


新概念を発見する帰納論理プログラミング手法に関する一提案

2P-1

山本 公洋 鈴木 英明 伊藤 正樹

 NTT ソフトウェア研究所

1 はじめに

本稿では、ソフトウェア開発上流工程における要求獲得に対する計算機支援の一手法を提案する。

逆導出原理 [1] を用いた事例からの概念生成法は、ユーザの持つ断片的なイメージから体系化されたシステム概念を組み立てる有効なツールである。逆導出原理を用いたシステムに CIGOL があるが、その探索戦略のためユーザが必要とする概念が学習できない場合や、生成途中の概念についてユーザが熟知しなければ体系化された概念を生成できないなどの問題がある。

これに対し本稿では、探索戦略として確率的探索を用い、生成途中の概念が求めたい概念に徐々に近づく遺伝的アルゴリズムに逆導出原理を組み込んだ概念獲得手法を提案する。また、本手法により上記の問題点が解決されることを適用実験で示す。

2 逆導出原理を用いた概念獲得

概念生成のためにアブダクティブな推論手法のひとつである逆導出原理を用いた場合、以下の様な問題が生じる。

- ★ 途中で生成する概念の数が組合せ的爆発を起こす
- ★ 途中で生成される概念の多くが人間にとって意味がない

このため逆導出原理を用いた概念獲得では、膨大な探索空間に対する効率良い探索戦略と、人間にとって意味ある概念を途中で得る機構を実現する必要がある。

このうち概念の有効性判別機構では、有効と思われる概念の性質をユーザが自由に設定・変更できる必要がある。何故なら、有効な概念の性質とは、例えば汎用性（モジュール化による再利用性）や運用効率（実行速度やメモリ量）などが考えられるが、一意に特定できず、ユーザの意図や状況により異なるからである。

逆導出原理を用いた CIGOL では、組合せ的爆発を防ぐため、逆導出原理を用いて生成される複数の概念から、ヒューリスティック関数を用いて後で必要となる仮説を選択し、選択された候補のみを残してその他

の概念は削除する。また、できるだけ意味のある概念を残すため、ユーザが直接、生成途中の概念候補の有効性の判別を行なう。

しかし、CIGOL を概念獲得のために用いる場合、次の様な問題点が生じる。

1. 求めたい概念を形成する重要な仮説を探索範囲から除外してしまう
2. 求めたい概念に関する具体的かつ詳細なイメージをユーザが事前に把握する必要がある
3. 逆導出原理を用いた概念形成に関する高度な判断を、多数回にわたりユーザに強いる

3 提案手法の概要

逆導出原理を用いた概念獲得に必要な概念の有効性を判別する機構や探索機構を、遺伝的アルゴリズム [2] の機構により実現する。

有効な概念の性質は評価関数形式でユーザに指定してもらい、その評価関数を用いて概念の有効性判別を試みる。確率的探索手法の一種である遺伝的アルゴリズムでは、学習アルゴリズムと評価関数が独立しており、評価関数のパラメータ化が可能で、ユーザが自由に設定できる。評価関数を用いることで、概念に関する具体的かつ詳細なイメージを事前に把握する必要なく、ユーザは有効な概念の性質を指定できる。

また、学習アルゴリズムと評価関数が独立しているため、ユーザの探索に関する負担を軽減できる。

探索機構は、以下の様な形式で実現する。逆導出原理を用いた概念獲得では、有効な概念が prolog のプログラム（規則集合）の一部として得られる。そのプログラムは事例に逆導出原理を繰り返し適用することで生成される。有効な概念を得るためには、有効な規則の組合せを求める必要があるが、事例に対する逆導出原理の試行錯誤的な適用と生成されたプログラムに対する評価を繰り返す中で重み付けを行なうことにより、経験的に有効な規則の組合せを求める。これにより、効率の良い探索が期待できる。また、確率的に仮説を保持するので、重要な仮説を探索範囲から除外する危険性を回避できる。

4 コーディング手法

以下に、逆導出原理と遺伝的アルゴリズムを組み合わせた具体的な概念獲得手法を示す。

遺伝的アルゴリズムでは、染色体を持った生物を学習の解に見立て、個体と呼ぶ。そして、個体の集団（個体集団）に対し、以下の様な世代交代を繰り返させることで学習（進化）を促す。

- 教師信号（評価関数）に比例する様に、個体の生存確率を決定する（選択淘汰）。
- 生き残った個体を元に、遺伝操作（交叉・突然変異）を施し、新しい個体（解）を生成する。

その個体の染色体に、事例から逆導出原理を用いてプログラムを形成する手順、具体的には、どの規則に対し逆導出オペレータ（3種類）を適用するか、その順序等をコーディングする。

染色体は固定長で、特徴を司る最小構成単位である遺伝子型（genotype）の部分と情報を持たないスペースの部分より構成される。長さが一定になる様に、遺伝子型を適当な長さのスペースで区切るものとする。各遺伝子型は、逆導出オペレータの一回の適用（オペレータの種別と適用対象となる規則）を表す。

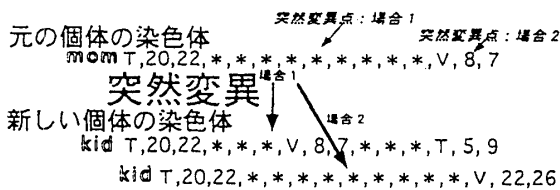
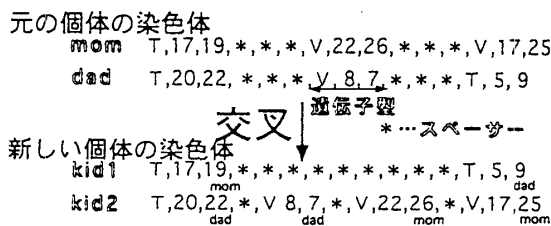


図 1: 交叉と突然変異

交叉と突然変異の方法を図 1 に示す。交叉と突然変異では、致死遺伝子（交配されえない遺伝子）が生じない様チェック機構を設け、致死遺伝子が生じる場合は、元となる個体の染色体をそのまま継承させる。

学習アルゴリズムを図 2 に示す。

5 適用実験

実験結果を図 3 に示す。

実験には文献 [1] のサンプルプログラムを用いた。また、評価関数は 2 種類用意した。評価関数 1 は、公理の規則数の逆数である。評価関数 2 は、文献 [1] で heuristic として用いられたものであり、変数、項、リテラル及び節の総合数である。

step1 個体集団を初期化する。具体的には、遺伝子型情報を全く持たない染色体を有する個体を用意する。

step2 各個体を評価関数に従って評価し、生存確率を求める。この際、個体が形成するプログラムの冗長性削除を行なう。

step3 生存確率に基づき、ランダムに選択淘汰を行なう。

step4 各個体に遺伝子操作を施し、新しい公理を生成する。

step5 2へ戻る。

図 2: 学習アルゴリズム

member 述語は評価関数 1 と評価関数 2 で同じプログラムを学習した。arch 述語は、評価関数 1 と評価関数 2 で学習結果が異なり、評価関数 2 で文献 [1] と同じ結果を得ることができた。reverse 述語は、member 述語や arch に比べ、学習にかなりの時間、step 数を要した。

Max generation = 1000 Population = 100
 Low pressure = 0.1 High pressure = 1.0
 Pressure period = 25
 Cross-over rate = 0.3 Mutation rate = 0.2

プログラム	正事例数	負事例数	世代交代数	
			評価関数 1	評価関数 2
member	13	10	57	12
arch	11	13	15	96
reverse	18	16	×	536

図 3: 実験結果

6 おわりに

本稿では、逆導出原理と遺伝的アルゴリズムを組み合わせた概念獲得手法を提案し、体系化行為として、ある種の評価関数に基づく体系の最適化行為を試みた。実際にシステムを作成、能力を検証した結果、簡単なプログラム学習に対しては、適用可能性を認識できた。

今後は、プログラム事例と評価関数のサンプルを増やし、提案した手法が有効な範囲を明確にすると共に、体系化行為について、さらに検討を加えていく予定である。

参考文献

- [1] Muggleton, S. and Buntine, W.: Machine Invention of First-order Predicates by inverting Resolution, *Proc of the 5th Int. Workshop on Machine Learning*, pp.339-352(1988).
- [2] 北野宏明: 遺伝的アルゴリズム, 人工知能学会誌, Vol. 7, No. 1, pp.26-37(1992).