2C3-07

or演算子を含んだ関数ノード群を持つGPによる拡張決定木の生成

Extended Decision Tree with GP's Function Nodes Including "OR Operator"

新美 礼彦　　田崎 栄一郎
Ayahiko Niimi　　Eiichiro Tazaki

桐蔭横浜大学 工学部 制御システム工学科
Department of Control and Systems Engineering, Toin University of Yokohama

It is easy for an unexpected decision tree to be generated in the decision tree generation with a genetic programming because the probability operation is contained. In the description of the decision tree by normal genetic programming, the division conditions by the attribute are connected with AND operator, and the tree evaluates effectiveness as a rule. However, if the description of the function node is changed, a more flexible rule is sure to be able to be generated. In this paper, we show that the description of a more flexible decision tree is possible by the addition of the OR function to the function node group.

1. はじめに

遺伝的プログラミングをデータマイニングに用いると、進化計算による確率的な操作により意外な知識を発見することが期待できる。遺伝的プログラミングでは、染色体表現に構造表現を用いることにより、使用できる知識表現が決定木からルールまで幅広く適用可能である。しかし、適応度関数により個体を評価する都合上、決定木のように知識全体をカバーできるような形式が主に利用されてきた。一般的な遺伝的プログラミングによる決定木の記述では、属性による分割条件をANDで接続して、ルールとして評価していく。

しかし、遺伝的プログラミングでは、遺伝子表現に置き換えられ適応度関数が定義できれば実装可能である。これは、他の知識表現も遺伝的プログラミングに実装可能なことを示している。そこで本論文では、今までわれわれが行ってきたif-elseによる決定木・ルール表現を検討することにした。まず相関ルールなどを参考に、AND結合によるルール表現を作成する。そこにOR結合を組み込むことにより、より柔軟な表現による決定木・ルールの表現を検討した。これらはすべて遺伝的プログラミングの関数ノードの定義を置き換えることにより実装している。そのため遺伝的プログラミングによる学習の枠組みの変更は最小限になっている。

検討した決定木、ルール表現による学習の違いを検討するために、これらの関数ノードと自動関数定義を組み込んだ遺伝的プログラミングによる学習の統合を行った。これをUCIのMachine Learning Repositoryの評価データからの決定木生成問題に適用し、従来の関数ノード定義による学習法による結果と比較・検討した。

2. 遺伝的プログラミング

遺伝的プログラミング(Genetic Programming:GP)は、生物進化論の考えに基づいた学習法であり、そのアルゴリズムの流れは遺伝的アルゴリズム(Genetic Algorithm:GA)と同様である。[伊庭 96] その特徴は染色体表現がGAと異なり、関数ノードと終端ノードを用い構造表現ができるように拡張してあることである。GPでは、関数ノードと終端ノードを用いてLISPのS式形式で個体を表現する。今回は、決定木を表現するためにツリー構造を用いた。このため、関数ノードに条件文、終端ノードをそれぞれの属性値とクラス名を用いて決定木を表現した。また、本論文では、生成される決定木をコンパクトにするため、自動関数定義(Automatically Defined Function:ADF)を用いた。[Koza 94]

連絡先: 〒225–8502 神奈川県横浜市青葉区鉄町1614
桐蔭横浜大学 工学部 制御システム工学科
田崎　栄一郎
TEL:045–974–5070 FAX:045–978–1311
E-mail:tazaki@intlab.toin.ac.jp

3. GPによる決定木・ルール表現

決定木からルールを抽出する手法からの考え方を用いて、GPで決定木を表現するときに関数ノードとしてif-elseを使うことが可能である。[Quinlan 93]

(if A C D) if (A) then C else D.

これを拡張して、データベースからの属性と比較できるように以下の定義を用いることもできる。[新美 99, 新美 00]

(ifeq A B C D) if (A == B) then C else D.

その他にも以下のような演算子を用いたルール表現が考えられる。

- if-else
- AND
- OR
- NOT

一般的な相関ルールでは、条件部分がANDで結合した形で表されている。[喜連川 97, 寺邊 00] これをif-else形式で表現すると、相関ルールで定義されていない部分の扱いが困難になる。そのため、相関ルールをGPで学習するのは、難しいと考えられる。また、if-elseにより決定木でORを表現するには、同じ部分構造を何度も持たなければならない。それに対して、ANDは単純に決定木の経路を伸ばしていくだけでよい。このことから、if-elseを用いた決定木表現では、ORを表現した部分による決定木のサイズの増加のほうが、ANDを表現した部分による決定木のサイズの増加よりも起こりやすいことが考えられる。

4. AND と OR を用いた GP

ここでは、OR を含んだルールを GP によって表現しやすくするため、AND と OR によるルール表現を以下のような関数ノードとして定義する。

(AND A B C D) if (A and B) then C else D.

(OR A B C D) if (A or B) then C else D.

GP において、多様なルールの表現法を実装するのは比較的容易である。if-else や AND、OR などの実装は、関数ノードの定義を変更するだけで行うことが可能である。関数ノードの定義を変えるだけなので、GP による学習の枠組みを変える必要がない。したがって、場合によっては適応度関数やその他のパラメータに関しても、そのままのものが使える可能性がある。

今回の変更でも、関数ノードの定義のみ変更でよく、適応度関数やパラメータを変更する必要がない。この定義では if-else の時に比べて AND や OR を含んだルールを表現しやすくなっているので、生成される決定木のサイズの縮小が期待される。しかし、定義する関数ノードが増えることにより組み合わせの増加が起こるため、学習速度に関しては、あまり改善を期待できない。

5. データベースからの決定木生成問題への適用

ルール表現の違いによる GP の学習の違いを検討するために、評価用データを用いた実験を行った。評価用データには、UCI の Machine Learning Repository から house-votes を使用した。 [Blake 98] これにより、他の手法と比較して提案した手法がどの程度有効かを検証した。評価データは “y”,“n”,“?” の 3 つの属性値を持つ “handicapped-infants”,“water-project-cost-sharing” などの 16 の属性と “democrat”, “republican” の 2 つのクラスからなるデータである。house-votes の全データ 435 件のうち 50 件を学習用に使用した。

学習データから GP により決定木を生成した。GP のパラメータは、事前に行った if-else を用いた実験の時と同じものを用いた。（結果は表 1）なお表では、個体のサイズ、木の深さに関しては、未使用の ADF 定義部分を除いてある。

AND 単体より AND と OR でルールを学習した方が、精度の高いルールを生成することができた。AND と OR の両方を使用する場合、定義する関数ノードが増えるので、組み合せの増加が起きる。このため、学習が遅くなり、最良個体獲得までの世代数が長くなってしまったものと思われる。決定木のサイズ、深さについては if-else を用いたものから改善することができた。

表 1: 各手法による生成決定木の比較

	訓練 (%)	全体 (%)	サイズ	深さ	獲得世代数
AND のみ	96.0	65.5	10	1	29
AND+OR	96.0	92.0	5	1	1730
if-else (参考)	100.0	86.0	11	3	586

6. おわりに

本論文では、遺伝的プログラミングによる決定木ルール表現を検討し、if-else 形式のルール表現のほかに AND と OR を用いた表現を遺伝的プログラミングに実装した。また、実装したルール表現の有効性を検証するために、UCI の Machine Learning Repository から house-votes データを用いて、決定木を構築し、その評価を行った。

その結果、決定木のサイズの改善を行うことができた。また、AND と OR を用いたものでは、精度の改善も認められた。拡張したルール表現は、遺伝的プログラミングの関数ノードの定義を置き換えることにより実装している。そのため遺伝的プログラミングによる学習の枠組みの変更は最小限になっている。このことより、AND や OR を用いたルール表現も遺伝的プログラミングでは有効であるといえる。

今後は、他の検証用データを用いた評価を行うとともに、NOT、NAND、NOR や XOR などによるルール表現についても利用できるか検討を行い、どのルール表現を使用するかに関する指針を検討していく予定である。

参考文献

[Blake 98] Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science. (1998).

[伊庭 96] 伊庭斉志: 遺伝的プログラミング, 東京電機大学出版局 (1996).

[喜連川 97] 喜連川 優: データマイニングにおける相関ルール抽出技法, 人工知能学会誌 Vol.12 No.4,pp.513–520 (1997).

[Koza 94] Koza, J.R., Kinner, K.E.(ed.), et.al: Scalable Learning in Genetic Programming Using Automatic Function Definition, Advances in Genetic Programming, pp.99–117 (1994).

[新美 99] 新美 礼彦, 田崎 栄一郎: 無効ノード削除と連続値属性の適応操作を加えた遺伝的プログラミング, 第 13 回 人工知能学会全国大会論文集, pp.257–258 (1999).

[新美 00] 新美 礼彦, 田崎 栄一郎: 相関ルールアルゴリズムと組み合わせた遺伝的プログラミングによる学習, 第 14 回人工知能学会全国大会論文集, pp.270–271 (2000).

[Quinlan 93] Quinlan, J.R.: C4.5: Programs for Machine Learning, Morgan Kaufman Publishers (1993).

[寺邊 00] 寺邊 正大, 片井 修, 椹木 哲夫, 鷲尾 隆, 元田 浩: 相関ルールにもとづく属性生成手法, 人工知能学会誌 Vol.15 No.1,pp.187–197 (2000).