([A \mid B],C,[A \mid D]) :- append(B,C,D).$

が得られる。

### 4.2 例の深さに不連続がある場合

与える例の深さに不連続がある場合、2-mmgによって連続な深さの例を切り出してProgolに与えることで正しい再帰節を得ることができる。

例)

$member(3,[4,2,3]).$

$member(3,[5,4,2,3]).$

$member(4,[3,2,4]).$

$member(4,[1,3,2,4]).$

$member(1,[3,2,1]).$

$member(2,[1,3,2]).$

$member(2,[4,1,3,2]).$

からProgolによって

$member(A,[B,C,A \mid D]).$

$member(A,[B \mid C]) :- member(A,C).$

なる結果が得られ、求められた再帰節を初期の例の集合に加えて再びProgolに与えることで、ベース節も求めることができる。

## 5 おわりに

本研究は、論理プログラムの中でも多くの有用なプログラムを含む二節からなる再帰論理プログラムのProgolによる学習を目的とし、Progolにある問題点と、2-mmgを応用してそれに対処する方法について述べた。この手法によってProgolの持つ不完全さを補間することが可能である。

### 参 考 文 献

1) Stephen Muggleton. Inverse entailment and Progol ,1995

2) H. Arimura, T. Shinohara, and S. Otsuki. Polynomial time inference of unions of tree pattern languages. In *Proceedings of the Workshop on Algorithmic Learning Theory*, p. 105–114, 1991.

3) E.S. Gomi, M. Ishizuka. Examples' Depth and Induction of Recursive Logic Programs. In 情処第*51*回全大,4J-1, 1995.

4) Stephen Muggleton. *Inductive Logic Programming course. Lecture notes, University of Tokyo.* ,1993.