2K－4

遺伝的プログラミングを利用した二進判別木の設計*

谷川 徹 (s1041123@u-aizu.ac.jp), 趙 強福†

会津大学コンピュータ理工学部‡

1 はじめに

現在ニューラルネットワークの学習にはバックプロパゲーションといわれる逐次修正方が主に用いられているが、この方法では局所解に陥る可能性が高いという問題点があり、しかもネットワークの構造、サイズ等事前に与えておく必要がある。判別木を設計しニューラルネットワークに変換することによってこの問題を解決できるが [1]、ＮＰ完全問題を含んでいるため [2]、本研究では遺伝的プログラミング（以下ＧＰ）を利用することにしている [3]。しかしＧＰにも最終的に求められる判別木のサイズが大きくなることが多いという問題点がある [4]。そこで本論文ではこのサイズを小さくする方法を提案する。

2 判別木について

私たちの研究では判別木は七つの項目によって定義し、これらは全て節点に該当する。節点には終端点と非終端点の二種類あり、特に以下の様に表すことができる。

節点 $= \{t, label, P, L, R, C, size\}$

- t は節点の番号を表しており、$t = 0$ のとき、この節点のことを根という。
- $label$ は終端点のクラスラベルである。
- P は親へのポインタで根のときは $P = NULL$ となる。
- L と R は左右それぞれの子へのポインタで終端点に関してはそれぞれ $NULL$ が入る。
- C にはレジスタのセットであり、非終端点に関して $n = C[0]$, $a = C[1]$ となり、以下の比較によって一つの判断を行う。

  $$feature_n < a?$$

  もしそうであれば左の子へ、そうでなければ右の子へと進む。
  終端点に関して $C[i]$ は、この節点に分類された i 番目のトレーニングサンプルの数である。
  そして終端点の $label$ は次のように決定される。

  $$C[K] = \max_{\forall i} C[i]$$

- $size$ は部分木として考えたときの節点のサイズであり、このパラメータは木の適応度を調べるときに役に立つ。根のサイズは木全体のサイズであり、終端点のサイズは1とする。

3 ＧＰについて

ＧＰとは生物の進化の様子をもとに考えられた木構造で表される最適化アルゴリズムである。その方法として、まず乱数を用いて木の初期集団を発生させる。ここで、この初期集団を第0世代とする。その後、木の交叉、突然変異、自然淘汰を繰り返し最適化していく。木の交叉とは木の集団の中から適応度の高い木をランダムで二つ取出し、それぞれの木の一部分を入れ替えることである。突然変異とは、木の節点の情報（条件）をある確率で変化させることで木の集団が局所解に陥っている場合にそこから脱出する可能性を高める働きをしている。自然淘汰とは交叉により発生した木の数だけ適応度の低い木を順に消去していくことである。この時の木の集団を第1世代として、目標とする適応度に到達するまで世代を繰り返す。以上が一般に行われているＧＰの方法である。

*Binary decision tree design based on the genetic programming

†Toru Tanigawa, Qiangfu Zhao

‡University of Aizu, Aizuwakamatsu 965-8580,Japan

4　ＧＰのサイズ縮小方法

判別木をＧＰに基づいて自動的に発生するには、さきほどの議題において"木"を"判別木"に置き換えれば良い。しかし、このままでは、通常非常に大きい判別木を発生してしまう。現在までにもサイズを縮小させる方法は色々と考えられているが[5]、ここでは０～９までの数字の認識を例にして新しい縮小方法を説明していく。

1. まず０か１を判別できる判別木をＧＰによって求め、これを部分木として保存しておく。同様に２か３、４か５、６か７、８か９を判別できる判別木をそれぞれＧＰによって求め、部分木として保存しておく。

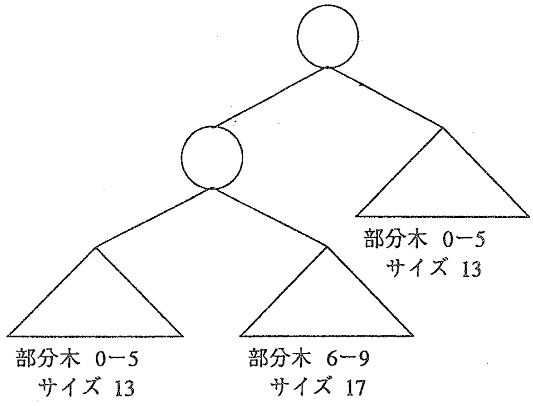
2. 次に（０，１）か（２，３）を判別できる判別木をＧＰによって求め、先ほどの作業と同様保存しておく。ここで、1. で求めた木をサブプログラムとして使用できる。（４，５），（６，７）も同じように求めて次の段階に入る。

3. さらに（４，５，６，７）か（８，９）を判別する判別木を求め、最後に（０，１，２，３）か（４，５，６，７，８，９）を判別する判別木を求めれば後は事前に求めてきた部分木を再利用すれば０～９までの数字を識別できることになる。

ここで使用したＧＰは全てＡかＢかの二択であり、求められた判別木は検索時間も短く、サイズも小さくなっていることが期待できる。また部分木を作り、あらゆる所に再利用できることから、実際のサイズは見掛けのサイズよりもさらに小さくなっていると考えることができる。
例えば、右上のような判別木（丸は節点、三角は部分木を表す）が求められたとすると、見た目にはサイズが $2+13+17+13=42$ あるように思えるが、実際には終端点にきてから部分木を呼び出すという作業を行うので、サイズは $2+13+17=32$ ということになる。

5　実験結果

発表時に追加したい。



6　現在考えられる問題点

この例の場合で考えるといくら二択とはいえ（０，１，２，３）か（４，５，６，７，８，９）を判別する判別木がどのくらい小さいサイズで求められるのか、ということがあげられる。単数と単数ならその特徴をつかみやすくすぐに判別できるが、複数と複数になると少ない特徴だけでは判別しきれない可能性があり、最終段階で細かい判別木を作ってしまうと、事前に作った部分木の効果が薄れてしまう。解決策としては、解く問題に対してそれぞれ適切な所で部分木を作ることをストップさせることであると現在考えている。

参考文献

[1]I.K.Sethi, "Entropy nets: from decision trees to neural networks," 18-7(1990)1605-1613

[2]L.Hyafil, "Construction optimal binary decision trees is NP-complete," Information Processing Letters, 5-1(1976)15-17

[3]J.R.Koza, Genetic Programming, Fourth Printing, The MIT Press,(1994)

[4]M.Shirasaka, Q.F.Zhao, et al, "Automatic design of binary decision trees based on genetic programming," Proc. The Second Asia-Pacific Comference on Simulated Evolution and Learning(SEAL'98), Canberra,(1998)

[5]Q.F.Zhao, "A Study on Evolutionary Design of Binary Decision Trees," Proc. CEC'99, PP.1998-1993, Washington,DC, July,1999