研究論文

疎な関数表現を与える学習機械を用いたサンプルからの規則抽出*¹早坂 太一*², 臼井 支朗*³国立豊田工業高等専門学校情報工学科*²理化学研究所脳科学総合研究センター*³

Logical Rule Extraction Using a Learning Machine with Sparse Function Representation

Taichi Hayasaka*² and Shiro Usui*³Department of Information and Computer Engineering, Toyota National College of Technology*²RIKEN Brain Science Institute*³

We propose a novel rule extraction algorithm adopting the mathematical model called Basis Pursuit (Chen, et al. 1998), where it is represented by a linear combination of kernel functions (similar to MLP or Support Vector Machines) but gives sparse function representation compared to those models. In this algorithm, a number of logical rules are set to the kernel functions in advance. Applying a linear programming method, we obtain classification rules as a small subset of them. If less logical rules are discovered, they will be the core knowledge in a database. We applied our algorithm to several known benchmark problems and its effectualness is verified.

1. ま え が き

医療や経済学の分野において、IT技術の発展により比較的容易にサンプルが集積される昨今では、そのようなデータベースから如何にして有用な知識を得るかが重要な問題であると言える。それに対して、従来よく用いられる方法の一つは、複数の入力属性値からカテゴリ分類を行う数学モデルを用いた統計的手法であり、その代表として、中間層および出力層ユニットの特性関数としてシグモイド関数を用いた階層型ニューラルネットが挙げられる。ネットワークの学習、すなわちパラメータである結合重み値の推定を行った後に、「有用な知識」すなわちどのような規則に基づいてサンプルが分類されたのかを知る問題は、ネットワークからの規則抽出 (rule extraction) と呼ばれ、規則抽出用のネットワーク構築に関する方法や、ネットワークから規則を抽出するアルゴリズムがこれまでに数多く提案されている¹⁻⁸⁾。

数学モデルによる統計的手法を用いた規則抽出法

は、大量のルールが生成され、連続量の扱いに弱く、さらに欠損値に対するロバスト性も低い⁹⁾といった従来の機械学習アルゴリズム¹⁰⁾に代わる新たな手法として注目された¹¹⁾。しかしながら、モデル選択やモデルパラメータ推定における最適化問題など、実問題に対するデータマイニング手法として応用されるためには、解決されなければならない課題は多い。

本研究では、従来の統計的手法と同等の表現能力を持つと考えられる数学モデル Basis Pursuit¹²⁾を用いたサンプルからの規則抽出法を提案する。Basis Pursuit は、疎な関数表現 (sparse representation) を実現できるモデルとして、主に信号 (または画像) の時間 (空間) 周波数成分への分解または再構成を行うために用いられているが、そのアルゴリズムの単純さゆえに、カテゴリ分類問題に適用することも可能である。本研究では、Basis Pursuit をカテゴリ分類問題に適用して、サンプルからの規則抽出を行うアルゴリズムを新たに提案し、その有用性を数値実験により評価する。

*¹ 2003 年 4 月 16 日受付, 7 月 28 日受理*² 〒471-8525 豊田市栄生町2-1*³ 〒351-0198 和光市広沢2-1

2. 基底関数の線形結合で与えられる 関数表現からのルール抽出

いま, d 個の属性値を持つ入力サンプル $x_i = (x_{i1}, \dots, x_{id})$, $i = 1, \dots, N$ が二つのカテゴリ A または B のいずれかに属すると仮定し, その分類規則を, 次に示すような IF-THEN-ELSE 形式の規則で表現する問題を考える.

IF rule₁ ($x_{i1}, \dots, x_{id}$) **AND/OR**...**AND/OR**
rule_K ($x_{i1}, \dots, x_{id}$)

THEN $x_i \in \text{Category A}$ **ELSE** $x_i \in \text{Category B}$

すなわち, 入力属性値が「どれほどの精度で」分類されるかと同時に, その分類が「どのように」実現されているかを知ることが本研究の主題である. こうした問題に対して, よく用いられる決定木などの機械学習的な規則生成法¹⁰⁾の他に, カテゴリ分類を実現できる数学モデルを構築し, その関数表現を IF-THEN ルールの形式に変換する数理統計的手法が知られている. 代表的な例として, 出力層および中間層の特性関数としてシグモイド関数を用いた3層階層型ニューラルネットにより, 属性値から2カテゴリへの写像を学習させ, 学習後のネットワークから上式における rule_j を抽出する方法が数多く提案されている¹⁻⁸⁾.

階層型ニューラルネットを規則抽出に用いる場合, ユニット数決定問題¹⁰⁾やパラメータ推定量の局所最適値問題などが生じるため, 学習結果から抽出される IF-THEN ルールが常に最適である保証はない. こうした問題を解決する方法として, シグモイド関数:

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-ax - \tau)} \quad (1)$$

のパラメータ a および τ を有限集合内の要素として扱うことが挙げられる. 例えば, 学習機械 SVM (Support Vector Machines)¹⁴⁾ は, τ をある値に固定して, a を入力サンプルの集合 $\{x_1, \dots, x_N\}$ からの実現値とする. 学習は, すべての a の値に対する N 個のシグモイド関数にかかる係数を2次計画法により推定することで行われる. SVM は従来の数学モデルと比較して, カテゴリ分類問題に対して優れた性能を発揮することが知られている¹⁵⁾. しかしながら, 階層型ニューラルネットおよび SVM 共に, シグモイド関数は2値関数の代役と見なせることから, その線形形で実現される入力値から出力値への連続写像により正しいカテゴリ分類がなされたとしても, それを強引に IF-THEN ルールに変換した場合には, 誤った分類結果が得られる可能性がある.

ところで, SVM による2カテゴリの分類は, ある

最適化の条件を満たすような基底関数 $\phi_j(x)$, $j = 1, \dots, M$ により規定される $M \times N$ の行列:

$$\Phi_{NM} \equiv \begin{pmatrix} \phi_1(x_1) & \dots & \phi_M(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_N) & \dots & \phi_M(x_N) \end{pmatrix} \quad (2)$$

を用いて, 属性値からカテゴリラベルへの写像を次式のように実現することと等価である.

$$y_N = \text{sgn}(\Phi_{NM} c_M) \quad (3)$$

ただし,

$$\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (4)$$

SVM の場合, $M = N$ であり, $c_M = (c_1, \dots, c_M)^T$ は近似的に $\sum_{j=1}^M c_j^2$ を最小化するような値として求められる. ここで, $\phi_j(x)$ に制約を設けず, さらに, $\sum_{j=1}^M |c_j|$ を最小化するような c_M を用いて (3) を構成するアルゴリズムが Basis Pursuit¹²⁾ と呼ばれる手法である. 次節では, Basis Pursuit による IF-THEN ルール抽出法を提案し, その有効性を評価する.

3. 疎な関数表現による IF-THEN ルール抽出 アルゴリズムの提案

Basis Pursuit¹²⁾ は本来, 調和解析において, 時間 (または空間) 周波数の局所化という観点から信号 (画像) の分解・再構成を効率良く行うために提案された手法であるが, (3) の形式で用いることにより, パターン分類問題にも適用することができる.

Basis Pursuit は $\sum_{j=1}^M |c_j|$ を最小化するような $c_M = (c_1, \dots, c_M)^T$ を求め, (3) を構成する¹⁾. すなわち,

$$\begin{aligned} &\text{Minimize } \sum_{j=1}^M |c_j| \\ &\text{subject to } y_N = \text{sgn}(\Phi_{NM} c_M), j = 1, \dots, M \end{aligned} \quad (5)$$

$c_j = c_j^{(1)} - c_j^{(2)}$, $j = 1, \dots, M$ とおくと, (5) は以下のように書き直すことができるため, Basis Pursuit は線形計画法を用いて解くことができる.

¹⁾ SVM と対応づける場合, (3) にはバイアス項を付加すべきであるが¹⁴⁾, 基底関数の個数に制約のある SVM と異なり, Basis Pursuit の場合には, バイアス項と等価な $\phi(x_i) = 1$, $\forall x_i$ となるような基底関数を常に用意できるため, ここでは省略している. 後述する数値実験においても, バイアス項と等価な基底関数が必ず設定されている.

$$\begin{aligned}
& \text{Minimize } \sum_{j=1}^M c_j^{(1)} + \sum_{j=1}^M c_j^{(2)} \\
& \text{subject to} \\
& \begin{cases} Y_N(\Phi_{NM}c_M^{(1)} - \Phi_{NM}c_M^{(2)}) \geq \mathbf{1}, \\ c_j^{(1)}, c_j^{(2)} \geq 0, j=1, \dots, M \end{cases} \quad (6)
\end{aligned}$$

ただし、 $Y_N = \text{diag}(y_1, \dots, y_N)$, $\mathbf{1} = (1, 1, \dots, 1)^T$, $c_M^{(1)} = (c_1^{(1)}, \dots, c_M^{(1)})^T$, $c_M^{(2)} = (c_1^{(2)}, \dots, c_M^{(2)})^T$ である。

本研究では、分類誤差 $\|y_N - \text{sgn}(\Phi_{NM}c_M)\| > 0$ を考慮し、(6) をさらに次のように書き直す。

$$\begin{aligned}
& \text{Minimize } \sum_{j=1}^M c_j^{(1)} + \sum_{j=1}^M c_j^{(2)} + C \sum_{i=1}^N \xi_i \\
& \text{subject to} \\
& \begin{cases} Y_N(\Phi_{NM}c_M^{(1)} - \Phi_{NM}c_M^{(2)}) \geq \mathbf{1} - \xi, \\ c_j^{(1)}, c_j^{(2)} \geq 0, j=1, \dots, M \\ \xi_i \geq 0, i=1, \dots, N \end{cases} \quad (7)
\end{aligned}$$

ただし、 $\xi = (\xi_1, \dots, \xi_N)^T$ である。 $C > 0$ は分類誤差の許容値を設定するハイパーパラメータであり、 $C = \infty$ の時、(6) と (7) は等価な問題となる。すなわち、分類誤差は 0 となるが、必要な基底関数の個数は、 C が小さい場合と比較して多くなる。

c_M の要素のほとんどが零となる場合、(3) は疎な関数表現と呼ばれる¹²⁾。サンプル数を N とするとき、 $M > N$ の場合には、高々 $M - N$ 個の係数 c_j は零となるため、 $M \gg N$ のときには必ず疎な関数表現が実現される。ここで、 $|c_j| < 1$ の場合に $|c_j| > c_j^2$ であることから、最適化問題の評価関数を最小化する上で、同じ M の場合に、SVM よりも Basis Pursuit の方がより疎な関数表現を実現しやすいことが知られている¹²⁾。このことは、IF-THEN ルールの組合せ数を抑え、より理解しやすい分類規則を抽出する上で重要な利点である。また、 $\phi_j(x)$ の種類に関する制約がないため、それ自身が IF-THEN ルールである微分可能な 2 値関数を多数用意することで、シグモイド関数を特性関数として用いる際に生じる規則抽出のコストに関する問題点を考慮することなく、カテゴリ分類を行うことができるという特長も有する。

以上の点を考慮し、次に示す Basis Pursuit による分類規則抽出アルゴリズムを提案する。

IF-THEN ルール抽出アルゴリズム：

1. Category A または B のいずれかに属する入力サンプル $x_i = (x_{i1}, \dots, x_{id})$, $i = 1, \dots, N$ に対して、適当な IF-THEN ルール $\text{rule}_j(x_{i1}, \dots,$

$x_{id})$, $j = 1, \dots, M$ を用意する。

例えば、 $X \equiv \{x_{ik}; i = 1, \dots, N, k = 1, \dots, d\}$ とする時、以下に示すように X のすべての要素をしきい値 x_{jk}^* とすることで、 $M = N \times d$ 個の IF-THEN ルールが設定できる。

$\text{rule}_j(x_{i1}, \dots, x_{id})$:

IF $(x_{ik} \geq x_{jk}^*)$ **THEN** $x_i \in \text{Category A}$

ELSE $x_i \in \text{Category B}$,

ここで、 k^* はしきい値となる属性値のインデックスである。

2. IF-THEN ルール $\text{rule}_j(x_{i1}, \dots, x_{id})$, $j = 1, \dots, M$ に対応させる形式で、(3) における $\phi_j(x_i)$ を次式のように定義する。

$$\phi_j(x_i) = \begin{cases} 1 & \text{rule}_j(x_{i1}, \dots, x_{id}) \text{ is valid,} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

3. x_i が Category A に属するならば $y_i = +1$ を、Category B に属するならば $y_i = -1$ を設定する。
4. ハイパーパラメータ $0 < C \leq \infty$ を設定し、最適化問題 (7) を解く。
5. 非零となった係数 c_j に対応する IF-THEN ルール $\text{rule}_j(x_{i1}, \dots, x_{id})$ に対して、以下に示すような、階層型ニューラルネットに対する代表的な規則抽出法の一つである Subset アルゴリズム^{1,4)}を適用し、最終的な IF-THEN ルールを抽出する。

Subset アルゴリズム：

- (i) $c_k > 0$ となるすべての結合重みの組合せの集合を Γ^+ , $c_k < 0$ となるすべての結合重みの組合せの集合 (空集合を含む) を Γ^- とおく。
- (ii) $c^+ \in \Gamma^+$ を選択する。
- (iii) $c^- \in \Gamma^-$ を選択する。 c^- に含まれないすべての負の結合重みの集合を c_*^- として、 c^+ および c_*^- の要素の和 Sum を計算する。
- (iv) Sum > 0 であれば、以下に示すような IF-THEN ルールが生成される。

IF $\text{rule}_{j_1}(x_{i1}, \dots, x_{id})$

AND $\dots$ **AND** $\text{rule}_{j_n}(x_{i1}, \dots, x_{id})$

AND (**NOT** $\text{rule}_{k_1}(x_{i1}, \dots, x_{id})$)

AND $\dots$ **AND** (**NOT** $\text{rule}_{k_n}(x_{i1}, \dots, x_{id})$)

THEN $x_i \in \text{Category A}$.

ただし、 $\text{rule}_{j_1}, \dots, \text{rule}_{j_n}$ は、 c^+ の要素に対応する基底関数 (8) を定義する IF-THEN ルールであり、 $\text{rule}_{k_1}, \dots, \text{rule}_{k_n}$ は、 c^- の要素に対応する基底関数 (8) を定義する IF-THEN ルールである。

- (v) すべての $c^- \in \Gamma^-$ について、(iii) および

(iv) を繰り返す。

(vi) すべての $c^+ \in \Gamma^+$ について, (ii), (iii) および (iv) を繰り返す。生成された IF-THEN ルールは論理和で結ばれる。

ここで, 多数のルールを事前に設定しておくという方法は, 従来の機械学習アルゴリズム¹⁰⁾に近いものとも受けとれるが, ルールを2値関数とにおいて, それらの線形結合として疎な関数表現を実現することで, 大量のルールが生成されるといった機械学習的手法の問題点⁹⁾に対しても有効であることが予想される。

また, 提案法は2カテゴリ分類問題を対象としているが, 実問題では3以上のカテゴリに分類する必要がある可能性がある。簡単な解決法として, 1つのカテゴリに属するサンプルをその他のカテゴリから分類する規則を抽出し, 同じサンプルセットを用いて, それをカテゴリ数だけ繰り返すという手法が考えられる。この場合, それぞれのカテゴリに対して排他的な分類規則が得られる, すなわちすべてのサンプルが唯一のカテゴリに分類される保証はないことは注意を要する。評価関数に排他性を考慮する点については今後の課題としたい。

4. 数 値 実 験

本章では, よく知られたベンチマーク問題を取り上げ, 各入力属性値に相関がない, すなわち1つの属性値に関する IF-THEN ルールの論理和・論理積で表現される分類規則を対象として, 提案法の有用性について議論する。属性値の線形結合で表されるような IF-THEN ルールは, 階層型ニューラルネットや SVM を用いることでよく表現できるが, 属性値間の相関の有無に関わらず抽出される傾向も見られる¹⁶⁾。そうした形式が分類規則として受理できるか否かについては, サンプルが与えられる状況により異なるため, ここでは議論の対象としない。

また, 3以上のカテゴリ分類問題については前節で述べた手法を適用する。その際, 2つ以上のカテゴリに含まれると判別されたサンプルについては, 誤分類例と見なすこととする。

4.1 アヤメの分類問題

パターン分類問題においてよく知られたベンチマークデータの一つとして, Fisher のアヤメのデータ¹⁷⁾と呼ばれるものがある。これは, ガクの長さ (sepal length), ガクの幅 (sepal width), 花卉の長さ (petal length), 花卉の幅 (petal width) という4つの属性値から, Setosa, Versicolor, Virginica という3種類

のアヤメを分類する問題で, サンプル数は1種類のアヤメにつき50ずつ, 計 $N = 150$ である。ここでは, すべてのサンプルを学習に用いた際に, 提案法および階層型ニューラルネットを用いた他の方法で抽出された IF-THEN ルールによる分類結果の比較を行うことで, 提案法の性能について解析する。

すべての属性値の集合 $X = \{x_{ik}; i = 1, \dots, 150, k = 1, \dots, 4\}$ に対して, $\forall x_{jk}^* \in X$ をしきい値として, 以下のような IF-THEN ルールを設定する。

rule_j (*i*-th IRIS with $x_{i1}, \dots, x_{i4}$):

IF ($x_{ik}^* \geq x_{jk}^*$) **THEN** this rule is valid.

サンプル数150に対して属性数は4であるから, 事前に設定できる IF-THEN ルールは $M = 600$ 個となるが, 属性値の重複を考慮すると, 独立な IF-THEN ルールは $M = 123$ 個定義できる。それらに対応するように, (3)における $\phi_j(x_i)$ を前節で述べたように定義する。

まず, Versicolor と他の2種類とを分類する問題を取りあげる。 x_i が Versicolor に属するサンプルであれば $y_i = +1$, それ以外の種類であれば $y_i = -1$ とおく。 $C = 0.5$ とし, 最適化法として SIMPLEX 法 (単体法) を用いて最適化問題 (7) を解いた結果, 以下の4個のみの IF-THEN ルールに対応する $\phi_j(\cdot)$ に掛かる係数が非零となった。

rule₁: **IF** ($x_{i1} \geq 4.3$) **THEN** this rule is valid.

rule₂: **IF** ($x_{i3} \geq 3.0$) **THEN** this rule is valid.

rule₃: **IF** ($x_{i3} \geq 5.0$) **THEN** this rule is valid.

rule₄: **IF** ($x_{i4} \geq 1.7$) **THEN** this rule is valid.

rule₁ から rule₄ に対応する基底関数に掛かる係数はそれぞれ, $c_1 = -1.0$, $c_2 = 2.0$, $c_3 = c_4 = -2.0$ であった。ここで, rule₁ はすべてのサンプルに対して成り立つ。よって, Subset アルゴリズムを適用すると, **IF** rule₂ **AND** (**NOT** rule₃) **AND** (**NOT** rule₄) **THEN** *i*-th IRIS is Versicolor という分類規則が導かれる。したがって, 最終的な IF-THEN ルールは,

IF ($3.0 \leq x_{i3} < 5.0$) **AND** ($x_{i4} < 1.7$) **THEN** *i*-th IRIS is Versicolor.

となる。すなわち, 「花卉の長さが 3.0 cm 以上 5.0 cm 未満, 花卉の幅が 1.7 cm 未満であれば, アヤメの種類は Versicolor である」という分類規則が得られる。この規則に対する誤分類率は 150 サンプル中 3 個 (2.0%) であった。

同様にして, Setosa および Virginica についても, $C = 1.25$ として以下のような IF-THEN ルールが導かれた。

Table 1 Classification results for the Iris Dataset.

	Classification rates	#Rules
KT algorithm ⁽⁴⁾	96.7%	10
Structural learning ⁽⁶⁾	97.3%	10
C-MLP2LN ⁽³⁾	98.0%	8
Proposed	98.0%	7

Iris Setosa:

IF (petal length < 3.0 cm) **THEN** it is Setosa.

Iris Virginica:

IF 2 rules out of (sepal width < 3.1 cm),
(petal length ≥ 5.0 cm),
(petal length ≥ 1.7 cm) are valid
THEN it is Virginica.

これらの規則に対する誤分類例は 0 および 2 個と、非常に少ない結果が得られた。

Table 1 に、階層型ニューラルネットから分類規則を抽出する従来のアルゴリズムと提案法の分類結果を比較したものを示す。表中において、#Rule とは、抽出された IF-THEN ルールに含まれる等式・不等式の合計数である。提案法は、誤分類率および規則の単純性の面から見て、Subset アルゴリズムの一種である KT アルゴリズムを用いた方法⁽⁴⁾や、構造学習に基づく階層型ニューラルネットを用いた方法^(3,6)よりも優れた分類結果を与えている。

4.2 修行僧の分類問題

この問題は、機械学習アルゴリズムのコンテストで用いられたもので、真の IF-THEN ルールが与えられている 3 つの問題から成る^(17,18)。それぞれは、一人の人間に対する六つの属性値のすべての組合せ (432 通り) から成るサンプルの一部から、その人が修行僧であるか否かを分類する問題である。真の分類規則があらかじめわかっているため、抽出される IF-THEN ルールとの比較により、提案法の表現能力について議論することができる。また、学習に用いないテストサンプルも合わせたすべてのサンプルに対する分類誤差から、汎化能力について知ることができる。

属性値が連続値で与えられていたアヤメのデータとは異なり、この問題における各属性値は離散値で与えられる。そこで、最初に設定する IF-THEN ルールを、以下のよう定義する。

rule_j (*i*-th person with $x_{i1}, \dots, x_{i6}$):

IF ($x_{ik_1} = x_{jk_1}$) **AND** ($x_{ik_2} = x_{jk_2}$) **THEN** the *i*-th person is a MONK. ただし、 $x_{jk_1}, x_{jk_2} \in \{x_{ik}; i = 1, \dots, 432, k = 1, \dots, 6\}$ である。独立な IF-THEN ルールは $M = 105$ 個定義できる。

まず、3 つの問題の中の第 1 の問題 (MONK-1) における真の分類規則は、以下のような IF-THEN ルールで与えられる。

IF ($x_{i1} = x_{i2}$) **OR** ($x_{i5} = 1$) **THEN** the *i*-th person is a MONK.

与えられるサンプル数 N は 432 個中 124 である。上記の規則に基づいて、 $y_i, i = 1, \dots, 124$ に +1 または -1 を割り当て、 $C = 1$ としてアヤメの分類問題と同様に提案アルゴリズムを適用すると、以下の IF-THEN ルールに対応する基底関数に掛かる係数が非零となった。

rule₁: **IF** ($x_{i1} = 1$) **AND** ($x_{i2} = 1$) **THEN** this rule is valid.

rule₂: **IF** ($x_{i1} = 2$) **AND** ($x_{i2} = 2$) **THEN** this rule is valid.

rule₃: **IF** ($x_{i1} = 3$) **AND** ($x_{i2} = 3$) **THEN** this rule is valid.

rule₄: **IF** ($x_{i5} = 1$) **THEN** this rule is valid.

rule₅: **IF** ($x_{i5} = 2$) **THEN** this rule is valid.

rule₆: **IF** ($x_{i5} = 3$) **THEN** this rule is valid.

rule₇: **IF** ($x_{i5} = 4$) **THEN** this rule is valid.

それぞれの係数値は、rule₁ から rule₃ が 2.0, rule₄ が 1.0 で、それ以外は -1.0 であった。rule₄ から rule₇ は同時に成り立つことはないことを考慮し、Subset アルゴリズムを適用して整理すると、最終的に **IF** rule₁ **OR** rule₂ **OR** rule₃ **OR** rule₄ **THEN** the *i*-th person is a MONK という IF-THEN ルールが導かれる。これが真の分類規則と等価なことは明らかである。すなわち、すべてのサンプルに対する誤分類例は 0 であった。

次に、以下のような IF-THEN ルールに従う第 3 の問題 (MONK-3) を取り上げる。

IF (($x_{i5} = 3$) **AND** ($x_{i4} = 1$)) **OR** (($x_{i5} \neq 4$) **AND** ($x_{i2} \neq 3$)) **THEN** the *i*-th person is a MONK.

与えられるサンプルは $N = 122$ であるが、誤分類されたものが 6 個 ($\approx 4.9\%$) 含まれている。 $C = 0.5$ として提案アルゴリズムを適用したところ、以下のような IF-THEN ルールが抽出された。

MONK-3:

IF ($x_{i2} \neq 3$) **AND** ($x_{i5} \neq 4$) **THEN** the *i*-th person is a MONK.

学習に用いていないものも含めた 432 個すべてのサンプルに対して 97.2% の精度を持つ分類規則が得られた。この値は階層型ニューラルネットおよびそれに関連した方法による結果 (93.1~97.2%)⁽¹⁸⁾ と同程度である。階層型ニューラルネットから規則抽出を行う強

力なアルゴリズムの一つである C-MLP2LN³⁾では 100% の分類結果が得られているが、33 個ものルールから構成されており、真の分類規則を抽出したとは言い難い。このように、Basis Pursuit を適用することで、誤分類サンプルが含まれている場合でも、それに内在する少数の IF-THEN ルールを抽出することが可能であることが確かめられた。

第 2 の問題 (MONK-2) における真の分類規則は以下のように与えられている。

IF # ($x_{ik} = 1$) = 2 **THEN** the i -th person is a MONK.

ここで # ($x_{ik} = 1$) は、 $x_{i1}, \dots, x_{i6}$ の中で $x_{ik} = 1$ となる変数の個数を表す。この問題は他と異なり、一種のパリティ問題である。したがって、設定した初期 IF-THEN ルールの論理積および論理和の組合せで表現しようとする、比較的多数の分類規則が必要となる。実際、与えられた $N = 169$ のサンプルに対して提案アルゴリズムを適用したところ、未学習のものも含めた 432 個すべてのサンプルに対する誤分類例は 0 となったが、31 個もの IF-THEN ルールに対応する基底関数の係数が非零となった。これらに対して Subset アルゴリズムを適用し、分類規則を抽出することは困難である。このように、初期ルールの設定の仕方によっては、疎な関数表現が得られず、IF-THEN ルールの抽出が困難になる可能性もある。

4.3 甲状腺障害患者の分類問題

次に、ある患者が甲状腺障害 (hypothyroid) を持つか否かを判別する問題¹⁷⁾を取り上げる。患者は not hypothyroid, primary hypothyroid, compensated hypothyroid という三つのカテゴリに分類される。学習用として 3772 人、テスト用として 3428 人分のサンプルが用意されている。属性数は 21 で、うち 15 個が 2 値で表され、6 個は連続値である。連続値で表される属性値についてはそれぞれ 100 個、2 値で表される属性値についてはそれぞれ 1 個、計 615 個の初期ルールを定義した。連続属性値に対しては、以下のような初期ルールを設定した。

rule _{j} (i -th Patient with $x_{i01}, \dots, x_{i21}$):

IF ($x_{ik} \geq x_{jk}^*$) **THEN** this rule is valid ($k^* = 1, 17, 18, 19, 20, 21$).

しきい値 x_{jk}^* , $j = 1, \dots, 100$ は、数値実験環境の制約上、与えられた属性値のすべてを用いることが困難であったため、外れ値を除いてその属性値がほとんど含まれるような境界値を設定し、0 からその値の範囲を 100 等分したものをを用いた。属性値の範囲とそれに含まれる学習用サンプルの割合を Table 2 に示す。

数値実験環境の制約上、3772 中 2200 サンプルを用いて最適化問題を解き、以下のような分類規則を得た。ただし、 $C = 0.75$ とおいた。

primary hypothyroid:

IF (TSH > 0.0059) **AND** (FTI ≤ 0.064)

THEN the patient is primary hypothyroid.

compensated hypothyroid:

IF { (T4U > 0.196) **AND** (0.150 < TT4 ≤ 0.154)

OR (T4U > 0.050) **AND** (TT4 ≤ 0.150) }

AND (TSH > 0.006) **AND** (FTI > 0.064)

AND (on thyroxine = No) **AND** (surgery = No)

THEN the patient is compensated hypothyroid.

上記の IF-THEN ルールを共に満たさないサンプルは、not hypothyroid と分類されるとして、3428 個のテストサンプルを分類した結果を Table 3 に示す。Table 3 より、C-MLP2LN³⁾と比較して、より単純な分類規則により同等の汎化能力が得られていることがわかる。

ここで取り扱ったアヤメ、修行僧、甲状腺障害患者の分類問題すべてについて、最適化問題 (7) を解いて得られた疎な関数表現と、抽出された IF-THEN ルールでは、分類精度に差は見られなかった。これは、分類に不必要な論理式が線形計画法により既に最適化されていることから、Subset アルゴリズムにより分類結果をさらに改善する必要はなかったことを示している。すなわち、Subset アルゴリズムは、関数表現を IF-THEN ルールの論理和・論理積の形式に変換するために用いられているだけであり、提案法の本質は疎な関数表現を得ることにあると示唆される。

Table 2 Threshold values of the initial rules for classification of the Hypothyroid dataset.

Attributes	Ranges of threshold values	Samples to be covered
AGE	$0 < x_{j01}^* \leq 100$	100%
TSH	$0 < x_{j17}^* \leq 1 \times 10^{-2}$	94.3%
T3	$0 < x_{j18}^* \leq 4 \times 10^{-2}$	98.1%
TT4	$0 < x_{j19}^* \leq 2 \times 10^{-1}$	98.3%
T4U	$0 < x_{j20}^* \leq 2 \times 10^{-1}$	99.9%
FTI	$0 < x_{j21}^* \leq 2 \times 10^{-1}$	98.4%

Table 3 Classification results for the Hypothyroid dataset.

	Classification rates	#Rules
C-MLP2LN ³⁾	99.36%	13
Proposed	99.36%	11

5. む す び

本研究では、信号解析において疎な関数表現を得るためのアルゴリズムである Basis Pursuit を用いた、サンプルに内在する分類規則の抽出法を提案した。階層型ニューラルネットをこうした課題に適用する際に問題となる、誤差関数の局所最小値問題や、ユニット数決定問題、シグモイド関数から論理関数への変換などを解決した提案法により、階層型ニューラルネットと同等もしくはそれ以上の分類精度を達成できることを示した。

Basis Pursuit は階層型ニューラルネットと同様に、基底関数の線形結合で表されるモデルであるが、サンプルをよく表現するためには、基底関数内に含まれるパラメータ値の候補を多数用意しておく必要があり、提案法では IF-THEN ルールを事前に設定しておくことに対応していた。このことは、誤差逆伝播学習によりパラメータが半ば自動的に最適化される階層型ニューラルネットと比較して不利な点とも思われる。しかしながら、アヤメや修行僧の分類問題に対する数値実験で示したように、与えられたサンプルの値を利用することで、比較的簡単に初期 IF-THEN ルールを多数設定できる。また、より複雑な分類規則を候補に含めることも可能であるから、サンプルに対する何らかの事前知識を IF-THEN ルールとして表現することにより、より少数かつ適切な規則の抽出が実現できると考えられる。

提案法と関係する機械学習法として、分類率の観点からその有用性が知られている Boosting (AdaBoost^[9]) がある。Boosting は、まず任意の学習アルゴリズムにより弱仮説 (weak learner) と呼ばれる、確率 0.5 の事象として説明されるランダム予測よりは少なくとも良い分類結果を与えるモデルを複数個求め、それらの線形和により最終的な分類結果を表現するアルゴリズムであり、弱仮説が既に得られている状況下では、分類誤差最小化による重み係数の推定は、それらの絶対値ノルムの最小化を概念的に近似していることが知られている。これに対して提案法の場合、弱仮説に相当する初期 IF-THEN ルールは、分類率 0.5 を超えるものである必要はない。さらに、係数の絶対値ノルムを陽に損失関数において、その最小化を線形計画法を用いて厳密に行っているものと解釈できる。提案法と同様に、分類誤差を考慮した上で、係数推定法として線形計画法を用いた Boosting アルゴリズムも提案されているが^[20]、係数のノルムを直接最小化しているわけではない。分類率だけでなく、抽出される

ルール数の観点からも見た場合には、係数の絶対値ノルムを直接最小化することで疎な関数表現を得ることのできる提案法の方が優れていることが予想できるが、Boosting および提案法共に、弱仮説または初期ルールの選択によって分類結果が左右されるため、対等な条件下で性能を比較することは困難であり、詳細な検討については今後の課題である。また、提案法で得られた学習結果を弱仮説として Boosting 法と組み合わせることで、より優れた分類規則を得ることも可能となると考えられる。

昨今の電子計算機における計算能力の発展には目覚ましいものがあり、大規模な線形計画問題を机上の計算機環境で解くことがそれほど困難ではなくなりつつある。したがって、計算時間の観点のみから見た場合には、提案法は大規模なデータベースに対しても適用が可能であると考えられる。一方で、(サンプル数) × (初期ルール数) + (サンプル数)² に比例するメモリ容量が必要となるため、計算機環境に応じて、提案法において事前に設定するルール数に制約を受ける。今後はこうした点も考慮し、大規模な実問題に対する有用なデータマイニングツールとしての性能についても評価していく予定である。

参 考 文 献

- 1) Andrews, R., Diederich, J. and Tickle, A. B. (1995): A survey and critique of techniques for extracting rules from trained neural networks, Knowledge-Based Systems, Vol. 8, pp. 373–389
- 2) Andrews, R. and Geva, S. (2002): Rule extraction from local cluster neural nets, Neurocomputing, Vol. 47, pp. 1–20
- 3) Duch, W., Adamczak, R. and Grabczewski, K. (2001): A new methodology of extraction, optimization and application of crisp and fuzzy logical rules, IEEE Trans. Neural Networks, Vol. 12, No. 2, pp. 277–306
- 4) Fu, L. (1994): Rule generation from neural networks, IEEE Trans. Systems, Man, and Cybernetics, Vol. 24, No. 8, pp. 1114–1124
- 5) Fu, L. and Shortliffe, E. H. (2000): The application of certainty factors to neural computing for rule discovery, IEEE Trans. Neural Networks, Vol. 11, No. 3, pp. 647–657
- 6) Géczy, P. and Usui, S. (1999): Rule extraction from trained artificial neural networks, Behaviormetrika, Vol. 26, No. 1, pp. 89–106
- 7) Thrun, S. B. (1993): Extracting provably correct rules from artificial neural networks, Technical Report IAI-TR-93-5, University of Bonn, Institut für Informatik III

- 8) Towell, G. G. and Shavlik, J. W. (1993): Extracting refined rules from knowledge-based neural networks, *Machine Learning*, Vol. 13, pp. 71-101
- 9) 津本周作, 田中 博 (1997): データマイニングの医療応用, *人工知能学会誌*, Vol. 12, No. 4, pp. 536-543
- 10) Quinlan, J. R. (1986): Induction of decision trees, *Machine Learning*, Vol. 1, pp. 81-106
- 11) Shavlik, J. W. (1994): Combining symbolic and neural learning, *Machine Learning*, Vol. 14, pp. 321-331
- 12) Chen, S. S., Donoho, D. L. and Saunders, M. A. (1998): Atomic decompositions by basis pursuit, *SIAM J. Sci. Comput.*, Vol. 20, No. 1, pp. 33-61
- 13) Hayasaka, T., Hagiwara, K., Toda, N. and Usui, S. (1999): Determination of the number of hidden units from a statistical viewpoint, *Proc. of ICONIP*, 99, pp. 240-245
- 14) Vapnik, V. N. (1999): *The Nature of Statistical Learning Theory*, 2nd Ed., New York: Springer-Verlag
- 15) Schölkopf, B., Sung, K.-K., Burges, C. J. C., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997): Comparing Support Vector Machines with Gaussian kernels to radial basis function classifiers, *IEEE Trans. Signal Processing*, Vol. 45, pp. 2758-2765
- 16) 上田博貴, 石川真澄 (1995): ニューラルネットによる連続値入力データからの規則の発見, *日本神経回路学会第6回全国大会講演論文集*, pp. 24-25
- 17) Blake, C. L. and Merz, C. J. (1998): UCI Repository of machine learning databases. (www.ics.uci.edu/~mllearn/MLRepository.html). Irvine, CA: University of California, Department of Information and Computer Science
- 18) Thrun, S. B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fahlman, S. E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R. S., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., Van de Welde, W., Wenzel, W., Wnek, J. and Zhang, J. (1991): The MONK's problem —A performance comparison of different learning algorithm, Technical Report CS-CMU-91-197, Carnegie Mellon University
- 19) Freund, Y. and Schapire, R.E. (1997): A decision-theoretic generalization of on-line learning and an application to boosting, *J. Computer and System Sciences*, Vol. 55, No. 1, pp. 119-139
- 20) Rätsch, G., Onoda, T. and Müller, K.-R. (2001): Soft margins for AdaBoost, *Machine Learning*, Vol. 42, pp. 287-320