データベースシステム 117－4
（１９９９．１．２２）

超高速分類木作成方法

吉井　裕人

概要

本論文では、新しい分類木作成アルゴリズム“Generalized Pyramid Architecture Classification Tree”(G-PACT)を開示する。このアルゴリズムは、特徴空間を分別する超平面の傾きに関する拘束条件と、位置に関する拘束条件の２つの拘束条件を有することを特徴とする。本方法の有効性を評価するため、Proben1ベンチマークデータセットを用いた認識実験を行った。結果によれば、本アルゴリズムは十分な認識率を示す分類木を極めて短時間で作成できる能力があることがわかった。また、経験的に過学習の回避できる能力が示唆された。

キーワード：分類木、高速、超平面

A High Speed Algorithm for Induction of Classification Tree

Hiroto Yoshii

Abstract

This paper discloses a novel algorithm for induction of classification trees. The algorithm, Generalized Pyramid Architecture Classification Tree (G-PACT), imposes two restrictions on coefficients of linear combinations and splitting points of each value, which generates classification trees with high accuracy in a remarkably high speed. We carried out nine experiments using benchmark datasets from Proben1, for analysis of Generalized PACT's capability. The results support that the algorithm has a great ability to construct adequate performance classification trees in quite a short period of time. The empirical study indicates that the algorithm has an ability to avoid overfitting problems.

Key words： classification tree, high speed, hyperplane

1　はじめに

分類木は“Data Mining”を含むパターン認識問題全般に適用可能なアルゴリズムである[1, 2]。一般にパターン認識問題とは、「特徴空間の中の点集合として学習パターンが与えられた時、特徴空間内のある点として表現されるテストパターンがどのカテゴリーに属するかを判別する問題」という定式化ができる。長年にわたって、このパターン認識問題に対する数々のアルゴリズムが提案されてきた。中でも分類木とニューラルネットは非常にポピュラーなアルゴリズムであり、それらは特徴空間上でカテゴリー領域を分別する境界を決定するアルゴリズムということがいえる。

分類木は、上記のカテゴリー判別境界として特徴空間の次元軸に平行な超平面を採用するアルゴリズムから、一般的な傾きを持った超平面を採用するものに拡張されてきた。カテゴリー判別境界としては、超平面の代わりに超曲面を採用する方がより一般的なのであるが、たとえ超平面を採用するアルゴリズムであっても、非常に多くの計算時間が必要となるという問題が存在した。一方、最近になって大規模データベースの利用が可能になってきたこともあり、非常に高速でデータベースから知識を抽出する需要は、ますます増しているのが現状である。

そこで、本論文では非常にシンプルであるが、超高速な辞書作成時間を実現する、新しい分類木作成方法“Generalized Pyramid Architecture Classification Tree”[3, 4, 5]を提案する。このアルゴリズムの能力を検証するため、ベンチマークとして利用されている Proben1[6]データセットを用いた実験を行った。実験結果によると、本アルゴリズムが、非常に短い時間で十分な認識率を持つ分類木が作成できる能力を有することが示された。以下、アルゴリズムを具体的に説明し、実験結果を報告する。

2　Generalized PACT

アルゴリズムは分類木を作成する際に、2つの拘束条件をかしている。一つは超平面の係数に関する拘束条件で、もう一つは超平面で空間を分割する場所に関する拘束条件である。以下、理解し易いように、まず軸に平行な平面を採用したバージョンを説明した後で、それを一般的な傾きを持った超平面を採用したバージョンへ拡張する。

2.1 軸平行バージョン G-PACT を 1 次元問題に適用した場合

図1は、軸に平行な超平面を用いる G-PACT を1次元の認識問題に適用した場合の模式図である。図が示す通り、分類木を作成する前に、あらかじめ特徴空間を再帰的に分断しておく。図下側の数直線上の点が個々の学習パターンであり、１０個の白丸がカテゴリーAのサンプルを表し、１０個の黒丸がカテゴリーBのサンプルを表す。全てのサンプルは 0.0 から 1.0 の範囲に分布する。まず初めに、中間地点 0.5 の場所で特徴空間を分断する。これにより2つの区間[0.0,0.5], [0.5,1.0]が得られる。次に、この2つの区間をそれぞれ中間地点で分断する。そして、4つの区間[0.0,0.25], [0.25,0.5], [0.5,0.75], [0.75,1.0]が作成される。このようにして、特徴空間の分断は再帰的に行われる。

分類木を作成する際、上記の最初の切断点 0.5 をまずチェックし、次に第2グループの切断点 0.25 と 0.75、次に第3グループの切断点、0.125, 0.375, 0.615, 0.875 をチェックするようにする。こうして得られた分類木を図上に示す。図において、四角はインターナルノードを示し、その中に書かれた数字はノードの番号を表す。白い丸と黒い丸は、それぞれカテゴリーA、カテゴリーBのリーフノードを示す。図が示すとおり、ルートノードにおいて全ての学習パターンは 0.5 未満のサンプルと 0.5 以上のサンプルに分割される。そして、2番のインターナルノードは 11 サンプルを含み、3番のインターナルノードは 9 サンプルを含むことになる。もし、これらのノードが複数のカテゴリーに属するサンプルを含む場合、インターナルノードとなり更に中間地点での分別を続けていくことになる。そして、最終的に全てのノードが単一のカテゴリーに属するサンプルしか含まない（＝リーフノード）状態になった時、

分類木作成を終了する。結果として、図1上に示すように、5つのインターナルノードと6つのリーフノードが作成される。

G-PACTアルゴリズムのキーポイントは、特徴空間の分割を、最初、大局的な観点より行い、必要があれば、どんどん特徴空間の分割を細密化していくことにある。そして、学習パターンを徹底的に分別する分類木が作成され、理論的には学習パターンに対する認識率は 100%になるのである。

2.2軸平行バージョンG-PACTを2次元問題に適用した場合

図2に軸に平行な超平面を採用した G-PACTを2次元認識問題に適用した例を示す。1次元認識問題と同様に、事前にx軸y軸それぞれの軸を再帰的に分割する。図2下に示した通り、32個のカテゴリーAのサンプルと32個のカテゴリーBのサンプルが学習パターンとして与えられている。全ての学習パターンはx軸y軸の0.0から1.0の範囲に分布する。前サブセクションと同様、まず初めにx軸y軸それぞれの分布範囲の中間地点（x = 0.5 and y = 0.5）で特徴空間を分断する。次に、それぞれ断片化された区間の中間地点（x = 0.25, x = 0.75, y = 0.25, and y = 0.75）で分断する。そして更にそれぞれの区間の中間地点(x = 0.125, x = 0.375, x = 0.615, x = 0.875, y = 0.125, y = 0.375, y = 0.615, and y = 0.875）で分断する。こうして、最終的には8*8 = 64の区間が得られることになる。

特徴空間の次元が1つしかない場合は、G-PACT アルゴリズムを用いて分類木を作成する際に何の不確定要素も入ってこない。ところが、2次元認識問題の場合は、それぞれのインターナルノードでx軸とy軸のどちらの次元を選ぶかということを決定しなければならなくなる。例えば、ルートノードにおいては、2つの中間地点（x = 0.5 と y = 0.5）のどちらで学習パターンを分ければよいかを決定しなければいけない。これを決定する指標として、G-PACTにおいては“相互情報量”を採用する。これは、エントロピー−Σp*log(p)の減少量の期待値である。（詳細は文献]1]の32ページ参照）例えば、ルートノードでのカテゴリーのバランスは(A: 32 B: 32 entropy: 0.69)となっている。そこで、直線x = 0.5で学習パターンを分けた場合、それぞれの子ノードのカテゴリーのバランスは(A: 5 B: 25 entropy: 0.45)と(A: 27 B: 7 entropy: 0.51)になる。そして、直線y = 0.5で学習パターンを分けた場合、バランスは(A: 20 B: 7 entropy: 0.57)と(A:12 B:25 entropy: 0.63)となる。ルートノードにおいて、上記の2つの選択肢があるわけであるが、エントロピーを分類効率の指数として、効率がよい方を選ぶわけである。前者の場合、エントロピー減少の期待値は(30/64*0.45 + 34/64*0.51) − 0.69 = 0.21 となり、後者の場合、(27/64*0.57 + 37/64*0.63) − 0.69 = 0.09 となる。よって、前者の直線x = 0.5で学習パターンを分けることを選択するわけである。そして全てのインターナルノードにおいて、水平な直線で分けるのが良いか、垂直な直線で分けるのか良いかを評価していく。図2のインターナルノードの右側に表示されている式は、そのノードで使用した判別直線を示す。これは、図2下の太い線で書かれた特徴空間上の直線に相当する。一般的にいって、全てのインターナルノードにおいて“相互情報量”を計算しなければいけないのであるが、判別点を固定してあるので、本アルゴリズムは極めて少ない計算量しか必要としないわけである。

2.2斜平面バージョンG-PACTを2次元問題に適用した場合

図3が一般斜平面を利用したG-PACTを図2と同じ認識問題に適用した例である。軸平行の超平面は一般斜超平面の特別な場合と考えることができる。つまり、ある一つの係数を除いて全ての係数を 0 に固定すると軸平行の超平面が得られる。同様に、特徴空間の次元の一次結合における係数を、ある特定の集合から選ぶという拘束条件をかけることによって、完全に自由な超平面の選び方には及ばないものの、軸平行な超平面に比べると、よりフレキシブルな超平面が分別境界として利用できるわけである。図においては、{-1, 0, 1}という3つの値をこの係数集合として選んだ。よって、全ての一次結合の組は、x + y, x − y, x, and y の4つとなる。（全ての係数の組み合わせ

は、これの2倍になるが、対称性に基づいて半減できる）一般的にいうと、係数集合として上記の3つの値を選ぶと d 次元の認識問題では$(3^d-1)/2$個の一次結合の組が得られる。そして、一旦、一次結合の組が固定されると、それぞれの一次結合によってできる変数を新しい変数として、それら全てを再帰的に分割していく。そして分類木構築時に、どの変数（一次結合）を使って分別するかを相互情報量を使用して決める。ルートノードでは $x = 0.5$, $y = 0.5$, $x + y = 1.0$, and $x - y = 0.0$ の4つの選択肢があり、第2ノードでは $x = 0.75$, $y = 0.5$, $x + y = 1.0$, and $x - y = 0.0$ の4つ、第3ノードでは $x = 0.25$, $y = 0.5$, $x + y = 1.0$, and $x - y = 0.0$ の4つ、そして第4ノードでは $x = 0.875$, $y = 0.5$, $x + y = 1.0$, and $x - y = 0.0$ の4つの選択肢がある。このそれぞれのノードで4つの相互情報量を計算し、もっとも量の大きい分別面を選ぶ。最終的に、図3上にあるように、4つのインターナルノードと5つのリーフノードを含む分類木が作成される。

3 実験結果

実験で使用したデータは Proben1[6]を用いた。このデータセットは認識アルゴリズムのベンチマークを行うため用意されてデータセットで、全てのデータは前処理が施されている状態でインターネット上で取得できる。データセットは training set, validation set, test set の3種類で構成されており、それぞれ、50%, 25%, 25%の割合となっている。この validation set というのは、ニューラルネットワークの過学習を防ぐために用いられる学習パターンの一部で、training set と validation set をあわせて、一般的な意味での学習パターンとなる。まず、最初に学習パターンとして training set のみを用いた結果を示し、その後で training set と validation set を学習パターンとして用いた結果を示す。なお、以下全ての実験において計算機の環境は CPU: PentiumII 300MHz, Memory: 384Mbyte, OS: Linux, Compiler: gcc, and Optimize option is On となっている。

3.1 training set のみを用いた実験結果

表1の4番目の列にそれぞれのデータセットにおける test set に対する認識率を示す。注意すべきことは horse1 の問題を除いて全ての問題の学習パターンに対する認識率は 100%であるということである。このデータセットで認識が完全でない理由は、「分類木の深さが 10 以上で相互情報量が 0.01 以下なら分類木作成を止める」という条件を設定したので、horse1 の問題の場合、2つのカテゴリーが混在するリーフノードが残ったためである。文献[6]の認識率の結果と比べてみると、G-PACT がニューラルネットと同等の認識能力を保持していることがわかる[7]。以下、詳細な実験条件を説明していく。

cancer1, diabetes1, glass1 の3つの問題では、{-1, 0, 1}を一次結合の係数集合として採用した。このことによって、例えば glass1 問題では、一次結合の総数は $(3^9-1)/2 = 9841$ となる。card1, heart1, heartc1, horse1, soybean1 の5つの問題では、2種類の一次結合の組を採用した。第1の種類は唯一の係数が1で、残りは0である一次結合で、第2の種類は一次結合中の2つの係数が{(1, 1), (1, -1)}のどちらかであり、残りは0である一次結合の組である。例えば card1 問題ではオリジナルな特徴量として 51 個の変数があり、一次結合の総数は $51 + ({}_{51}C_2)*2 = 2601$ となる。thyroid1 問題に対しては、3種類の一次結合の組を採用した。第1の種類と第2の種類は前記5つの問題と同じで、第3の種類は、一次結合の中の3つの係数が{(1, 1, 1), (1, 1, -1), (1, -1, 1), (-1, 1, 1)}のとれかであり、残りは0である一次結合の組である。それぞれの問題で準備した一次結合の総数は表1の3番目の列に示してある。

分類木の作成時間を表1の5番目の列に示す。G-PACT アルゴリズムは、最高で1秒、最悪でも2分以内に分類木を作成していることがわかる。

3.1 training set と validation set の両方を用いた実験結果

G-PACT アルゴリズムは、理論的には学習パターンに対して 100%の認識率を実現する。この結果は本アルゴリズムが過学習の問題を回避できる可能性を示唆する。この仮説を確かめるため、経験的な実験を行った。表1の右端の列は training set に加えて validation set を学習パタ

ーンとして用いた認識結果である。heartc1 問題を除いて全ての問題に対して 3.1 で示した認識率以上の認識率が実現できた。

4 結論と考察

本論文において、新しい分類木作成アルゴリズムである G-PACT を開示した。ベンチマークデータセットを用いた実験結果によれば、G-PACT アルゴリズムは十分な高認識率を有する分類木を極めて高速に作成できることがわかった。また、過学習に対する耐性があることが経験的に示された。G-PACT アルゴリズムの計算時間の大きな部分として、区間を再帰的に分割する処理がある。この処理は容易に並列処理できることから、並列計算のハードウェアを利用することにより更に高速な分類木作成能力が追及できる。

歴史的に、パターン認識研究者はコンパクトな分類木を作成することを追求してきた。例えば図1の問題は、3つの分別面さえ用意すれば全ての学習パターンが認識できる。よって、ルートノードを含めインターナルノードが3つの分類木がもっともコンパクトな分類木ということになる。そして、この分類木が最良解だと考えられてきた。G-PACT アルゴリズムはこの方向に逆行した方法をとることにより、高速な学習時間と過学習に対する耐性を手に入れたわけである。もちろん、今後、過学習に対する耐性に関する研究は、理論的、確率論的に議論されていくことが必要になってくる。

参考文献

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman & Hall Inc., New York, NY, 1993

[2] S. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology", *IEEE Trans. SMC*, Vol. 21, No. 3, pp. 660—674, 1991

[3] Hiroto Yoshii, "Pyramid Architecture Classification Tree", *Proc. of ICPR'96*, pp. 310—314 Vol. 2 , 1996

[4] Hiroto Yoshii, "Binary PACT", *Proc. of ICPR'96*, pp. 606—610 Vol. 4 , 1996

[5] Hiroto Yoshii, "Distributed PACT", *Proc. of CESA'96 IMACS Multiconference*, pp. 548—553, (Symposium of Robotics and Cybernetics), 1996

[6] L. Prechelt, "PROBEN1—A Set of Neural Network Benchmark Problems and Benchmarking Rules", *Technical Report 21/94*, ftp://ftp.ira.uka.de/pub/papers/techreports/1994/1994-21.ps.Z ftp://ftp.ira.uka.de/pub/neuron/proben1.tar.gz, 1994

[7] S. Salzberg, "On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach", *Data Mining and Knowledge Discovery*, Vol. 1, pp. 317—327, 1997

Problem	Number of Attributes	number of linear combination	accuracy for test set	learning time (sec)	accuracy for test set (using all training set)
cancer1	9	9,841	95.4%	10.2	95.4%
card1	51	2,601	75.0%	7.1	75.0%
diabetes1	8	3,280	71.9%	4.2	74.0%
glass1	9	9,841	60.4%	4.7	64.2%
heart1	35	1,225	74.8%	3.3	79.1%
heartc1	35	1,225	78.7%	1.0	77.3%
horse1	58	3,364	65.9%	5.5	71.4%
soybean1	82	6,724	91.8%	28.4	92.4%
thyroid1	21	5,761	98.6%	101.7	98.6%

表1： Proben1 を用いた認識実験結果

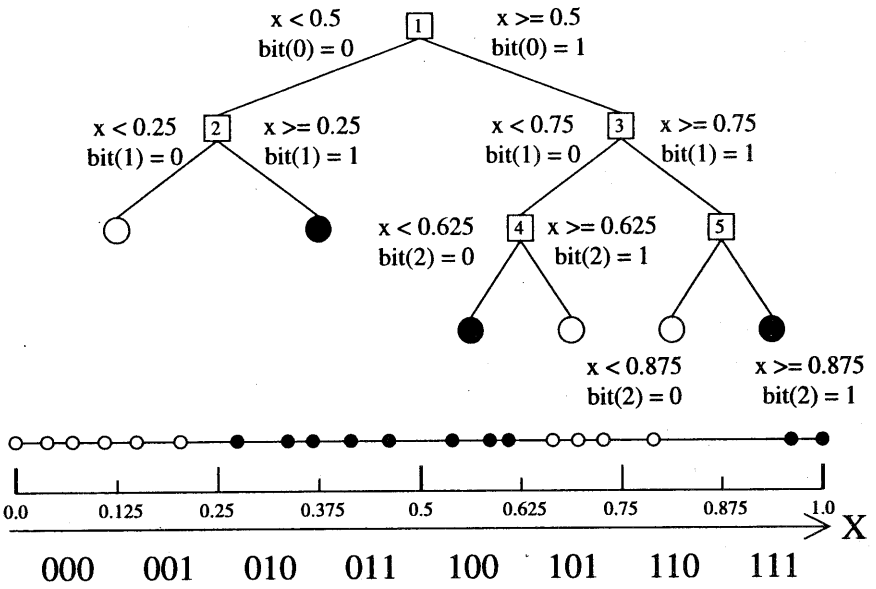
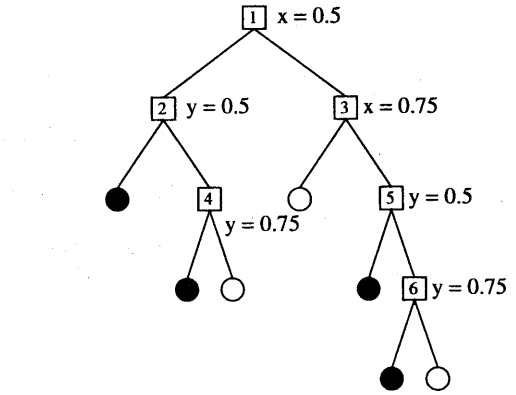


図１　軸平行バージョンのG-PACTを１次元認識問題に適用した例



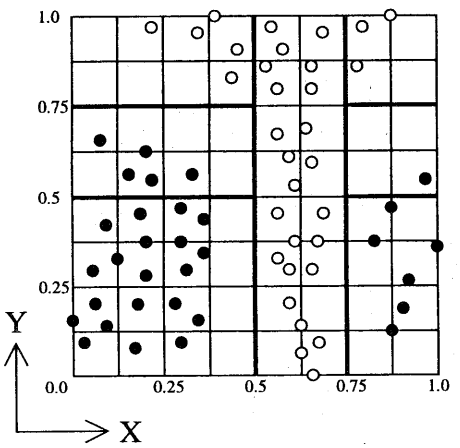
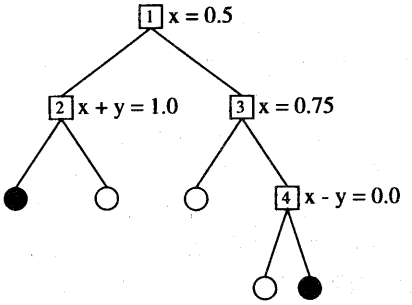


図２　軸平行バージョンのG-PACTを
２次元認識問題に適用した例



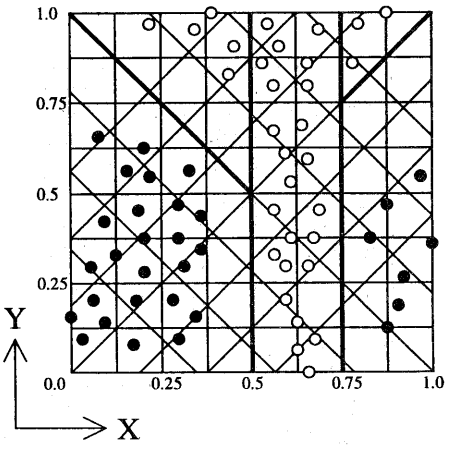


図３　斜平面バージョンのG-PACTを
２次元問題に適用した例