Genetic Programing,Decision Treeによる学習について

新美 礼彦

1998 年 8 月 7 日

1 キーワード

Evolutionary Algorithm, Genetic Programming, Selection, Mutation, Crossover, Fitness Function, Decision Tree, Consept Learning System, Information Gain, Pruning, Estimated Error

2 はじめに

今まで研究してきた内容について、おもに遺伝的プログラミングと決定木構築法について、まとめてみた。遺伝的プログラミングに関しては、遺伝的プログラミングの基本的なアルゴリズムである遺伝的操作についてまとめた。また、進化アルゴリズムとしての理論的研究についても触れた。決定木構築法に関しては、C4.5 を取り上げ、その基本的なアルゴリズムについてまとめた。最後に、現在研究している内容についての構想メモ的なものをのせた。

3 進化論的アプローチ

進化アルゴリズム (EA:Evolutionary Algorithm) とは、生物の遺伝と進化を模倣し、個体群を用いて解を探索する計算手法であり、幅広い問題に適用可能なロバスト性の高い最適化手法として注目されている。EA とはいくつかの独立して展開されてきた類似の手法の総称であり、以下の手法が含まれている。

ES:Evolutionary Strategy:進化戦略 オペレータとして突然変異を主に用いて、n 個体の親から m 個体の子を選択するという世代交代を行う。定量的な研究が困難ではなく、突然変異の効果などが数学的に解析されている。

EP:Evolutionary Programming:進化プログラミング オートマトンの適応的学習が中心。

GA:Genetic Algorithm:遺伝的アルゴリズム 文字列の染色体を交叉、突然変異などの遺伝的操作を用いて操作する。GA を応用したクラシファイアシステム (Classifier System) という研究分野も成立。

GP:Genetic Programming:遺伝的プログラミング GA の染色体表現をグラフ構造やツリー構造を扱えるように拡張したもの。

特に情報への進化論的アプローチでは、環境との相互作用や問題解決の過程を通して自らの内部に持つ情報を変革し、適応的な学習を行うことにより、従来の人工知能 (AI:Artificial Intelligence) と異なり、「知能とは何か?」といった疑問は問わずに、自然がやっているような賢く見えるやり方での問題解決や学習の実現を試みている。

4 遺伝的プログラミング (GP)

生物進化論の特に自然淘汰（選択 (selection)）と遺伝子変異（突然変異 (mutation)）の考えに注目している。自然淘汰により、優れた能力を持った個体の遺伝子を後の世代に残し、遺伝子操作によって遺伝子に変異を加えることにより新しい可能性を

探る。遺伝子操作には、交叉、突然変異などが使われる。

染色体 (chromosome) 表現：GA が文字列（一般的には１次元文字列で 0,1）を使うのに対し、GP では GA の染色体表現をグラフ構造やツリー構造を表現できるようにしたものである。一般的にはツリー構造を用い、LISP の S 式表現で表記されている。

4.1 GP のアルゴリズム

1. 問題ごとの関数ノードと終端ノードのランダム文法から初期集団を発生させる。
2. 集団内のそれぞれの個体を計算し、問題の解決にどのくらい関係しているかという適応度を求める。
3. 遺伝的操作により、次の世代を発生させる。
   (a) 最良個体をコピーする
   (b) 突然変異 (mutation) により新しい個体を発生させる
   (c) 交叉 (crossover) により新しい個体を発生させる
4. 終了条件が満たされたかどうかを調べ、満たされていたら終了する。満たされていなかった時は、2. へ戻る。
5. 各世代の最良個体が、GP により求められた近似解になる。

4.2 GP の設計要素

GP では次の５つの基本要素を設計することで、さまざまな問題への適用が可能となる。

- 非終端記号：非終端ノードで使う記号。LISP の S 式での関数。
- 終端記号：終端ノード（葉）で使う記号。LISP の S 式でのアトム。
- 適応度
- パラメータ：交叉、突然変異の起こる確率、集団のサイズなど。
- 終了条件

また、問題によって使用する遺伝的操作を変更する場合がある。

4.3 適応度関数 (Fitness Function)

適応関数により、各世代である個体がどの程度適応したかを評価する。一般的にこの適応度は選択や遺伝的操作の確率やプログラムの終了条件に深く関係している。

適応度には以下の４つがある。[6]

生適応度 (Raw Fitness) 各個体から適応関数により計算される。(f_r)

標準化適応度 (Standerd Fitness) 生適応度を 0 が最も良く、適応度が正数になるようにしたもの。(f_s)

修正適応度 (Adjusted Fitness) 標準化適応度から計算され、0 から 1 の間の値を取り、1 が最も良い。($f_a = 1/(1 + f_s)$)

正規化適応度 (Normalized Fitness) 修正適応度から計算される。ある世代における個体に貢献度の指標となる。($f_n(i) = f_a(i)/\sum_j f_a(j)$)

4.4 選択 (Selection)

選択とは、集団の中から、ほかの個体よりも優れた性質を持つ個体の遺伝子をより高い確率で生存させる機構である。適応関数による評価によって求められた適応度により、遺伝的操作の割合が異なる。

以下、代表的な選択方法を紹介する。

ルーレット選択方式 適応度に比例した面積を有するルーレットを作り、そのルーレットを回して当たった場所の個体を選択する。ルーレットを個体数が選られるまで繰り返し回す方式。

トーナメント選択方式 集団の中からある個体数（トーナメントサイズ）をランダムに選び出し、

その中で最も良いものを選択する。この過程を集団数が得られるまで繰り返す方式。

ランク選択方式 各個体を適応度の大きいものから順に並べ、この順位に応じた関数により子供の数を決める方式。

エリート戦略 成績の良い親のいくつかを常にコピーして、次の世代に残す方式。

4.5 遺伝的操作

GP では解空間の探索に遺伝的操作を用いる。各個体の遺伝子に変更を加えることにより、新しい遺伝子構成を持った染色体を作り、解空間での新しい領域の探索を試みる。この変更を加える操作を遺伝子操作と呼び、生物の遺伝子に起こる変化に基づいた操作を行う。GP で一般的に用いられる操作を以下に示す。

- 複製 (reproduction)
- 交叉 (crossover)
- 転位 (inversion)
- 突然変異 (mutation)

4.5.1 複製 (Reproduction)

集団中のある個体をそのまま次の世代の集団にコピーする操作である。選択によりどの個体を複製するかが決定され、より適応度の高い個体が次の世代に受け継がれることになる。

4.5.2 交叉 (Crossover)

交叉は、2個の親となる個体の染色体を組み換えることによって親の形質を受け継ぎつつも、親とは異なる形質を持ちあわせた子供の個体を生み出す操作である。

GA では、文字列を交叉点によりつなぎかえるので、交叉点の取り方により、以下のような種類がある。

- 一点交叉 (one-point crossover)
- 複数点交叉 (n-point crossover)
- 一様交叉 (uniform crossover)

これに対し、GP ではそれぞれの親の部分構造を交換することにより、新しく子供の2個体を生成する。GP の染色体はその部分木を任意に入れ換えても、生成される染色体が有効であり、2個体間での部分木の入れ換えも可能である。

ex.1

```
(* (- X Z) (+ X Y))
(- (* X 1.2) (+ Y (* Z X)))
        ↓
(* (+ Y (* Z X)) (+ X Y))
(- (* Z 1.2) (-X Z))
```

ex.2

```
(- (* X 1.2) (+ Y (* Z X)))
(- (* X 1.2) (+ Y (* Z X)))
        ↓
(- (+ Y (* Z X)) (+ Y (* Z X)))
(- (* X 1.2) (* X 1.2))
```

ex.3

```
(- (* X 1.2) (+ Y (* Z X)))
        ↓
(- (* Z X) (+ Y (* X 1.2)))
```

なお1個体の親の中での部分木の入れ換えを転位という。

進化の初期の段階では、交叉のために選ばれた2個体の染色体は大きく異なっていると思われる。そのため、交叉により大きな変化を加えることができ、生成された染色体は、それまでに存在していない遺伝子構成を持つことが期待される。進化が進ん

だ段階では、集団内に似通った遺伝子構成が広まっていくために、染色体間の違いが小さくなり、交叉の結果として新しい遺伝子構成を持った個体が生成されることが期待できなくなる。

4.5.3 突然変異 (Mutation)

突然変異はあるノードを他のノードに置き換える操作のことである。作用するノードによって、以下のように分類することができる。

- 終端ノードから非終端ノードへの突然変異：新しい部分木の生成
- 終端ノードから終端ノードへの突然変異：ノードラベルの付け替え
- 非終端ノードから終端ノードへの突然変異：部分木の削除
- 非終端ノードから非終端ノードへの突然変異：新しい非終端ノードと古い非終端ノードの子の数が等しい場合にはノードラベルの付け替え、異なる場合には部分木の生成・削除

ex.

```
(* (- X Z) (+ X Y))
        ↓
(* (* X Z) (+ Z Y))
```

突然変異には、集団の多様性維持と遺伝子型での局所探索の役割がある。しかし、GP では、木構造の任意の場所に任意の部分木が形成されるために、集団内での遺伝子の多様性は十分確保されると考えられる。また、終端ノード間の突然変異は交叉で実現できる。以上のような理由から、GP ではあまり突然変異率を高くしなくても良い。

4.6 GP の適用例

GP はさまざまな分野に適用されている。GP の適応範囲は AI の問題解決からロボット、分子生物学など実際的な問題まで多岐にわたっている。GP では木構造に交叉と突然変異を用いることにより木を変形し、LSIP(S 式) プログラムや概念木などを探索する。

以下にその一部を示す。

- 概念学習システム (CLS:Consept Learning System)
- ニューラルネットワークの学習
- 人工生命のプログラム

ex.XOR の学習

- GP による XOR

```
(P (* 1.841 (P (* 1.66 D0)
               (* -1.387 D1) ) )
   (* 1.901 (P (* 1.191 D1)
               (* -0.989 D0) ) ) )
```

- ニューラルネットによる XOR

```
y = (P (* 0.5 (P (* 1 D1)
                 (* -2 D0) ) )
       (* 0.5 (P (* 1 D0)
                 (* -2 D1) ) ) )
```

p:引数の和を求め、閾値 (1.0) 以上なら 1 を、そうでないければ 0 を出力する。*,+:乗算、和算

4.7 GP の問題点

GP は現在、以下のような問題を抱えている。

- 数学的な背景が証明されていない
- 並列化と個体数の増加による計算量の増大
- 交叉によるスキーマの破壊
- ノードの設計時の表現問題
- 木構造の評価方法

4.8 拡張 GP

GP の問題点を解決するため以下のような手法が提案されている。

- 自動関数定義 (ADF:Automatic Defined Function)
- その他のモジュール構造獲得アルゴリズム [10]
- MDL 基準による適応度計算
- 型理論に基づく GP
- 並列 GP(Parallel GP)

5 進化アルゴリズムの理論的研究

EA では、適用されている問題が一般に厳密解を効率よく求めることが困難な問題が多く存在するので、近時解法として位置づけられてしまい、強力な理論展開が難しい。交叉は、生成される子が複数の親個体に依存することやこの遺伝子のとりうる範囲が両親の遺伝子型に依存して変化することなど、突然変異に比べて動作が複雑であるため、解析が難しい。

- 環境により適応した個体がより多くの子孫を残すというダイナミクスの解析
- 突然変異や交叉によって探索される適応度関数の形状に関する解析

6 進化のダイナミクスに関する研究

- 集団遺伝学の基本定理とスキーマ理論
- マルコフ連鎖による解析
- 遺伝的浮動：個体数の有限性と子孫を生成する不確定性を考慮した場合、たとえ個体間に適応度の差がなくても、個体群が次第に多様性を失う現象
- ビルディングブロック仮説

7 コード化 (Coding)

GA/GP のコード化については以下のような基準が提案されている。[7] この場合、いずれかの犠牲のもとに他が実現することもありえるので、すべてを完全に満たす必要はない。形質遺伝性は領域依存性が大きく問題領域ごとに調べなくてはならないが、GA/GP において収束速度に決定的に影響する。

完備性 (completeness) 解候補はすべて染色体として表現できること

健全性 (soundness) 染色体はすべて解候補と対応すること

非冗長性 (non redundancy) 染色体と解候補は１体１対応となること

形質遺伝性 (character preservingness) 親の形質を適切に子に継承すること

8 設計レベル

GA/GP は無作為な標本抽出に基づく確率的探索法の１種である。他の確率的探索と比較すると、集団からの選択や染色体の交叉など複数の解候補を操る点に特徴がある。ここで、GA/GP の性能を３つに分類してみる。[7]

ランダムレベル ランダムサーチによっても実現できてしまう性能レベルであり、精度を上げるには非常に多くの試行が必要とされる、最も低い性能レベル。

局所探索レベル 局所探索（突然変異）の繰り返しによって実現できる性能レベル

大局探査レベル 大域探索（選択と交叉）によらなければ実現されないような性能レベル

GA/GP を応用するからには大域探索レベルの性能が発揮されなければ意味がない。大域探索レベルを実現する形質遺伝は、コード化・交叉設計によって生み出され、多様性維持によって支持されなければ、性能レベルが落ちてしまう。

9 決定木 (Decision Tree)

一般に分類モデルは、クラスと属性間の関係はフローチャートのような簡単なものか、あるいは手順を書いたマニュアルのように複雑で構造化されていないもので定義される。モデルを作るには、関係のある熟練者にインタビューすることによってモデルを作る方法や、記録された膨大な分類データを調べ、特定の例を一般化することによりモデルを帰納的に作る方法などがある。

決定木では、木の根から葉に初めて出会うまでテストを繰り返しながら事例を分離するものである。中間ノードでは、事例のテストが行われる。この結果にしたがって、処理プロセスは対応する部分木の枝を進む。このプロセスが最終的に葉まで達した時、その事例のクラスは葉に記録されているクラスであると判断される。

決定木の例：

天候 = 晴れ：
　　湿度 ≤ 75：開催
　　湿度 > 75：中止
天候 = 曇り：開催
天候 = 雨：
　　強風 = 真：中止
　　強風 = 偽：開催

9.1 他の分類モデル

- 実例に基づいた分類器
- ニューラルネットワークによる分類
- 統計手法を用いた分類手法

9.2 決定木の構成要素

- 葉：クラスを表わす。
- 判別ノード：1つの属性値を調べるテストを指定する。テストのそれぞれの結果に1つの分岐と部分木が対応している。

9.3 決定木構築のアルゴリズム

C4.5 では以下の2つの過程を経て決定木を構築する。

1. 初期決定木の生成
2. 枝刈り

9.3.1 初期決定木の生成

CLS 法は決定木による分類学集の代表的な方法である。CLS 法は期待獲得情報量最大化原理に基づく分類を行う。これは、事例集合をそれぞれの属性によって分類した時に得られる情報量の期待値を求め、これから入力項目の期待値を引くことにより、分類における重要さを計算し、これを最大にするような属性による分割をするアルゴリズムである。これにより、決定木の根に重要な属性を集めることができる。通常は過剰に分類され決定木が複雑になるのを防ぐためにそれぞれの事例集合における最小限の事例数が決められている。

9.3.2 決定木構築アルゴリズムの条件

アルゴリズム使用のために、以下の条件を満たしている必要がある。

- 解析されるデータでは、1つの事例に関するすべての情報はあらかじめ決められた性質ある

いは属性によって表現できなくてはならない。それぞれの属性は離散値か連続値を取るが、ある事例を表現するために使われた属性のタイプは、別の事例で別のタイプとして扱われてはならない。

- 事例が割り当てられるべきカテゴリは前もって準備されていなくてはならない。(教師付き学習 (supervised learning))
- クラスは事例がそのクラスに属するか属さないかを決めることができるようにはっきりと定義できる必要がある。また、事例の数はクラスの数よりも十分に多い必要がある。
- データの中から同じようなパターンを見つけることによって行われるため、十分な事例が必要である。必要となる事例の数は、属性やクラスの数と分類しようとするモデルの複雑さなどの要因から決まる。

9.4 連続値属性に関する取り扱い

連続値属性を扱う場合、統計的手法により層別分類することがある。一般にこの方法は計算量がかかるので、C4.5 では閾値との比較により、大小判断によってテストしている。

閾値を求めるアルゴリズムには以下のようなものがある。

1. 連続属性値により訓練事例集合をソートする。
2. 隣り合う連続属性の区間の中点を閾値として選び、すべての場合について評価する。
3. 最も評価の高かった閾値を採用する。

このアルゴリズムを使用すると、訓練事例数が増加するにつれ、選んだ決定木を実際に構成するために必要な処理の計算量は線形に増加する。しかし、連続値のソートに必要な計算量はログオーダーで増加する。このため、連続値属性のソートの方が実際に決定木を構成するよりも処理時間を必要とする可能性がある。

9.5 枝刈り (Pruning)

再帰的な分割法は、そのままでは複雑な木となることがある。木が複雑になってしまう 1 つの原因は、初期決定木が入ってくる情報を不用意に取り込んでしまうことである。これにより、木が複雑になるだけでなく、単純な木よりも精度が落ちることがあるということである。

決定木は、1 つ以上の部分木の代わりに葉で置き換えることによって、単純化される。葉はその部分木に含まれるテスト事例のうち、最も多いクラスを選ぶことによって定義される。枝刈りを行うかどうかは、誤り率を予測することによって判断する。枝刈り後に予測された誤り率が枝刈り前よりも低いのであれば、その部分木を置き換え、枝刈りする。誤り率を予測する方法は 2 種類の手法が考えられる。

- 訓練事例とは異なる新しい事例を使って、木とその部分木の誤り率を予測する方法
- 訓練事例から誤り率を求め、訓練事例の含まれる母集団の誤り確率を考え、ある信頼度から求まる上限を用いる方法

10 決定木からのルールの抽出

作成された決定木からルールを抽出することができる。これにより、より一般的なルールを得ることが可能となる。決定木では、ある事例があるクラスに分類されるために満たさなければならない条件は、根から葉までの経路に沿ったテストの結果を追うことによって得られる。このようにして求められたものをプロダクションルールとみなすと、元の木とまったく同じように事例を分類するはずである。しかし、ここで得られるルールは、元の木からそれほど単純化されていない。得られたルールから無意味な条件を取り除くことによって、より単純化されたルールを得ることができる。

単純化の例：

F=0:

```
            J=1:no
            J=1:
                  K=0:no
                  K=1:yes
      F=1:
            G=1:yes
            G=0:
                  J=0:no
                  J=1:
                        K=0:no
                        K=1:yes
```

rule.1 if (F=0, J=1, K=1) then yes

rule.2 if (F=1, G=1) then yes

rule.3 if (F=1, G=0, J=1, K=1) then yes

rule if (J=1, K=1) then yes

11 現在やっている研究

C4.5 と GP の組み合わせ（ハイブリッドシステム）

C4.5 で連続値属性の取り扱いと初期決定木の構築を行い、GP で探索と収束を行い、C4.5 で探索により構築された決定木の枝刈りを行う。

- 連続値属性の意味のある分割
- 初期決定木のある程度理論的な構築
- GP での冗長性を枝刈りによって取り除く

C4.5 のアルゴリズムをファジイ決定木に応用し、GP とのハイブリッドシステムからファジイ決定木生成アルゴリズムを構築する。

12 参考文献

[1] 遺伝的アルゴリズムの基礎, 伊庭斉志,1994

[2] 遺伝的プログラミング, 伊庭斉志,19946

[3]AI によるデータ解析,J.Ross Quinlan, 古川康一,1995

[4] 数学で見た生命と進化, カール・シグムンド, 富田勝,1996

[5] 人工生命と進化システム,ATR 進化システム研究室,1998

[6]lil-gp 1.01 User's Manual,Douglas Zongker, Bill Punch, Bill Rand,1996

[7] 遺伝的アルゴリズムの工学的応用, 山村雅幸, 小林重信,1994

[8]The Genetic Programming Tutorial Notebook,Jaime Fernandez,1997

[9] 進化アルゴリズムの理論について, 喜多一,1997

[10] 遺伝的プログラミングによるモジュール構造獲得, 橋山智訓, 苗村高義, 大熊繁,1997

[11] 遺伝的アルゴリズムを用いたファジイ決定木の生成, 馬野元秀, 吉村正義, 鳩野逸生, 田村坦之,1996

[12] 遺伝的アルゴリズムの現状と課題, 小林重信,1993

[13] 遺伝的アルゴリズムの基礎, 伊庭斉志,1994

[14] 決定木を用いた遺伝的プログラミングによるニューラルネットワークの創発的学習, 松本昇,1997