論理モデルとしての多重線形関数空間を用いた定性的命題獲得

月本　洋

①

論理モデルとしての多重線形関数空間を用いた定性的命題獲得

月本　洋

目次

第 1 章

序論

1.1 概要

従来、学習は大きく二つに分けられてきた。一つは記号学習であり、もう一つはパターン学習である。記号学習とは人工知能の帰納学習等の主に非数値的データから定性的命題を獲得する学習である。パターン学習とは多変量解析やニューラルネットワーク等を用いて主に数値データを学習して予測するものである。記号学習の欠点の一つはクラス（被説明変数）が連続値の場合には有効ではないと言うことである。パターン学習の欠点の一つは予測はできるけれど回帰式やニューラルネットが何を学習したかが分からない、即ち予測式のみを求めて定性的命題を求めてないということである。

また記号学習とパターン学習は原理的にも異なる。記号学習の原理は、もちろん種々あって一概に要約するのは難しいが、帰納学習に関しては「帰納」と言う言葉には個個の特殊な事柄から一般的原理や法則を導き出すと言う意味があることからしても、抽象化が主であると言って良い。これに対し、パターン学習の原理は基本的には誤差等の評価量を最小にすることであると言って良い。

本論文では上記の記号学習とパターン学習の欠点を解決することを目指す。即ちクラス（被説明変数）が連続値の場合にも有効であり、ニューラルネットワークおよび回帰式から定性的命題を獲得できるアルゴリズムを提示する。また原理面に関して記号学習のアルゴリズムがパターン学習の原理で得られることを示す。即ちパターン学習の原理が記号学習でも有効であることを述べる。

さて本論文の題は論理モデルとしての多重線形関数空間を用いた定性的命題獲得であるが、まず若干耳なれない言葉である「多重線形関数」を説明する。これは一般的に n 変数で言えば

$$\sum_{i=1}^{2^n} a_i x_1^{e_1} \cdot \cdot x_n^{e_n}$$

である。但し a_i は実数、x_i は変数、e_i は 1 か 0 である。例えば 2 変数で言えば

$$axy + bx + cy + d$$

である。この多重線形関数は各々の変数に関して言えば線形であるけれど全体としては非線形である。

次に「論理モデルとしての多重線形関数空間」について簡単に述べると、多重線形関数空間は関数の定義域が $\{0,1\}$ の時には古典論理の代数モデルであるブール代数の拡張であり (図 1.1 参照)、ユークリッド空間となる。従って多重線形関数空間は非古典論理のモデルになっていると言える。より正確に言えば、多重線形関数空間のある部分集合が、ある非古典論理のモデルになっている。本論文では自明な古典論理の場合を除いては、具体的な非古典論理の構文を提示はしないがこれに関しては別稿で述べる。本論文では定義域が [0,1] でもユークリッド空間になることを示す。

「論理モデルとしての多重線形関数空間」について簡単に説明したので、最後にそれを用いた定性的命題獲得のアルゴリズムについて簡単に説明する。本アルゴリズムの基本は多重線形関数をブール関数で近似することである。多重線形関数をデータから求める手法は種々存在するが、それを重回帰分析で求めるとアルゴリズムは以下のようになる。

1. データを重回帰分析して多重線形関数を得る。

2. 多重線形関数をブール関数で近似する。

定性的命題獲得の研究としては人工知能の帰納学習があるが、それと比較して本アルゴリズムの優位性は以下の通りである。

1. 性能が良い。

2. クラス (被説明変数) が連続値の時でも有効である。

1 項に関しては人工知能の帰納学習の代表的なアルゴリズムである C4.5[33] と実問題で比較実験して本アルゴリズムの方が優秀であることを示す。2 項に関しては人工知能の帰納学習ではクラス（被説明変数）が連続値の時には有効ではないが、本アルゴリズムは有効であり、それを簡単な例で示す。また上記の応用はブール関数で近似する多重線形関数を重回帰分析で求めたのであるが、多重線形関数をニューラルネットで求めることも考えられる。この場合にはニューラルネットワークの構造分析と言う応用になる。すなわち本アルゴリズムはニューラルネットワークの構造分析にも有効である。

本論文の応用を整理すると以下の通りである。

1. 人工知能の帰納学習

2. 数値データからの定性的命題獲得

3. ニューラルネットワークの構造分析

重回帰分析は最小 2 乗法であり、多重線形関数のブール関数近似は準最尤法であるから、これらパターン学習の原理で記号学習のアルゴリズムが得られたことになる。

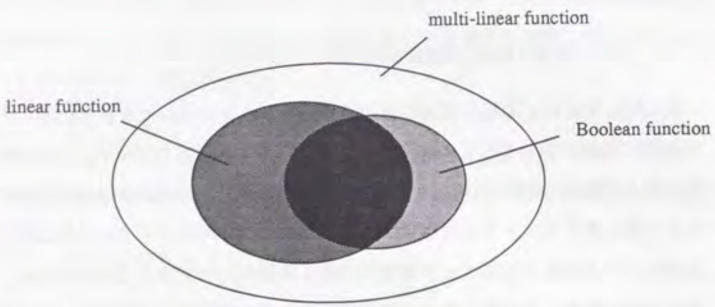

図 1.1: 多重線形関数

1.2 従来の研究

1.2.1 人工知能の帰納学習

近年、人工知能の分野で帰納推論、帰納学習と呼ばれる研究が盛んに行なわれている。人工知能では「帰納推論」は主に形式言語、オートマトン、帰納関数等を正負の事例から求める推論のことを意味するが [47]、これは本論文とはあまり関係ないのでここでは省略する。これに対し、「帰納学習」は各種データから論理式、決定木等を求める推論のことを意味する。即ち帰納学習とは、記号表現された概念集合（例えば命題論理の命題のあるクラス）の中から与えられた事例を基にしてある概念を得ることである [49]。

帰納学習は教師あり学習と教師なし学習に分類される。教師あり学習とは与えられた事例がその概念を満足する正例かその概念を満足しない負例に分類されている場合である。これに対して教師なし学習とは正例、負例に分類されてなく、存在する事例から適当な概念を学習する（見つける）場合である。教師なし学習は概念を事例から形成するので概念形成とも呼ばれる。教師あり学習に関しては多くの研究があり [31]、最も代表的なアルゴリズムは C4.5 であると思われる。

しかしこれらのアルゴリズムはクラス（被説明変数）が連続値の場合には有効ではない。またアルゴリズムの原理は基本的に抽象化であり、具体的には探索を基本にしたアルゴリズムである。これに対し本アルゴリズムは重回帰分析即ち最小 2 乗法と準最尤法であるので多変量解析等のパターン学習のアルゴリズムと同様の原理に基づいている。従ってパターン学習の原理に基づいて記号学習のアルゴリズムが得られることになるが、このことは思想的に有意味であると言える。実行面での、人工知能の帰納学習アルゴリズムに対する本アルゴリズムの新規性、優位性は以下の通りである。

1. 性能（予測精度）が良い。
2. クラス（被説明変数）が連続値の場合にも有効である。

1.2.2 多変量解析

データから何らかの学習をし予測する手法としては統計学の一部門である多変量解析がある [52]。多変量解析の中で最も良く使われている手法である重回帰分析は、通常被説明変数を複数の説明変数の線形関数として表現し、その係数を誤差を最小にするようにして求める。これは被説明変数と複数の説明変数の因果関係を線形と仮定してその蓋然的な関係を求めていると言える。多変量解析で教師あり学習に相当するのは被説明変数のある重回帰分析、判別分析や、外的基準のある数量化 1 類、2 類である。これに対して教師なし学習とは正例、負例に分類されてなく、存在する事例から適当な概念を学習する（見つける）場合である。多変量解析で教師なし学習に相当するのは被説明変数のない主成分分析や外的基準のない数量化 3 類である [73]。

これらの手法の原理は何らかの評価量を最小にすると言うものである。例えば重回帰分析は 2 乗誤差を評価量にしている。しかし上記の手法では予測のみでありどのような事柄を学習したのか分からない。そこで本論文では予測式から定性的命題を獲得するアルゴリズムを提示する。具体的には重回帰分析の回帰式から定性的命題を獲得するものであり、問題としては判別分析、数量化 2 類の問題、即ちデータをいくつかのクラスに分類する問題を扱う。

1.2.3　その他の関連研究

多重線形関数空間は、定義域が $\{0,1\}$ の場合には Linial 等が議論している [22]。彼らは適当なノルムを導入してユークリッド空間になった多重線形関数空間にブール関数を埋め込み、フーリエ変換等を用いた予測学習アルゴリズムを提示してそのアルゴリズムのサンプル計算量等を議論している。これと比較した場合の本論文の主な新規性は以下の通りである。

1. 定義域が $[0,1]$ の場合でもユークリッド空間になる。

2. 仮説生成学習アルゴリズムを提示した。

1.3　本論文の構成

本論文はまず多重線形関数空間について述べる。多重線形関数空間は以下の性質を有している。

1. 古典論理（本論文では命題論理のみを取り扱うので、以降古典論理とは古典命題論理を意味するものとする。）の代数モデルであるブール代数の拡張であり、ユークリッド空間になる。

2. 非独立事象の確率関数の空間に対応している。

最初の性質から説明する。定義域が $\{0,1\}$ の場合には次のように簡単に説明できる。n 変数のブール代数の原子は以下の通りである。

$$a_i = \prod_{j=1}^{n} e(X_j)\ (i = 1 \sim 2^n, j = 1 \sim n)$$

ここで $e(X_j) = \overline{X_j}$ または X_j

例えば 2 変数の場合にはブール代数の原子は $a_1 = XY, a_2 = X\overline{Y}, a_3 = \overline{X}Y, a_4 = \overline{XY}$ である。これらの原子は以下の性質を持っている。

$$a_i \cdot a_i = a_i(\text{単位性})$$

$$a_i \cdot a_j = 0\ (i \neq j)(\text{直交性})$$

$$\sum a_i = 1\ (\text{完全性})$$

である。即ちブール代数の原子は単位直交ベクトルと同様の性質を持っていると言える。例えば $\overline{X} \vee Y$ のベクトル表示は $(1,0,1,1)$ となる。これを論理ベクトルと言う。ここで $XY = (1,0,0,0), X\overline{Y} = (0,1,0,0), \overline{X}Y = (0,0,1,0)$ $\overline{XY} = (0,0,0,1)$ である。また図 1.2 は 2 変数のブール代数の Hasse 図だが、これは 4 次元の超立方体を 2 次元に射影したものと見なせる。ブール代数の原子が単位直交ベクトルに対応することはこの図からも読み取れる。

任意のブール関数はこの原子の線形結合で表現される。すなわち

$$\sum e_i a_i$$

ここで a_i は原子であり、e_i は係数（1 か 0）である。ここで係数を $\{1,0\}$ から実数に拡張すれば線形空間になる。即ちその線形関数は

$$\sum r_i a_i$$

となる。但し r_i は実数である。また関数間に自然に 2 乗距離が入れられる。即ち二つの関数を

$$\sum r_i a_i, \sum s_i a_i$$

とすればその距離は

$$\left(\sum (r_i - s_i)^2\right)^{1/2}$$

である。このように定義域が $\{0,1\}$ の場合には多重線形関数空間がユークリッド空間になるが、本論文では多重線形関数空間が関数の定義域が [0,1] の場合にもユークリッド空間になることを示す [57][58]。定義域が [0,1] の場合には内積の定義が違うだけでノルム、直交関数は同じとなり、同じユークリッド空間になる。

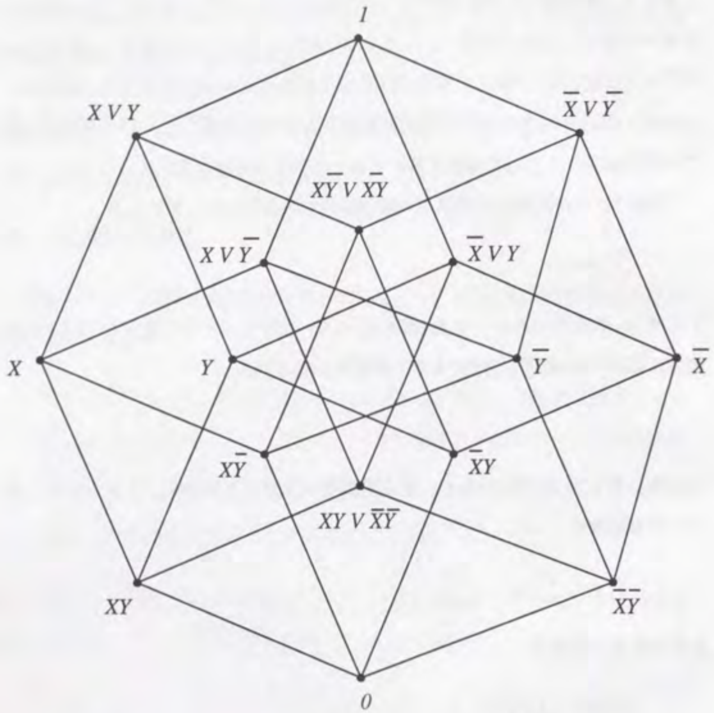

図 1.2: ハッセ図

さて２番目の性質の「非独立事象の確率関数の空間に対応している」は古典論理の論理関数の空間を多重線形関数空間に拡張する際の話しであり、概略は以下の通りである。古典論理計算と独立事象の確率計算は非常に良く似ている。このことは過去多くの人によって指摘されてきたことであるが、具体的にその類似性が調べられた事はない。本論文ではまず古典論理計算と独立事象の確率計算の関係を具体的に調べてその差が独立変数に対する巾等律の有無であることを明確にする。即ち古典論理の代数モデルであるブール代数を初等代数で書き換えたモデルは独立変数に対する巾等律があるのに対し独立事象の確率計算には独立変数に対する巾等律がないのである。この独立事象の確率計算と古典論理計算の対応関係を非独立事象の確率計算に拡張することによって、非独立事象の確率論に対応した非古典論理のモデルを得る。このモデルが多重線形関数空間である。これが「多重線形関数空間が非独立事象の確率関数の空間に対応している。」と言うことである。独立事象の確率計算から非独立事象の確率計算への拡張は干渉項を付加することによっておこなう [60] [66]。多重線形関数空間が非独立事象の確率関数の空間に対応していると述べたが、干渉項の具体的な関数形その他の詳細な議論は今後の課題である。

なお上記のような論理と確率を対置するような考え方は論理確率二元論であり、従来の論理一元論とは異なる。また初等代数による古典論理のモデルは値域、定義域を $\{0,1\}$ から $[0,1]$ に拡張することによって連続値論理になる。この連続値論理の特徴は古典論理の公理を全て満たすことであり、代表的な連続値論理であるファジー論理との比較も行なう [39] [64][69]。

多重線形関数に情報量を定義する。この情報量は無知の等確率、無差別原理 [18] を用いると Shannon が確率分布に定義した情報量と一致することが証明できる。この情報量を用いることによって非古典論理のモデルとしての多重線形関数空間の解析が可能となる。

以上のようにして得られた多重線形関数空間を用いて定性的命題獲得の議論に入る。多重線形関数空間は関数の定義域が $\{0,1\}$ でも [0,1] でも同様に扱え、かつ連続値論理関数が古典論理の公理を全て満たすので、定義域が [0,1] の連続

値論理関数も、以降ブール関数と呼ぶことにする。多重線形関数空間はユークリッド空間であるから、関数の近似が可能である。任意の多重線形関数はそれにもっとも近いブール関数で近似することができる。この近似は情報量を用いると準最尤法であることが証明できる。しかしこの近似法の計算量は指数オーダーであるため、変数の数が多くなると実際的でない。そこで回帰関数が線形の場合の効率的な近似アルゴリズムを提示する。このアルゴリズムでは関数（項）を低次からある次数まで生成する。ある次数で生成を終了すれば計算量は多項式オーダーになる。近似誤差に関しての議論は [22] の結果を用いて行なう。またこのアルゴリズムは閾値論理での閾値論理関数をブール関数で近似するアルゴリズムの拡張になっている [43]。

以上のような近似手法を具体的な 3 種類の問題に適用する。

1. 帰納学習

   多重線形関数空間は $\{0,1\}$ データと [0,1] データを扱えるが、更にダミー変数を導入することによって離散データを $\{0,1\}$ データに還元することによって帰納学習で通常扱うデータを扱えるようにする。このようにして重回帰分析を行ない多重線形関数を得る。この関数をブール関数で近似することによって論理命題を得る。このとき効率的なアルゴリズムを使用する。誤差解析は [15] と [22] の結果を用いて行なう。また比較実験は代表的なアルゴリズムである C4.5 と行ない、本アルゴリズムの方が良好であることを示す。

2. 数値データ

   数値データを重回帰分析して多重線形関数を求め、その関数をブール関数で近似することによって論理命題を求める。

3. ニューラルネットワーク

   ニューラルネットワークの各素子は線形関数と出力関数から構成されている。この出力関数を線形関数で近似し、得られた線形関数をブール関数で近似することによってニューラルネットワークの構造分析を行なう。本論文では排他的論理和の例を示す。

直交関数に関してはブール代数のアトムを拡張したものが基本的であるが、それだけではなく論理回路で展開されてきたスペクトル理論で提出された直交関数 [22] およびそれを [0,1] に拡張した直交関数も意味があり、その直交関数に基づく予測アルゴリズムについても簡単に述べる。

そこで本論文の構成は 2 章ではまず古典論理の初等代数モデルを提示しこれを [0,1] に拡張することによって連続値論理を得る。そして関数空間を拡張することによって多重線形関数空間を得、この関数空間が非独立事象の確率関数の関数空間に対応していることを示す。また思想的背景についても述べる。次にこの関数空間を内積の導入によりユークリッド空間にし、論理命題がベクトル（論理ベクトル）として表現されることを述べる。最後に情報量を導入し、このユークリッド空間内での論理命題の表現を調べる。3 章では多重線形関数のブール関数による近似手法について述べるが、最初に基本的なアルゴリズムを提示してその手法が準最尤法になっていることを説明する。次に回帰関数が線形の場合の効率的なアルゴリズムを提示する。4 章では 3 章で提示したアルゴリズムの帰納学習、数値データ、ニューラルネットワーク、への応用について述べる。またスペクトル直交関数による予測アルゴリズムについても簡単に触れる。

第 2 章

論理モデルとしての多重線形関数空間

2.1 はじめに

本章では 2 節でまず古典論理の代数モデルであるブール代数を初等代数で書き換える。そして 3 節でそのモデルの定義域を [0,1] に拡張し、連続値論理を得る。この連続値論理は古典論理の全公理を満たす。代表的な連続値論理であるファジー論理との比較を行なう。4 節では古典論理の論理計算と独立事象の確率計算の関係を調べ、その差異が古典論理には独立変数に対する巾等律があるのに対し独立事象の確率計算には独立変数に対する巾等律がないことであることを明らかにする。そしてこの独立事象の確率計算と古典論理計算の対応関係を非独立事象の確率計算に拡張することによって、非独立事象の確率に対応した非古典論理のモデルを得る。なお独立事象の確率計算から非独立事象の確率計算への拡張は干渉項を付加することによって行なう。このような論理と確率を対置する考え方は論理確率二元論であり、従来の論理一元論とは異なるので、その辺りの議論を 5 節で簡単に行なう [39] [57][58][60][62] [64][66][69]。

5 節までの議論でブール代数を拡張して多重線形関数空間を得たので 6 節でその性質を調べる。多重線形関数の定義域は $\{0,1\}$ と [0,1] であるが、両者は内積の定義以外は同様に扱えるので、以下の記述で特に断りがない限り定義域はどちらでも良い。序論で述べた通り古典論理は線形空間と類似の性質を有している。具体的にはこのモデルにある内積を導入することによってユークリッド空間とし、論理命題がベクトル（論理ベクトル）して表現されることを述べる。

そして 7 節でこのモデルに情報量を定義する。この情報量は無差別原理 [18] を仮定すると通常の確率分布に定義された情報量と一致することをが証明される。最後に 8 節でこの情報量を用いて論理ベクトルが表現する命題を調べる [57][58] [59][61]。

以下で使う記号は次のとおり。$F, G...$— 論理命題、$X, Y...$— 命題変数、F,G...— 事象、 $f, g...$— 論理関数、 $x, y...$— 変数、$\mathrm{f}, \mathrm{g}...$— 論理ベクトル、$\mathrm{p}, \mathrm{q}...$— 確率ベクトル

2.2 古典論理の初等代数によるモデル

本論文では定義域が $[0,1]$ の場合も取り扱うためまた確率との対応関係を調べるために古典論理の代数モデルであるブール代数を初等代数で書き換える。古典論理の初等代数によるモデルを構築するのであるが、それを以下のような L_1 を定義し、それが古典論理のモデルになっていることを示すという手順で行なう。まず、$\tau(\tau_x)$ という写像を定義して、その性質等について述べる。その後に、モデルの議論に入る。

2.2.1 τ

古典論理（ブール代数）の関数の定義域は $\{0,1\}$ である。実数の範囲で $x^n = x (n \geq 2)$ と $x = 0$ または $x = 1$ は同値である。従って定義域が $\{0,1\}$ ならば $x^n = x (n \geq 2)$ は成立する。この $x^n = x$ に基づく書き換えを写像として定義したのが τ_x である。即ちこの τ_x はブール関数の定義域 $\{0,1\}$ を表現する写像である。

τ_x の定義

今、

$$f(x) = p(x)(x - x^2) + q(x) \tag{2.1}$$

という式を考える。ここで、$f(x), p(x)$ は実係数多項式関数とする。（以下「実係数」を省略する。）また $q(x)$ は 1 次関数である。

τ_x を次の様な写像と定義する。

$$\tau_x : f(x) \to q(x)$$

これは、$f(x)$ から、$f(x)$ を $x - x^2$ で割った剰余への写像である。上記の定義から $\tau_x(x^n) = x$ となる。

τ_x の計算について

$f(x)$ から $q(x)$ を求めるには、実際に $f(x)$ を $x - x^2$ で割ってその剰余を求めれば良いのだが、より簡単に求めるには、次の式が便利である。

$$\tau_x(f(x)) = f(0)(1-x) + f(1)x$$

上式は、式 2.1において、$q(x) = ax + b$ として、$x = 0, 1$ を代入する事より求まる。

τ_x, τ_y に関して、以下の式が成立する。

1. $\tau_x(f(x) \pm g(x)) = \tau_x(f(x)) \pm \tau_x(g(x))$
2. $\tau_x(f(x)g(x)) = \tau_x(f(x)(\tau_x(g(x))))$
3. $\tau_x(\tau_y(f(x,y))) = \tau_y(\tau_x(f(x,y)))$

上式は定義より簡単に求められる。証明は省略する。尚 $\tau_x(f(x))$ の計算をするには、$f(x)$ 中の x^k を x に書き換えれば良い事になる。手計算にはこれが便利であろう。

τ の定義

n 変数の場合、τ を次の様に定義する。

$$\tau = \prod_{i=1}^{n} \tau_{x_i}$$

計算例は次の通り。

$$\tau(x^2 + y + 1) = x + y + 1$$

また τ は次の性質を待つ。簡単に確認できるため、証明は省略する。

1. $\tau(\prod_{i=1}^{n} \tau(f_i)) = \tau(\prod_{i=1}^{n} f_i)$
2. $\tau(\sum_{i=1}^{n} f_i) = \sum_{i=1}^{n} \tau(f_i)$

2.2.2 L_1

L_1 を以下のように定義する。

1. 変数は L_1 の要素である。
2. もし x と y が L_1 の要素ならば, $\tau(x \cdot y)$ と $\tau(x + y - x \cdot y)$ と $\tau(1-x)$ も L_1 の要素である。(以降この計算を τ 計算と呼ぶ。)
3. L_1 は上記の 1 と 2 を有限回適用して生成された関数の全体である。

上記計算にこの τ 計算を有限回適用して生成される多項式関数の全体を L_1 とする。

τ 計算は $\tau(f^2) = f$ を保存する。即ち τ 計算を有限回適用して生成される多項式関数は $\tau(f^2) = f$ を満たす。これは 2.2.3 で証明する。従って $f \in L_1 \to \tau(f^2) = f$ である。

2.2.3 初等代数による古典論理のモデル

モデルの提示

さて、古典論理のモデルであるブール代数に対して以下の対応を与える。

$$F \wedge G \Leftrightarrow \tau(fg) \tag{2.2}$$

$$F \vee G \Leftrightarrow \tau(f + g - fg) \tag{2.3}$$

$$\overline{F} \quad \Leftrightarrow \tau(1-f) \tag{2.4}$$

但し、$\overline{F}$ は否定である。又、τ と多項式関数計算を以後、τ 計算と呼ぶ事にする。

モデルの確認

この L_1 と τ 計算が古典論理のモデルになっているのだが、以下でその確認を行う。ここではブール代数の公理でそれを行う。ブール代数の公理は多くの書物で、様々な形で表現されているが、[3] の公理を採用する。

1. $F \wedge F = F, F \vee F = F$ （巾等律）
2. $F \vee G = G \vee F, F \wedge G = G \wedge F$ (交換律)
3. $F \wedge (G \wedge H) = (F \wedge G) \wedge H,\ F \vee (G \vee H) = (F \vee G) \vee H$ （結合律）
4. $F \wedge (F \vee G) = F, F \vee (F \wedge G) = F$ （吸収律）
5. $F \wedge (G \vee (F \wedge H)) = (F \wedge G) \vee (F \wedge H),\ F \vee (G \wedge (F \vee H)) = (F \vee G) \wedge (F \vee H)$ （モジュラ律）
6. $F \wedge (G \vee H)) = (F \wedge G) \vee (F \wedge H),\ F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H)$ （分配律）
7. $F \wedge O = O, F \vee O = F,\ F \wedge I = F, F \vee I = I$ （普遍限界）
8. $F \wedge \overline{F} = O, F \vee \overline{F} = I$ （補元性）
9. $\overline{\overline{F}} = F$ （対合律）
10. $\overline{F \wedge G} = \overline{F} \vee \overline{G}, \overline{F \vee G} = \overline{F} \wedge \overline{G}$ （ド・モルガンの法則）

τ 計算は 2,3,7,9,10 を満たすが、その他を満たさない。しかし L_1 に属す f は $\tau(f^2) = f$ を満たすので、これを用いてその他の 1,4,5,6,8 を τ 計算が満た す事がわかる。ここでは $\tau(f^2) = f$ を証明して 1,4,5,6,8 のうちの 8 を証明する。他の証明は省略する。

まず $\tau(f^2) = f$ を証明する。変数 x, y は $\tau(f^2) = f$ を満たす。これは τ の定義より明らかである。L_1 の関数は $\tau(fg)$、$\tau(f+g-f\cdot g)$ 、$\tau(1-f)$ により帰納

的に作り出されるので、$\tau(f^2) = f, \tau(g^2) = g$ を仮定して $\tau((\tau(fg))^2) = \tau(fg)$ と $\tau((\tau(f+g-fg))^2) = \tau(f+g-fg)$ と $\tau((\tau(1-f))^2) = \tau(1-f)$ を示せば良い。なお、以下では 2.2.1 で示した $\tau(\prod_{i=1}^n \tau(f_i)) = \tau(\prod_{i=1}^n f_i)$ と $\tau(\sum_{i=1}^n f_i) = \sum_{i=1}^n \tau(f_i)$ を用いる。

$$
\begin{aligned}
&\tau((\tau(fg))^2)\\
=&\tau(\tau(fg)\tau(fg))\\
=&\tau(fgfg)\\
=&\tau(f^2g^2)\\
=&\tau(\tau(f^2)\tau(g^2))\\
=&\tau(fg)
\end{aligned}
$$

$$
\begin{aligned}
&\tau((\tau(f+g-fg))^2)\\
=&\tau(\tau(f+g-fg)\tau(f+g-fg))\\
=&\tau((f+g-fg)(f+g-fg))\\
=&\tau(f^2+g^2+f^2g^2+2fg-2f^2g-2fg^2)\\
=&\tau(f^2)+\tau(g^2)+\tau(f^2g^2)+\tau(2fg)-\tau(2f^2g)-\tau(2fg^2))\\
=&\tau(f)+\tau(g)+\tau(fg)+\tau(2fg)-\tau(2fg)-\tau(2fg)\\
=&\tau(f+g+fg+2fg-2fg-2fg)\\
=&\tau(f+g-fg)
\end{aligned}
$$

$$
\begin{aligned}
&\tau((\tau(1-f))^2)\\
=&\tau(\tau(1-f)\tau(1-f))\\
=&\tau((1-f)(1-f))\\
=&\tau(1-2f+f^2)\\
=&\tau(1)-\tau(2f)+\tau(f^2)\\
=&\tau(1)-\tau(2f)+\tau(f)\\
=&\tau(1-2f+f)\\
=&\tau(1-f)
\end{aligned}
$$

以上で $\tau(f^2) = f$ を証明した。

次に 8 の証明をする。8 は $F \vee \overline{F} = I$ であり、これに対応する式は $\tau(f + (1-f) - f(1-f)) = 1$ である。左辺を計算すると以下の様になる。

$$\begin{aligned}
\tau(f + (1-f) - f(1-f)) &= \tau(f + 1 - f - f + f^2) \\
&= \tau(1 - f + f^2) \\
&= \tau(1) - \tau(f) + \tau(f^2) \\
&= 1 - f + \tau(f^2) \\
&= 1 - f + f \\
&= 1
\end{aligned}$$

従って、$\tau(f + (1-f) - f(1-f)) = 1$ となり、$F \vee \overline{F} = I$ を確認した事になる。以上により、L_1 と τ 計算が、古典論理のモデルになっている事がわかった。簡単な計算例を挙げる。$(X \vee Y) \wedge (X \vee Y) = X \vee Y$ を示す。

$$\begin{aligned}
\tau((x + y - xy)(x + y - xy)) &= \tau(x^2 + y^2 + x^2y^2 + 2xy - 2x^2y - 2xy^2) \\
&= x + y + xy + 2xy - 2xy - 2xy \\
&= x + y - xy
\end{aligned}$$

2.3 連続値論理

前節で得た初等代数モデルの定義域を $\{0,1\}$ から $[0,1]$ へ拡張する。この拡張によって連続値論理を得るが、この連続値論理は古典論理の全公理を満たす。このことは前節の証明が定義域が $\{0,1\}$ であることを使用してないことからわかる。

2.3.1 はじめに

本節では二つの連続値論理の比較を行なう。一つは前節で得た初等代数モデルを [0,1] に拡張して得られる連続値論理であり（以降この節では T 論理呼ぶ）、他の一つはファジー論理 [45] である。定義における違いは、ファジー論理が論理積を $\mathrm{MIN}(f,g)$ で定義し論理和を $\mathrm{MAX}(f,g)$ で定義している（勿論他の定義もあるが最も典型的なのは MAX と MIN である。）のに対し、T 論理が論理積を $\tau(fg)$ で定義し論理和を $\tau(f+g-fg)$ で定義している点である。この定義の違いが以下の差異を生ずる。

1. ファジー論理が古典論理の補元性 ($\overline{f} \vee f = 1$) を満たさない [75] のに対し、T 論理は満たす。（T 論理の方が古典論理をより保存している。）
2. ファジー論理は、複数個の命題の論理和の真理値が 1 に近付かず、直観からずれているのに対し、T 論理は 1 に近付き、直観からずれていない。
3. ファジー論理は主加法標準形が一意に決まらないのに対し、T 論理は一意に決まる。

但し上記の 2 と 3 は 1 から帰結されるものである。以上のように T 論理の方がファジー論理より優れていると言える [39][64][69]。

2.3.2 定義に関するファジー論理との比較

ファジー論理の論理積、論理和、否定は次の通りである。

1. $F \wedge G \Leftrightarrow \mathrm{MIN}(t(f), t(g))$
2. $F \vee G \Leftrightarrow \mathrm{MAX}(t(f), t(g))$
3. $\overline{F} \quad \Leftrightarrow 1 - t(f)$

但しここで $t(f), t(g)$ は f, g の真理値である。

論理演算の定義の仕方に関してT論理とファジー論理を比較すると、T論理の方は論理演算を関数に対して定義しているのに対し、ファジー論理の方は論理演算を真理値に対して定義していると言う違いがある。最大の違いはファジー論理では論理演算を真理値に対する関数(真理値関数)として定義されているのに対し、T論理では論理演算は論理関数に対する作用素として定義されている点である。

以下で連続値論理では論理積、論理和、否定の論理演算を真理値関数として定義することは困難(もしくは不可能)であることを説明する。まず $0.5 \wedge 0.5$ がある一つの値を持つとしよう。この時以下の3命題を考える。

1. $X \wedge X = X$

2. $X \wedge Y$

3. $X \wedge \overline{X} = 0$

$X = Y = 0.5$ を上3式に代入すると以下のようになる。

1. $0.5 \wedge 0.5 = 0.5$

2. $0.5 \wedge 0.5.$

3. $0.5 \wedge 0.5 = 0$

1番目と3番目の式より0.5=0となるが、これは明らかに矛盾である。従って $0.5 \wedge 0.5$ がある一つの値を持つと仮定すると矛盾することになる。しかしながら、$0.5 \wedge 0.5$ が複数の値を持つと言うのもおかしい。このことは $0.5 \wedge 0.5$ と言う式がうまく定義できないと言うことを意味する。これは論理積が真理値関数として定義できないことを意味する。同様にして論理和も真理値関数として定義出来ない[9]。以上の議論より論理積、論理和は真理値関数として定義できないことがわかったが、作用素として定義するとこのような欠点はない。但し論理演算を作用素として定義すると、入力の真理値が決まっても論理和、論理積の値が決まらない。しかしこの性質は欠点ではない。なぜなら入力の真理値から論理積、論理和の値が決まると上で説明したように矛盾するからである。

2.3.3 関数形に関するファジー論理との比較

図2.1にファジー論理の論理積、図2.2にT論理の論理積、図2.3にファジー論理の論理和、図2.4にT論理の論理和を示す。これらの図よりわかる通り、ファジー論理が滑らかでないのに対し、T論理は滑らかである。滑らかであると言うことも好ましい点であろう。

2.3.4 古典論理の公理との関係

T論理が古典論理(ブール代数)の全公理を満たすことは前節で確認した通りである。これに対し、ファジー論理は $F \wedge \overline{F} = O, F \vee \overline{F} = I$(補元性)を満たさない。またMIN, MAX以外にも論理積、論理和の定義があるが古典論理の公理をすべて満たすファジー論理はない。

2.3.5 前項からの帰結

前項でファジー論理が満たさない補元性をT論理が満たすことを述べたが、これよりT論理が、ファジー論理が満たさない以下の二つの事項を満たすと言う長所を有することが帰結される。

1. 複数個の命題の論理和と論理積の真理値に関して我々の直観と合う。

2. 主加法標準形が一意に決まらない。

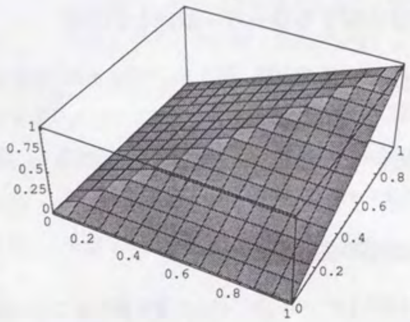

図 2.1: AND (Fuzzy Logic)

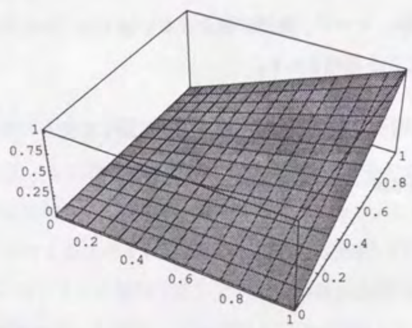

図 2.2: AND (T Logic)

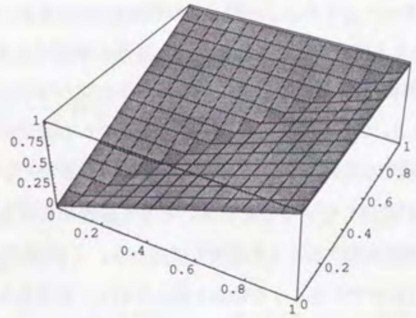

図 2.3: OR (Fuzzy Logic)

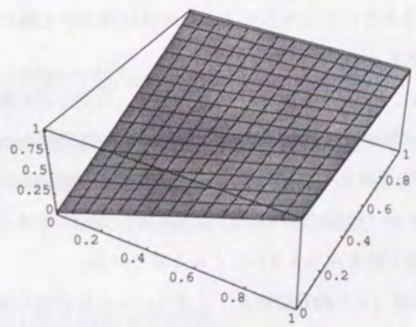

図 2.4: OR (T Logic)

複数個の命題の論理和と論理積の真理値

不確実な命題を考えてみよう。例えばある冒険家が密林で出会った変な動物の名前がわからないとしよう。この時「この動物は A_1 である。」と言う命題の真理値は $[0,1]$ のある値である。これを t_1 としよう。同様にして「この動物は A_n である。」の真理値は t_n となる。そこでこれらの命題の論理和は「この動物は A_1 であるまたは A_2 である....」となる。さてここでこの論理和の真理値を考えてみよう。論理和をとるに従い、確かさは増えていくと考えるのが自然であろう。即ち真理値が増加していくであろう。そして論理和をとる命題の数をどんどん増やせば真理値は限りなく 1 に近付くであろう。T 論理では論理和を $\tau(f+g-fg)$ と定義したのでこのような性質を持っていて、論理和をとるに従い真理値が増加し徐々に 1 に近付く。例えば $n=1000$ とし、$t_1 = ... = t_n = 0.001$ としよう。T 論理の論理和の真理値 T は $T(A_1 \vee A_2) = 0.001 + 0.001 - 0.001 \times 0.001 \simeq 0.001, ..., T(A_1 \vee ... \vee A_n) \simeq 1$ となる。それに対し、ファジー論理では論理和を $\mathrm{MAX}(f,g)$ と定義したため、いくら論理和の命題の数を増やしても 1 に近付くことはない。即ち $T(A_1 \vee ... \vee A_n) = 0.001$ である。このようにファジー論理は常識的に受け入れられないような非現実的性質を有している。これは大きな欠点であろう。そしてこのような欠点は補元性を満たさないことから来ていることである。

これと同様のことが論理積についても言える。今度は逆に論理積の命題の数を増やして行けば限りなく矛盾に近付くから、その真理値も 0 に近付くことになる。T 論理は論理積を $\tau(f,g)$ と定義したので論理積の命題の数を増やせば 0 に近付くが、ファジー論理は論理積を $\mathrm{MIN}(f,g)$ と定義したため 0 に近付くことはない。これも補元性を満たさないことから来ている。

なお、同じ命題（その論理関数を f とする。）の複数個の論理積、論理和については T 論理では $\tau(ff) = f$ と $\tau(f+f-ff) = f$ より真理値は元の命題の真理値と同じままである。

主加法標準形

古典論理では主加法標準形が一意に決まるが、ファジー論理では補元性を満たさないことから主加法標準形が一意に決まらない [75]。これに対し T 論理は補元性を満たすため主加法標準形が一意に決まる。

2.3.6 T 論理とファジー論理の比較のまとめ

以上の議論より T 論理の方がファジー論理より好ましい連続値論理であると言える。ファジー論理と比べて優位な点は以下の通り。

1. 補元性を満たす。
2. 無限個の命題の論理和の真理値が 1 に近付き、常識的に妥当である。（ 1 よりの帰結）
3. 主加法形式が一意に決まる。（1 よりの帰結）
4. 論理演算が滑らかである。

なお T 論理では変数に対する巾等律が重要な役割を果たしていることに注目してもらいたい。

2.4 多重線形関数空間への拡張

前節では定義域を拡張したが、本節では関数空間を拡張する。この拡張は論理計算と確率計算の対応関係を手がかりにして行なう。その結果多重線形関数空間が得られる。なお本節で古典論理と言う時は定義域は $\{0,1\}$ と $[0,1]$ の両方を含む。

2.4.1 確率計算との形式的対応

独立事象の確率計算は以下の通りである。（以下本節では「独立事象の確率計算」を「確率計算」と略記する。）ただし、p は確率を意味する。

1. $p(\mathrm{F} \cap \mathrm{G}) = p(\mathrm{F})p(\mathrm{G})$

2. $p(\mathrm{F} \cup \mathrm{G}) = p(\mathrm{F}) + p(\mathrm{G}) - p(\mathrm{F})p(\mathrm{G})$

3. $p(\mathrm{F}^c) = 1 - p(\mathrm{F})$

式 2.2, 2.3, 2.4の論理計算と上記の確率計算を比較して見ると，論理計算の f, g と確率計算の $p(\mathrm{F}), p(\mathrm{G})$ を同一視すれば差異は τ という演算の有無だけである。したがって、確率計算に τ を付加すると論理計算になり、論理計算から τ を削除すれば確率計算になるという関係が存在することが分かる。命題、事象を一般化して対象と呼ぶ。また独立変数に対応する対象を原子対象と呼ぶ。論理の原子対象は原子命題であり、確率の原子対象は原子事象である。なお、この原子命題はブール代数等の原子 [19] とは別物である。用語に関しては表 2.1を参照。

表 2.1: 対応表

論理	確率	数学	概念
命題	事象	関数	対象
原子命題	原子事象	独立変数	原子対象
真理値	確率	値	

τ は $\tau(x^n) = x$ であり、これは $\tau(x^2) = x$ と同値である。従って τ は原子命題に対する巾等律と言える。古典論理は命題全体が巾等律を満たすから原子命題の巾等律が命題全体の巾等律をも意味していることになる。また独立事象の確率から構成される空間はすべて巾等律を満たさない。これは原子事象に対して巾等律を外す事が全事象に対して巾等律を外すことを意味する。即ち論理の方は原子命題に巾等律を付加すれば命題全体に巾等律を付加したことになり、確率の方は原子事象から巾等律を外せば事象全体から巾等律がはずれることになる。

2.4.2 対応の解釈

上記の原子対象に対する巾等律（τ）の有無の関係は形式的なものであるが、確率計算が事象の付値である確率に関する計算であり、論理計算が命題の付値である真理値に関する計算であることを考えると、この関係を事象と命題の最も基本的な対応関係と考えて良いと思う。以下ではこの対応関係の意味を考える。

確率論ではサイコロ投げみたいな偶然的な対象（事象、事実と言う言葉の方が確率には適当かと思うが論理との比較のためにここでは対象と言う言葉を用いる。表 2.1 参照）を扱ってきた。しかし我々人間にはその対象が偶然的であるか必然的あるかはあらかじめ分かっていない。サイコロ投げなどの試行は偶然的と考えても良いと思うが、確率論はこのような試行だけではなく、自然現象や社会現象もその対象とする。自然現象等は偶然的とも必然的とも言える面を有しているため、その自然現象が偶然的であるか必然的であるかは我々人間には分からない。従って「確率論は偶然的な対象を扱っている。」というよりは「確率論は対象を偶然的であるとみなしている。」と言うべきであろう。従って確率論的な見方をすれば、例えば「地球が丸い」という対象も、各時点毎に独立に生起している偶然的な対象と考えることになる。このような考え方は、我々の常識からすると不自然であるが、ある対象が確率論の対象として不適切であるか確率論の対象になり得るかは別問題であり、上記のような例でも確率論的見方が可能であると言うことである。ここでは確率論が対象を偶然的とみなすと言う立場に立っていると言うことを簡単に確認した。この立場は言い換えると同じ対象が時間的に前後して起こってもそれらの対象は相互に無関係であると言う立場に立っていることである。即ちそれらの対象間に時間内の同一性を認めない立場である。これが確率計算で対象に対して巾等律が成立しない理由であり、τ が無い意味である。

一方古典論理は、基本的には数学を対象としてきた。数学的対象の存在は、多くの数学者が物理的存在とは異次元の存在であると考えているように、時間

的制約を受けず、それゆえに必然的存在でもある。しかし古典論理の対象は数学だけではなく、それ以外の現象も対象である。確率論の対象が偶然的であるのではなく、確率論が対象を偶然的であるとみなしていると述べたのと同様の考え方をして、ここでも古典論理の対象が必然的であるのではなく、古典論理が対象を必然的とみなしていると言う立場をとる。古典論理の立場に立てば、例えばある命題が正しければ、未来永劫正しいのであり、ある時点で偽となる様なことはないと言うことになる。数学的対象はこのような性質を有していると思われるが、それ以外の一般の自然現象、社会現象がこのような性質を有しているかどうかは定かではないが、その対象を古典論理的に扱うのが適当であることと、扱うことが可能であることは別である。古典論理が対象を必然的とみなすという事は言い換えると対象の時間内での同一性を認めることを意味する。即ち何度起こるのも一度起こることと同じであるということであり、「起こる」という表現を用いるよりは「である」という表現を用いる方が適当である。即ち時間内で発生しているのではなく非時間的存在が時間内に現れているということである。これが論理計算で命題に対して巾等律が成立し、τ が有る意味である。

以上見たように τ の有無は確率論と古典論理の立場の違いであり、その違いを要約すると確率論は対象を偶然的とみなすのに対し、古典論理は対象を必然的とみなすという違いである。なお、いままで古典論理と確率論の立場の違いを「必然的」「偶然的」と言う概念で説明したが、これらは存在論的な概念であるから、厳密に言えば、それらに対応する認識論的な概念である「確実的」、「不確実的」で古典論理と確率論の立ち場を述べねばならない [50]。即ち、古典論理は確実的に対象を措定し、確率論は不確実的に対象を措定すると。また古典論理も確率論もこの世界のモデルであるのだが、古典論理は時間内の同一性をすべての対象に要請し、確率論は時間内の同一性をすべての対象に要請しないと言う両極端のモデル化であるとも言える。なお、独立事象の確率計算は古典論理の代数モデルであるブール代数から巾等律を外したものである。巾等律は形式体系で言えば contraction であるから、独立事象の確率計算は contraction

の無い論理のモデルとなる [14]。今まで本節で述べて来た古典論理と独立事象の確率の対応関係は論理モデルと物理的外界との関係の最も基礎的事項といえる。

2.4.3 非独立事象の確率計算

事象 F と G が独立でない時、F と G の積事象の確率 $p(\mathrm{F} \cap \mathrm{G})$ は $p(\mathrm{F})p(\mathrm{G})$ ($p(\mathrm{F}), p(\mathrm{G})$ は各々 F, G の確率) ではなく、一般に $p(\mathrm{F})p(\mathrm{G}) + r$ となる。この r は $r = r(\mathbf{x})$ ($\mathbf{x}$ は複数の独立変数の略記) である。

$r(\mathbf{x})$ は種々の関数形が考えられるが、ほとんどの関数は多項式で近似できるから、$r(\mathbf{x})$ は $\mathbf{x}$ の多項式で近似できるものとして良いであろう。即ち独立変数の多項式で表現されるとして良いであろう。従って、以下では $r(\mathbf{x})$ は独立変数の多項式で表現されるものとする。具体的な関数形としては、例えば 2 変数では $r(x, y) = 0.5(x + y)$ 等が考えられる。和事象、余事象に関しての確率計算は以下の通りである。

$$p(\mathrm{F} \cup \mathrm{G}) = p(\mathrm{F}) + p(\mathrm{G}) - p(\mathrm{F} \cap \mathrm{G})$$

$$p(\mathrm{F}^c) = 1 - p(\mathrm{F})$$

これを整理すると以下の様になる。

1. $p(\mathrm{F} \cap \mathrm{G}) = p(\mathrm{F})p(\mathrm{G}) + r(\mathbf{x})$
2. $p(\mathrm{F} \cup \mathrm{G}) = p(\mathrm{F}) + p(\mathrm{G}) - p(\mathrm{F})p(\mathrm{G}) - r(\mathbf{x})$
3. $p(\mathrm{F}^c) = 1 - p(\mathrm{F})$

積事象が独立変数の多項式で表現されるのだから、上式より、和事象の確率も余事象の確率もやはり同様に多項式で表現されることになる。従って、一般には非独立事象の確率関数は独立変数の多項式で表現されることになる。

2.4.4 非古典論理の論理関数の空間

独立事象の確率関数の空間と古典論理の論理関数の空間 (L_1) が対応しているように非独立事象の確率関数の空間に対応する非古典論理の論理関数の空間があると考えるのは自然であろう。古典論理の論理関数の空間が独立事象の確率関数の空間に独立変数に対する巾等律を付加して得られたのであるから、非古典論理の論理関数の空間が非独立事象の確率関数の空間に独立変数に対する巾等律の付加によって得られるとするのが最も妥当であろう。前項で見たように非独立事象の確率関数は一般には独立変数の多項式で表現される。それに独立変数に対する巾等律を付加して得られる論理関数の空間は多重線形関数空間である。2 変数で言えば $axy+bx+cy+d(a,b,c,d$ は実数) となる。従ってこの多重線形関数の空間が非古典論理の論理関数の空間であると言える。以上の議論より、非古典論理のモデル (数学的構造) が得られたことになる

この空間 (L) での論理計算は以下のようになる。

1. $F \wedge G \Leftrightarrow \tau(fg + r(\mathbf{x}))$
2. $F \vee G \Leftrightarrow \tau(f + g - fg - r(\mathbf{x}))$
3. $\overline{F} \quad \Leftrightarrow \tau(1 - f)$

今までの議論によってこの L が非独立事象の確率関数の空間に対応する非古典論理の論理関数の空間であることが分かった。この計算は独立事象も包含していて、独立事象の場合 $r(\mathbf{x}) = 0$ となり、古典論理の計算と一致する。一般に $r(\mathbf{x})$ を具体的に決めるのは、相関係数等を正確に決めるのが難しいために、容易ではないであろう。今までの議論をまとめると図 2.5のようになる。

2.4.5 L_1 と L の関係

$\tau(f^2) = f$ の L における意味を 1 変数で簡単に調べてみる。$f = ax + b$ として $\tau(f^2) = f$ に代入して $\tau((ax+b)^2) = ax + b$ を解くと 0, 1, x, $1 - x$ が解となり、各々矛盾, 真理 (トートロジ), 或る命題, 或る命題の否定と対応付けら

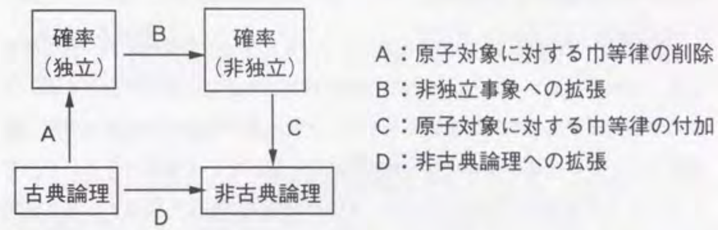

図 2.5: 論理と確率の対応

れる。1 変数の古典論理の命題は上記の 4 個であり、古典論理の命題と $\tau(f^2) = f$ が対応している事がわかる。これを 2 変数以上で実施しても同様の結果が得られるが、計算が繁雑になる為、省略する。以上は $(f \in L) \wedge (\tau(f^2) = f) \Rightarrow f \in L_1$ となり、2.2.3 の $f \in L_1 \Rightarrow \tau(f^2) = f$ と併わせて、$(f \in L) \wedge (\tau(f^2) = f) \Leftrightarrow f \in L_1$ となる。$\tau(f^2) = f$ はブール代数で書けば $F \wedge F = F$ となり、巾等律を意味し、又、L_1 は 2.2.3 より古典論理の命題と対応している。従って、L の $\tau(f^2) = f$ (巾等律) を満たす部分集合は、L_1(古典論理の命題) である事がわかった。

L は巾等律を満たす L_1 と巾等律を満たさないそれ以外の部分から構成されるから、L_1 から L への拡張は巾等律を部分的に外して得られる事になり、ある弱い論理のモデルになっている。弱い論理は線形論理 [13] を始めとしていくつか提案されているが [14][27] [53]、それらは contraction や weakening 等を外すことによって得られる論理である。この場合は contraction を部分的に外すことによって得られる論理である。古典論理を推論規則等を外すことによって拡張する方法は種々存在する。多くは数学的動機で拡張が行われ、論理的、実際的意味づけがなされないか、もしくは後で行われているが、本拡張モデルは確率事象と整合的であることを動機として拡張したので、その拡張によって得られたモデルが非独立事象に対応する論理モデルであると言う明確な意味を有している点が長所である。次にこの拡張モデルの解釈を行う。

2.4.6 拡張モデルの解釈

古典論理では全命題が巾等律を満たした。この事実を時間内の同一性と解釈した。これに対しこの非古典論理の空間では原子命題は巾等律を満たすが、一般の命題は必ずしも巾等律を満たさない。これは原子命題には時間内の同一性があるが一般の命題には必ずしも時間内の同一性がないと解釈できる。このことは人間の思考方法と合致している。人間は時間内の現象を記述する時に時間的に不変の個体を措定し、その個体が時間的に変化すると言う記述方式をする。例えば「彼は身長が伸びた。」という表現で考えてみよう。以前の「彼」は現在の「彼」と物質的には同じではない。しかしわれわれは物質的に同じでない以前の「彼」と現在の「彼」を同一視し、同じ「彼」と言う個体として措定する。そしてその個体が「身長が伸びた」という時間的変化をしたとして、「彼は身長が伸びた」に対応する現象を記述するのである。この時間的に不変と措定された個体（この例では「彼」）が時間内で同一性がある原子命題と対応し、その他の時間的にかならずしも不変でない現象が時間内で同一性のない一般の命題と対応する。時間的に不変な個体とそれを修飾する表現で現象を把握する形式は主語述語形式に通ずるものがあるが、主語述語形式は単に文法概念ではなく人間の思考の論理構造の核心部分でもある [20][29][36]。このようにこの非古典論理のモデルは人間の思考の核心部分を自然に反映した論理モデルとなっている。

2.5 思想的背景

以上論理と確率を対置させながら議論してきたが、これは論理確率二元論であり、従来の主流の考え方である論理一元論とは異なる。ここでは両者の差を議論し論理確率二元論の方が適切であることを述べる。論理一元論の元に人工知能等の研究者が定式化しようとしている問題は人間の経験的命題に関する推論（以降、常識推論と呼ぶ。）である。経験的命題とは一般には経験によって知ることが出来る命題であり、具体的例としては、日常生活の場で言えば天気

に関する命題（雨ならば曇っている）とか科学の場で言えば実験や観測によって得られる命題である。従ってここで取り扱う問題は常識推論と呼ばれている分野と大きな関係がある [62]。

2.5.1 はじめに

人工知能の目的の一つは常識推論を実現することである。常識推論は推論の非単調性等によって定式化されているが、非単調論理等の論理主義的試みは主に数学のために創られた論理学の枠組みであった構文論 - 意味論、論理一元論を見直さずそのまま使用している。ここではこの枠組が常識推論には適していないことを指摘し、この枠組を見直し、新しい常識推論の為の論理的枠組みを提示する。新しい論理的枠組みは構文論 - 事象論 - 意味論と論理確率二元論である。そして事象論として確率論的事象論を提示する。まず 2 項で従来の理論である非単調論理と確率論理を簡単に見て、構文論 - 意味論、論理一元論 [42] と言う枠組が前提されていることを確認する。3 項ではこの枠組は数学には合うが、常識推論には合わないことをいくつかの点で確認する。4 項では新しい枠組である論理確率二元論、構文論 - 事象論 - 意味論を提示する。この事象論 - 意味論は従来の意味論を二つに分けたものであり事象論は外延に、意味論は内包に対応する。そして確率論を事象論として採用することにより確率論的事象論を提示する。確率論的事象論での事象とは確率事象のことである。この確率論的事象論は、従来の確率を論理に還元する確率論理等とは違い、確率論を論理と並置して扱うものである。5 項でこのような考え方に基づく理論展開と応用を概観することで論理確率二元論、確率論的事象論の妥当性を確認する。

2.5.2 従来の理論について

常識推論の定式化の仕方はいくつかある。その一つが論理主義であり、その代表的な理論として Default Reasoning[34], Circumscription[23], Non-monotonic Logic[24] 等のいわゆる非単調論理がある。これらは常識的思考における推論の非単調性を形式的に定式化している。しかしそのような非単調性は常識推論の

本質的な特徴であろうか。良く引合に出される例であるが、「ペンギンは飛べない。」、「ペンギンは鳥である。」、「鳥は飛べる。」を整合的に処理するために、例えば Default Reasoning では「飛べない」と言う事実がなければ「その鳥は飛べる。」と結論する規則を導入している。そしてこれらの規則は話者間の通念や慣習によって決定されるとされているが [23]、それらの規則は事象の観察により多くの基礎をおいており、本質的には話者間の通念や慣習によって決定されるというよりはその観察から決定されると言うべきである。「鳥は大抵飛べる。」の様なすでに分かっている命題だけを考えるならば通念や慣習に基礎をおいても良いが、未知の事象を処理する時はその様なことはできない。その事象を観察することからしかどれが例外で何が蓋然的な規則であるかは決められない。そして「鳥は飛べる」はより正確に言えば「ほとんどの鳥は飛べる。」、「鳥は大抵は飛べる。」等の蓋然的な命題の簡約した表現であり、それらは確率的な、情報の欠如している命題であり、本質的な部分は確率、情報量の観点から扱うべきものと考える。

実際、非単調論理の確率的意味論がいくつか提案されているが [1][28]、彼らも例外や、デフォルトルールを決めるのは通念や慣習よりは統計的事実のほうであると主張して、確率に基礎をおいた意味論を展開している。[28] は定性的確率的推論 と称しており、確率計算の定性的定式化をおこなおうとしている。[1] は確率論理に基づく議論を展開している。ここでは非単調論理によって定式化されている常識推論の本質的な部分は推論の非単調性ではなく確率、情報量の観点から扱うべきであるという立場に立つ。

このような立場に立った時、常識推論は確率論理によって基礎づけられることになる。確率論理に関して簡単に述べる。[26] は可能世界(ブール代数の原子に相当する事象)に確率を割り付けて命題の確率を計算する方法を提案している。[11] はそれを Dempster-Shafer 風に確率の割りつけを点から区間に拡張している。また [5] 等も同様の手法を提示している。その他の手法もこれらと同様、命題に点か区間で確率を割り付けその計算方式を議論しているものである。

これらは古典論理を拡張して確率を論理の枠内で扱おうとするものだが、命

題と事象を同一視し、命題(閉論理式)に確率を割りつけるということをしている。即ち確率と論理を同一範疇のものとして扱っている。これは確率を従来の論理的枠組の中で扱おうとする論理一元論である。

ところで確率論理の命題の確率はどのように与えるのであろうか。確率論理はこのような問題には答えていない。命題に確率を付与する方法を語らずして計算方式だけ詳細に議論しても片手落ちであろう。現実の問題を考えれば命題に確率が付与される前にそれに関連のある何らかの事象上の確率が観察されるはずである。したがって現実的な問題設定は命題から出発するのではなく、事象から出発すべきであり、命題の確率の計算方式と同時に命題の確率を得る方法も議論すべきである。この議論は頻度確率から主観確率を求める問題であり [2]、デフォルトルールを求める帰納推論の問題である。即ち一種の演繹推論である命題の確率の計算方式と同時にその命題の確率を計算する帰納推論も議論すべきである。確率論理で帰納推論が軽視されてきたが、その背景には帰納推論を扱えない構文論 - 意味論という枠組を前提にしてきたからだと言える。望ましい理論は演繹推論と同時に帰納推論も扱えるものと言える。

以上の簡単な議論をまとめると、従来の常識推論の論理主義的な理論である非単調論理の本質的部分は確率的に解釈されるのが妥当であるが、それを基礎づける確率論理にはいくつかの問題があるということになり、この問題の背景には構文論 - 意味論の枠組や論理一元論があると言うことになる。そして望ましい枠組は演繹推論と同時に帰納推論も扱えるものであるが、その枠組を求める前に、次項で構文論 - 意味論の枠組と論理一元論が数学には合うが、常識推論には合わないことを述べる。

2.5.3 数学と常識推論の違い

本項では数学と常識推論の違いを、対象、正しさ、正しさの検証方法、枠組、推論、命題の 6 項目に関して述べる。

数学的対象と常識推論の対象

数学の対象は点とか直線とかであるが、それらは我々人間は見ることも触ることもできないので我々人間の感覚の対象ではなく、人間の直観の対象である。この数学的対象の身分はギリシャ哲学以来常に議論の対象となってきたが、現在にいたっても結論はでていない。数学的対象は存在すると言う者もいるし、それは単に規約にすぎないという者もいる。しかし多くの数学者は数学的対象の存在を認めている。数学的対象は我々の感覚の対象でないのであるから、存在するとしてもそれはこの世界、我々が生きているこの物理的世界ではない。それでは何処に存在するのであろうか。この答の一つとしては Plato の答、イデア界、が最も有名であろう。そして数学に携わる多くの者は自身が意識しているかどうかは別にして実質的にはプラトニストであろう。即ち数学的対象を理念的存在として認め、その理念的存在物が存在する理念的世界を認めているといえよう。これに対して、常識推論は、人間の日常的な推論を模倣しようとするのであるから数学的対象みたいな理念的存在物を扱うというよりは、我々が生きているこの物理的世界を扱うのである。従って常識推論の対象は数学の対象とは違い、理念的ではなく、現実的なのである。内包、外延と言う観点からみると数学的対象は内包だけで外延を持たないのに対し、通常の対象は内包と外延の両方を持つと言える。

数学的正しさと常識推論的正しさ

数学ではある命題が正しいか否かはたとえ途中で実験等によって正否の見当をつけるにしても最終的には証明で決着がつく。それは数学的対象が理念的存在であるからである。これに対し常識推論の対象は理念的でなく現実的であるから、常識推論の命題は、なかには証明で決着がつく命題があるかも知れないが、多くはその命題が指し示す事態を観察することによって正否を決定しているといえよう。ところで観察によって完全に正しさが検証できることはない。なぜならばいかなる観察と言えども、将来その観察と矛盾する事実が発生しないことは保証できないからである。従って常識推論の命題が観察によって検証されるのは「正しい」ではなく「ほぼ正しい」になる。上記の簡単な議論より、数学の主要な問題は「正しい」と「証明できる」であるのに対し常識推論の主要な問題は「正しい」ではなく「ほぼ正しい」であり、また「証明できる」ではなく「観察、実験等によって検証できる」であろう。

「正しい」と「証明できる」は数学の二つの独立した特徴ではなく一つの特徴の二つの側面である。即ち、数学的行為とは「正しさを証明する」と表現でき、その行為の目的語が「正しさ」であり、その行為の動詞が「証明する」と言える。従って「正しさ」は「正しさを証明する」という数学的行為の目的的側面であり、「証明する」はその行為の動詞的側面である。そして基本的には「正しい」のみが証明されるのであって、「ほぼ正しい」は証明されないし、もしくは証明できないのである。（確率論的定理にしてもその定理の「正しさ」自体は確率的ではない。）証明されるのは「正しさ」だけであって「蓋然的な正しさ」は決して証明と言う行為の対象ではないし対象になり得ないのである。

これに対し常識推論は先ほど確認したように「ほぼ正しい」と「観察実験等で検証できる」が特徴である。この二つの特徴も各々独立した特徴ではなく、常識推論の特徴の二つの側面である。常識推論の特徴は数学的行為と対比させて言えば「観察実験等によって蓋然的正しさを検証する。」と言えよう。蓋然的正しさはこの行為の目的語であり、観察実験等はこの行為の動詞的側面である。「ほぼ正しい」-「観察実験等で検証する」は常識推論の不可分の概念の組み合わせである。「蓋然的正しさ」は基本的には証明されないし、「観察実験等で検証できる」のは「蓋然的正しさ」だけであり、「正しさ」ではない。

常識推論的正しさは蓋然的であるのに対し、数学の正しさは非蓋然的であり、その検証手段が常識推論では観察実験であるのに対し、数学では証明であるということである。

構文論 - 意味論

このように「正しい」-「証明できる」は数学的行為を特徴づける不可分の概念の組み合わせである。正しさは意味論的概念であり、証明できるは構文論的な概念である。さて周知のように、古来より漠然と信じてこられた「正しいことは証明できる」がK.Godelによって1階述語論理では成立しても自然数論まで拡張すると成立しないことが示された。これは証明が正しさを確かめる手段として完全ではないことを意味するのであるが、「正しい」と「証明できる」の関係が数学において重要であることを物語るものである。言い替えれば、構文論と意味論の関係が数学においては重要であり、またこういう枠組みが数学に適していると言うことでもある。

これに対し常識推論の「ほぼ正しい」は意味論的概念であるが、「観察実験等によって検証できる」は意味論的概念であろうか。現在の構文論 - 意味論の枠組みで言えばこれは明らかに構文論的概念でないから意味論的概念として整理されるであろうが、この分類には無理があり、構文論 - 意味論は常識推論の論理学の枠組みとしては適切でなく、「観察実験等で検証できる」が属すべき**論が必要となってくる。

これは従来の意味論に**論を追加するともいえるが、従来の意味論を二つに分けたとも言える。そしてこの2分割は内包と外延の区別に対応するものである。従来の意味論は内包と外延の両方を扱おうとしていたが、モデル論的意味論等は実質的には内包しか扱ってなかったと言える。(もっともモデル論的意味論のモデルは外延であるとの見解もある。)内包が意味に対応して意味論で扱われるのであるから、外延は**論で扱われることになる。

数学では構文論と意味論の関係が重要であったのに対し、常識推論では**論 - 意味論の関係が重要であると言えよう。これは構文論 - 意味論と言う枠組が外延を持たない内包だけの対象を扱う数学には適しているが外延を持つ対象を扱う常識推論には適していず、新しい**論が必要であることである。その**論はどのようなものなのであろうか。これを構文論の類似で考えてみよう。

構文論は命題の意味を捨象して絵図の部分だけを扱うと言って良い。これと同様に考えれば**論は観察実験データの意味を捨象した部分だけを扱うことになる。

演繹推論 - 帰納推論

以上の議論を推論と言う観点から見る。証明とは公理から定理を導出することであり、演繹推論である。従って、構文論 - 意味論または「正しい」-「証明できる」と言う枠組みは演繹推論を念頭においていると言える。そしてこの枠組は帰納推論を念頭においていないのである。すなわち演繹推論しか念頭においていないのである。演繹推論が証明の手段であるのに対し帰納推論は常識推論での観察実験による検証の手段である。以上のように考えてくると常識推論で重要な働きをする帰納推論が、演繹推論しか考慮せずに創られた構文論 - 意味論と言う枠組みでは処理できないと言うことである。これが新しい**論を必要とする事につながるのである。ところでその帰納推論とは何であろうか。ここで簡単に帰納推論を概観しておこう。

帰納推論の定義は種々あるが、帰納推論を「観察された個々の事例を総括し、それらの事例が導出されうる一般的命題を確立する推論」として良いであろう。帰納推論の有効性はギリシャ時代からAristotle等によって指摘されており、またルネッサンス時代にF. Baconもその著作「新機関」(Novum Organum)で帰納推論の有効性を指摘している。しかし帰納推論を科学の有効な研究法として確立したのはJ.S. Millである。帰納推論は発見の方法としては有効であっても、演繹推論的に見れば特殊から一般を推論することは誤謬であり、これを許す根拠として彼は「自然の斉一性」と言う原理を提出した。その原理により帰納推論を正当化し、自然の斉一性の原理を因果の法則の持つ普遍性に読みかえ、帰納推論とは現象の中にある原因 - 結果の発見を目的とするものとした。しかしその後この自然の斉一性を支持する者はほとんどいない。自然の斉一性のような原理によって帰納推論の演繹化が無理とすれば帰納推論によって得られた命題の確からしさを数量的に測る方法が考えられる。このような考えのもとでR.

Carnap は帰納確率論 [6][7] を提案した。R. Carnap は確率の解釈として最も一般的であった頻度概念に対して論理概念を主張し、命題に確率を付与した。これにより帰納推論によって得られた命題の確からしさを扱かうことによって帰納推論の正当化を考えた。R. Carnap の試みは現在でも続けられているが [72]、それほど成功しているとは言い難い。

以上、見てきたように帰納推論は演繹化できないのであり、このことからも帰納推論に基づく常識推論は構文論 - 意味論の枠組では処理できないのである。その帰納推論は観察実験データから一般的な命題を推論するのであるが、これは一種のデータの整理であると考えられる。命題の意味が意味論に属すのであるから、その命題の意味を捨象した部分は**論に属すことになる。公理から演繹推論で定理を導出することを問題とする数学の命題の意味を捨象した部分を構文論で扱うのに対し、観察事例から帰納推論で一般的な命題を導出することを問題とする常識推論の命題の意味を捨象した部分を**論で扱うと言える。

先験的 - 経験的、分析的 - 総合的

また上記の議論を、先験的 - 経験的、分析的 - 総合的と言う観点から述べてみよう。先験的とはある事柄が経験に先立って知られるということであり、雑な言い方をすれば先天的とほぼ同じである。経験的とは先験的の反対の概念であり、経験によって知られるということである。先験的、経験的の概念は認識論的概念である。総合的とは大ざっぱな言い方をすれば同語反復（トートロジ）ではないと言うことであり、分析的とは同語反復（トートロジ）と言うことである。総合的、分析的の概念は意味論的概念である。

数学的命題は先験的分析的とみなされている。それは数学的概念が経験によっては知り得ないから先験的であり、また数学の定理が公理から演繹されると言う点で同語反復であるから分析的であると言うことである。そして常識推論の命題は経験的総合的である。それは人間が経験しなければ獲得できないと言う点で経験的であり、その命題が同語反復的でないと言う点で総合的であると言うことである。しかし I. Kant 等は数学の命題を先験的総合的とみなしたし [17]、

更に W.V.O. Quine の総合的 - 分析的と言う区分は絶対的でないと言う議論もあるので [30]、このような分類はいまだ議論の余地があるだろうが、ここではいちばん無難である、数学的命題は先験的分析的、通常の常識推論の命題は経験的総合的という分類に従っておく。今までの論理学は主に数学を対象としてきた。即ち先験的分析的命題を対象としてきた。先験的と言うことが数学的対象が理念的であることに対応し、分析的であることが数学では演繹推論がそして証明が重要であることに対応する。それに対し常識推論で扱う命題は経験的総合的命題である。即ち現実的で、経験的に即ち帰納的に獲得される命題であり、それ故演繹推論や証明で正しさが検証されるような分析的命題ではなく実験観察等によって正しさが蓋然的にしか検証されない総合的命題である。本項を要約すれば、表 2.2の様になる。

表 2.2: 比較表

	対象	正しさ	確認方法	枠組	推論	命題
数学	理念的	正しい	証明	構文論 - 意味論	演繹	先験的分析的
常識推論	現実的	ほぼ正しい	実験, 観察	意味論 - **論	帰納	経験的総合的

2.5.4 常識推論のための論理的枠組

前項で常識推論のための枠組が数学の枠組とは違うことを確認し、構文論 - 意味論の枠組が常識推論には適さないことを見た。本項では新しい枠組について述べる。

事象論

前項までの議論より常識推論の論理的枠組への要件は蓋然的正しさの観察実験等による検証であり、これより意味論と**論という枠組みが必要であると言うことである。**論を以後事象論と呼ぶことにする。構文論と合わせると新しい論理の枠組みは構文論 - 事象論 - 意味論となる。これは従来の意味論を意

味論と事象論の二つに分けたことに相当する。二つに分けたことにより従来の意味論よりは精緻な議論を展開することができる。なお構文論は常識推論でも必要である。それは帰納推論で求めた命題から演繹を行なうことがあるからである。事象論 - 意味論と言う枠組みは内容的には特に新しいと言うわけではなく過去何人かの学者によって指摘されてきたものであるが、ここでは G.Frege と L.Wittgenstein の例を挙げる。

G.Frege は Sinn と Bedeutung を区別した [12]。Sinn は内包、Bedeutung は外延に相当する。このような区別は他の学者も指摘しているところであり、その必要性に関しては議論の余地がないと思われる。内包は意味、外延は事象と対応していると言えるので、内包しか持たない数学的対象が意味論でうまく処理できるのに対し内包と外延の両方を持つ通常の対象が意味論だけでは数学的対象ほどうまく処理できないのである。

また L.Wittgenstein（前期）および論理実証主義は事態と命題の対応をその理論の基本に据えた。例えば「命題は現実の像である。」[44]。哲学の運動としては論理実証主義は挫折したが、その基本的なテーゼは科学としては有効であろう。事態、事実、現実と類似の語がいくつかあるが、これらは事象と読み直せるので、ここでは事象という語を用いる。事象という語を用いれば上記のテーゼは「命題は事象の像である。」と言う命題と事象の関係への言及となり、事象論 - 意味論と言う枠組みと対応することになる。

さてこの事象論は観察事例から帰納推論で一般的な命題を導出する常識推論の意味を捨象した部分を扱う。観察事例と命題から意味を捨象すると何が残るであろうか。構文論では命題から意味を捨象し絵を残した。これは証明が絵の変形規則であるから、絵の段階まで捨象したと言える。同様に考え、観察事例と命題から意味を捨象し「データ」を残すことにする。もちろん、さらに捨象すれば絵になるが、帰納推論はデータの整理と見做せ、観察事例と命題はともにデータであるので、そこまで捨象するのである。このように考えてくると帰納論の問題とはデータの変形規則に関する問題になる。そこでデータとは何であろうか。少なくとも絵に何らかの意味を付加したものである。この意味の

付加の仕方によって種々の議論が可能となろう。ここでは次項で述べるようにデータは絵に確率事象の意味を付加したものと考える。

確率論的事象論

意味論にも何通りかの意味論があるのと同様に、事象論にも数種類の事象論が有り得るであろう。今、データを絵に確率論で言う事象の意味を付加したものと見做すとした。即ちデータを確率的に発生した絵とみなすのである。統計データを想起するとここで言うデータの感じがわかると思う。このような事象論を確率論的事象論とよぶ。従って確率論的事象論の事象とは確率事象のことである。これはモデル論的意味論の意味がモデルであることに対応している。

はじめに構文論 - 意味論、論理一元論と言う枠組が常識推論には不適切であると述べた。前項まで構文論 - 意味論の枠組が不適切であることは述べてきたが論理一元論については言及しなかったので、ここで確率論的事象論の立場から確率論を論理的に取り扱おうとしている [1] 論理一元論が常識推論に不適切であることを述べる。それらは確率論を構文論 - 意味論の枠の中で取扱い、古典論理を拡張して確率論を論理に還元しようとする。この枠組の中で意味論に属する真理値、命題と言う概念と事象論に属する確率、事象を同一視している。しかし論理と確率では記号操作と対象の措定の仕方が違うのである。ここでは簡単にその違いを見る。論理の記号操作の基本は下記の論理積、論理和、否定である。但し F, G は命題である。

1. $F \wedge G$
2. $F \vee G$
3. $\overline{F}$

これに対して確率論の記号操作の基本は下記の独立事象の積集合、和集合、補集合の確率計算である。但し $p(\dot{})$ は確率、F, G は事象である。

1. $p(\mathrm{F} \cap \mathrm{G}) = p(\mathrm{F})p(\mathrm{G})$

2. $p(\mathrm{F} \cup \mathrm{G}) = p(\mathrm{F}) + p(\mathrm{G}) - p(\mathrm{F})p(\mathrm{G})$

3. $p(\mathrm{F}^c) = 1 - p(\mathrm{F})$

従来からこの二つの記号操作が良く似ていることは指摘されていたが、筆者はこの二つの記号操作の違いが独立変数に対する巾等律の有無であることを明らかにし、この有無の背景に存在する確率論と論理の立場の違いが確率論が対象を不確実的に措定するのに対し論理は対象を確実的に措定することである事を示した [66]。従って記号操作と対象の措定において論理と確率は違うのであるからこの二つの範疇は同一視すべきではなく区別すべきなのである。

ここでは確率論的事象論の立場に立ち、確率論の諸概念を論理的に取り扱うことをせず、確率論を事象論として意味論に併置し対応づけながら議論する。従って従来の人工知能の論理主義における確率論の取扱い方とは大きな違いがある。このような確率論の扱い方は論理一元論ではなく論理確率二元論である。

この論理確率二元論の思想上の優位な点は上記以外には確率概念の整理ができることである。確率には古典的確率、頻度確率、論理的確率 [18]、主観的確率 [72] と複数の概念がある。上記のような論理確率二元論の立場に立つことにより、事象論に属する確率は頻度確率であり、意味論に属する [0,1] に拡張された真理値が主観的確率であると整理できる。また古典的確率と論理的確率は無差別原理として頻度確率から主観確率への変換の基礎として位置づけられる [66]。

2.5.5 確率論的事象論に基づく理論展開と応用

さて常識推論では或る命題が実験観察等を用いてほぼ正しいことを帰納推論で確認することが重要な問題である。従って意味論 - 事象論の課題の一つはこの問題に答えることであり、これは頻度確率から主観確率または蓋然的な命題を求める問題とも言える。このような問題設定は数学で言うと或る命題が証明を用いて正しいことを演繹推論で確認することが重要な問題であり、その証明、演繹推論に関する議論が構文論 - 意味論の主要な問題であることに対応する。

モデルに対する要件

このような課題に答え得るにはデータを扱う確率論的事象論に対応した意味論がまず必要になってくる。モデル論的意味論で言うと、これは非独立の確率事象に対応した非古典論理のモデルが必要であることになる。このモデルが備えるべき要件は常識推論を扱えることである。これは蓋然的な命題を情報量の観点から扱え、帰納推論が行なえることである。

帰納推論で得られる命題が蓋然的であるとは近似的に正しいと言うことである。従って近似ができるように距離が導入されてなければならない。この距離は、人工知能における従来の論理主義を含む記号主義の大きな欠点であるパターンマッチングに基づく計算量の爆発を解決する手段ともなる。

また論理は情報を伝達する命題の形式に関するものである。従来の論理学に欠けているのは情報という視点である [76]。我々は情報を伝えるのに命題を用いるので命題に情報量が定義されているのが自然であるが、現在までのところ定義されていないのである。特に常識推論の蓋然的な命題を扱う場合には情報量が基本的な役割を果たすのである。

従って構築すべきモデルは非独立の確率事象に対応していて、距離と情報量が定義されている、非古典論理のモデルである。そしてデータから命題を（または頻度確率から主観確率を）帰納推論する具体的方法を提示しなければならない。ここではこれらの課題に答えている。

常識推論の実現

上記の理論は非単調論理が扱う常識推論の問題を扱うことができ [58]、この理論に基づいたツールを作成中である [55]。そのツールでは規則と事例をともに上記のモデル内の論理ベクトルとして表現することにより、規則と事例を統一的に扱うことを可能にしている。その際規則を公理としてではなく不確実なもの（仮説）として扱うことにより、常識推論の一面である推論の非単調性を実現している。

2.6 ユークリッド空間

数学的な議論に入る前に、われわれの自然言語には命題間の距離や命題の情報量が存在することを簡単に見る。まず、命題間の距離や、命題の情報量について例を挙げて見てみよう。例えば次の様な命題を考える。

1. 彼は老人かも知れない。
2. 彼は老人でない。
3. 彼は老人である。
4. 彼は老人であり、男性である。

上記命題間には自然な距離がある。例えば、2よりは1の方が3に近いと言えるし、2よりは4の方が3に近いとも言える。この様に、命題に関して近い／遠いと言う議論ができると言う事は、自然言語の命題の間に自然な距離（位相）がある事を意味していると言えるだろう。命題の情報量に関しては、上記例で言えば、1よりは3の方が情報量が大きいし、3よりは4の方が情報量が大きいと言える。従って自然言語の命題に関して情報量の議論が可能である。この様に自然言語の中には位相（幾何）的性質が存在するにもかかわらず、今までの論理学は、これらの性質を等閑視してきた感じがある。ここで構築する論理関数のユークリッド空間は自然言語が有する上記の位相的、情報論的性格を形式的に実現していると言える。

2.6.1 内積

内積 $<f,g>$ を次の様に定義する。

$$<f,g>=2^n\int_0^1\tau(fg)dx$$

関数の定義域が $\{0,1\}$ の場合は内積 $<f,g>$ を次の様に定義する。

$$<f,g>=\sum\tau(fg)$$

但しこの和は定義域すべてに渡って行う。例えば2変数の場合 (即ち $f=f(x,y)$, $g=g(x,y)$) は $<f,g>=(\tau(fg))(1,1)+(\tau(fg))(1,0)+(\tau(fg))(0,1)+(\tau(fg))(0,0)$ となる。ここで $(\tau(fg))(1,1)$ は $\tau(fg)$ に $x=1,y=1$ を代入した値である。例えば $f(x,y)=g(x,y)=xy$ の時は $<f,g>=<xy,xy>=\sum\tau(xy\cdot xy)=(\tau(xy\cdot xy))(1,1)+(\tau(xy\cdot xy))(1,0)+(\tau(xy\cdot xy))(0,1)+(\tau(xy\cdot xy))(0,0)=1\cdot1+1\cdot0+0\cdot1+0\cdot1=1$ となる。

この内積の定義は以下の内積の性質を満たす。

1. $<f,f>\geq 0, <f,f>=0\Leftrightarrow f=0$
2. $<af,g>=a<f,g>$
3. $<f+g,h>=<f,h>+<g,h>$

2,3 は明らかなので1の証明を行なう。（証明は1変数で行なうが一般性を失わない。）f を

$$f=f_1x+f_0(1-x)$$

とおく。

[0,1] の場合は以下の通り、

$$\begin{aligned}2^1\int_0^1\tau(ff)dx &= 2^1\int_0^1(f_0^2(1-x)+f_1^2x)dx\\ &= f_0^2+f_1^2\geq 0\end{aligned}$$

又、

$$<f,f>=0\leftrightarrow f_0^2+f_1^2=0\leftrightarrow f=0$$

最後の $f_0^2+f_1^2=0\leftrightarrow f=0$ は $f\in L$ 故に成立する。

$\{0,1\}$ の場合は以下の通り。

$$\begin{aligned} <f,f> &= \sum \tau(ff) \\ &= \sum (f_1^2 x + f_0^2(1-x)^2) \\ &= f_0^2 + f_1^2 \ge 0 \end{aligned}$$

また

$$<f,f>=0 \leftrightarrow f_0^2+f_1^2=0 \leftrightarrow f=0$$

である。

2.6.2 ノルム

ノルムの定義を次の様に行う。

$$\| f \| = \sqrt{<f,f>}$$

上記ノルムが下記3条件を満たす事は簡単にわかる。

1. $\| f \| \ge 0, \| f \| = 0 \Leftrightarrow f = 0$
2. $\| af \| = |a| \, \| f \|$
3. $\| f+g \| \le \| f \| + \| g \|$

Lは上記の内積，ノルムに関し、有限な内積空間，即ちユークリッド空間になる。これはLが多重線形多項式関数から構成される事から簡単にわかる。上記ノルムを$N_\tau(f)$と書く。これは、トートロジー（即ち1）であっても、そのノルムが空間の次元に依存する。即ち

$$N_\tau(1) = 2^{n/2}$$

となる。このノルムは直交関数展開の時に便利である為導入したが、ノルムが空間に依存する為相対的ノルムと呼ぶ。$N_\tau(f)$に対して、絶対的ノルムを

$$N(f) = \sqrt{2^{-n}} N_\tau(f)$$

と定義する。このノルムの論理的な意味を簡単に見ると次の様になる。簡単な計算より、

$$(N(1))^2 = 1, (N(x))^2 = 0.5, (N(xy))^2 = 0.25, (N(0))^2 = 0$$

となり、これらの数値は、“命題の充足度”を表していると考えられる。例えばトートロジー(即ち1)はどんな対象に対しても真であるが、xは半分の対象，xyは1/4の対象に対して真であり、矛盾（即ち0）はどんな対象に対しても真とならないと言う様に解釈できる。このノルムは、今後命題の情報量（論理エントロピー）の定義に使用する。

このノルム以外にも関数の近さを評価する量は存在するであろう。関数解析においてもヒルベルト空間、バナッハ空間等の関数空間があり、関数の近さを評価する量はそれぞれ異なる。したがって2乗ノルム(ユークリッドノルム）はそれらいくつか考えられる内の一つであるが、われわれの直観に最も近いので最初にとりあげるにふさわしいノルムであると考える。他のノルムとの比較は将来の課題としたい。

2.6.3 直交関数展開

ブール代数の原子を拡張して直交関数を次の様に定義する。（n変数とする。）

$$\phi_i = \prod_{j=1}^{n} e(x_j) \ (i = 1 \sim 2^n, j = 1 \sim n)$$

ここで $e(x_j) = 1 - x_j$ または x_j

例えば、1変数の場合の正規直交系は$\{x, 1-x\}$となり、2変数の場合の正規直交系は$\{xy, x(1-y), (1-x)y, (1-x)(1-y)\}$となる。上記の様に定義した$\phi_i$に対して、

$$< \phi_i, \phi_j > \ = \ 0 (i \neq j)$$

$$= \ 1 (i = j)$$

$$f = \sum_{i=1}^{2^n} < f, \phi_i > \phi_i$$

となる事は容易に確められる。従って、以下が成立する.

1. 全ての $f \in L$ は $\{\phi_i\}$ により展開される

$$f = \sum_{i=1}^{2^n} < f, \phi_i > \phi_i$$

2. 全ての i で $< f, \phi_i > = 0$ ならば、$f = 0$ である。

以上の事より、n 変数よりなる命題(論理関数)は 2^n 次元のベクトルとして表現できる事がわかる。具体例として $f = x + y - xy$ を挙げる。正規直交系は $xy = (1,0,0,0), x(1-y) = (0,1,0,0), (1-x)y = (0,0,1,0), (1-x)(1-y) = (0,0,0,1)$ であるから、直交展開した時の各係数は以下の様になる。以下の例は $\{0,1\}$ の場合であるが、[0,1] の場合も同様である。

$$< f, xy > = \sum \tau((x + y - xy)xy) = 1$$

$$< f, x(1-y) > = \sum \tau((x + y - xy)x(1-y)) = 1$$

$$< f, (1-x)y > = \sum \tau((x + y - xy)(1-x)y) = 1$$

$$< f, (1-x)(1-y) > = \sum \tau((x + y - xy)(1-x)(1-y)) = 0$$

従って、

$$f = 1 \cdot xy + 1 \cdot x(1-y) + 1 \cdot (1-x)y + 0 \cdot (1-x)(1-y)x$$

となり、ベクトル表示で (1,1,1,0) となる。これは、ブール代数での原子による表現に対応するものになっている。即ちこの直交関数展開は、ブール代数の原子による展開を拡張したものと言える。

2.6.4 その他の正規直交系

正規直交系はブール代数の原子を拡張した $\{x, 1-x\}$ 等以外にも無数にある。それらは直交性と単位性を満たしてさえいれば良いからである。ここでは、古典論理が有している諸性質をなるべく保存して、モデルを拡張する方針を取っているのであるから、当然の事ながらブール代数の原子を拡張した正規直交系による議論を今後も展開する、尚、ノルムは定義よりわかると思うが、使用する正規直交系には依存しない。

2.6.5 ベクトルによる論理計算

独立事象の場合、ベクトルでの論理計算は以下のようになる。

$(\mathrm{f} = (f_i), \mathrm{g} = (g_i))$

1. $F \wedge G(\tau(fg))$ は, 各座標の直交性から $\mathrm{f} \circ \mathrm{g} = (f_i g_i)$ となる。（$\circ$ は各要素の積を意味する。）
2. $F \vee G(\tau(f + g - fg))$ も同様に $\mathrm{f} + \mathrm{g} - \mathrm{f} \circ \mathrm{g} = (f_i + g_i - f_i g_i)$ となる。
3. $\overline{F}(\tau(1-f))$ は $\mathbf{1} - \mathrm{f} = (1 - f_i)$ となる。

非独立事象の場合にはこれに r に対応するベクトル $\mathrm{r} = (r_i)$ が加わることになる。

2.7 情報量 H の導入

確率分布には情報量が定義されているが、命題には現在までのところ情報量が定義されていない。しかし命題が情報を伝達するものであることを考えれば命題に情報量が定義されているべきものと思われる [76]。ここでは従来の確率分布の情報量と整合的な情報量を導入する。

2.7.1 情報量の定義

前出の L を以後論理空間と呼び、その要素を論理関数と呼ぶ。下式の H を情報量として定義する。

$$H(f) = -log_2(N(f))^2 \tag{2.5}$$

但しここで f は論理関数である。この H は命題の持つ情報量を意味すると考えられる。具体例は以下の通りである。

1. $H(1) = 0$ —— トートロジは情報量が 0 ビット。これは、トートロジが恒等的に正しい為、現実世界に関する具体的情報を含んでいない事に対応する。

2. $H(x) = 1$ —— x 即ち「A である」は情報量が 1 ビット。これは、「A である」と言う命題が同時に「A ではない」を陰に否定しており、{「A である」、「A でない」} から「A である」をその他の条件（発生頻度 etc. ）を考慮せずに指示している事に対応する。

3. $H(xy) = 2$ —— xy 即ち「A かつ B である」は情報が 2 ビット。これも同上の解釈が可能であり、{「A であるかつ B である」、「A であるかつ B でない」、「A でないかつ B である」、「A でないかつ B である } から「A である B である」を指示している事に対応する。

4. $H(0) = \infty$ —— 矛盾は情報量が無限大。命題が多くなり、それらを連言して行けば、情報量が多くなるが、一方徐々に命題間で矛盾が発生する可能性が高くなる。その極限状態として、情報量=無限大が矛盾と対応すると解釈できる。

上記例よりもわかる通り、この情報量の定義は妥当と思われる。通常の情報量は事象に定義されているが、この情報量は命題に定義されているので、論理エントロピと命名する。$H(x) = 1$ だが、x に対応する命題 A を「さいころの

目が 3 である。」とすると、通常のエントロピは $log_2 6$ となるが、論理エントロピは $log_2 2$ となり、その値が異る。通常のエントロピはさいころの面が 6 個ある事を知った上で算出するが、論理エントロピはその様な情報が与えられていないとして算出するから、値が異なるのである。さいころの面数に関する情報を命題にして付加すれば、論理エントロピを $log_2 6$ とする事は可能である。即ち論理空間の変数を「さいころの目が 1 である。」〜「さいころの目が 6 である。」の 6 変数にすれば良いのである。(但し、有り得ない状態を除かねばないらないが) 別の言い方をすれば、最初の場合は「さいころの目は 3 である。」と「さいころの目は 3 以外である。」を同等に扱っているのである。論理的推論と言うものは与えられた命題のみを考慮すると言う性質を有するものであり、論理エントロピはそういう性質を浮彫りにしているとも言える。又ノルムが増加すればする程エントロピーは小さくなるから、原点から遠い命題ほど情報量が少いとも言えるし、その逆に原点に近い命題ほど情報量が多いとも言える。

2.7.2 情報量 H と確率の情報量 I は古典論理では一致する

以下にその証明を示す。m 個の要素が 1 であるような論理ベクトル f の場合、論理エントロピは以下のようになる。以下の計算は $\{0,1\}$ の場合であるが、[0,1] の場合も同様である。

$$\begin{aligned}
H(f) &= -log_2(N(f))^2 \\
&= -log_2(2^{-n/2}N_r(f))^2 \\
&= -log_2(2^{-n}\sum\tau(f^2)) \\
&= n - log_2(\sum f) \\
&= n - log_2 m
\end{aligned}$$

確率の情報量 (I) はある事象がわかった時に得られる情報量であり、以下のよう

になる [35]。

$$
\begin{aligned}
I &= n - H_e \\
&= n - (-\textstyle\sum_1^{2^n} p_i log_2 p_i) \\
&= n + \textstyle\sum_1^{2^n} p_i log_2 p_i \\
&= n + m \times (1/m log_2(1/m)) \\
&= n - log_2 m
\end{aligned}
$$

但し、H_e は通常のエントロピであり、p_i は確率であり、n は独立変数の数である。上記の式の変形で 3 段目から 4 段目への変形では無差別原理 [18] を使用している。即ち m 個の各々の事象が発生する確率は、特に情報がないので等確率とし、$p_i = 1/m$ としている。したがって、古典論理の範囲で無差別原理を使用すれば $H = I$ となる。これを以降では非古典論理の命題にも拡張して適用する。

2.8 不確実な命題の表現

2.8.1 情報量に基づく議論

1 変数の場合には論理関数は一般に $ax+b(1-x)$ となる。x と $1-x$ は直交するので、横軸を x とし、縦軸を $1-x$ とすると、情報量 $=1$ の曲線は半径 $=1$ の円弧になる（図 2.6参照）。従って円弧上の点は情報量的に完全である、即ちある明確な事実と対応づけられるが、$(1,0)(=x)$ に対応する命題と $(0,1)(=1-x)$ に対応する命題の中間的な命題を意味すると解釈できる。例えば $(1,0)$ に対応する命題を「その木は高い。」とすれば $(0,1)$ は「その木は低い。」となり、円弧上の点に対応する命題はその中間的な「その木は低くも高くもない。」とか「その木はやや高い。」と言うような命題になる。

点 $(1,b)(0<b<1)$ に対応する命題は、情報量が $(1,0)$ に対応する命題「その木は高い。」と点 $(1,1)$ に対応する情報量の無い命題「その木は高いかも知れないし、高くないかも知れない。」の中間にあるため、「その木はどちらかと

言うと高いであろう。」と言う情報の欠如した命題であると解釈できる。同様に $(a,1)$ は「その木は低いかも知れない。」と言う情報の欠如した命題を意味すると解釈できる。

極座標 (r,θ) を用いれば、r 成分が情報の不完全性を表現し、θ 成分が述語の不確定性を表現することになる。図 2.6の情報量 $=1$ の曲線は事実を完全に指示する命題に対応するのでこの曲線より原点側の点、即ち情報量が 1 以上の点は我々の自然言語では表現できないような「命題」に対応する。自然言語で表現できないような「命題」は命題とは言えないので、この様な点は命題と対応できないことになる。一般的に言えば n 変数の時には情報量が n ビット以上の命題は存在しないと言うことになる。

以上の議論より分かるように、本モデルでは不確実性を述語の不確定性（θ 成分）と情報の不完全性（r 成分）の二つに分けて議論できる。曖昧さを表現する言葉は「不確実」「不完全」「ファジィ」といろいろあり、その用法も人によって異なるのでここで筆者の用法について簡単に述べておく。「不確実」は最も一般的な概念として使い、「不完全」は情報が欠けている時に使い、「不確定」は述語が明確でない時に使う。この「不確定」は「ファジィ」とほぼ同じであろう。

2.8.2 命題の不確実性の評価量

本論文では論理確率二元論に立脚し、確率は事象の付値であり、命題の付値ではないと言う立場に立っていて、命題の確率と言う概念を認めない。その代わりに情報の不完全性と述語の不確定性と言う量で不確実性を評価する。情報の不完全性を評価する量は情報量であり、これは式 2.5より計算できる。述語の不完全性はその様な評価量がないので、ここで次のように定義する。評価の対象になる論理ベクトル（例えば確率ベクトルより得られた論理ベクトル）を f とし、評価の基準になる論理ベクトル（例えばある述語を表現する古典論理ベクトルであり、この基準になるベクトルは古典論理ベクトルの数だけあることになる。）を g とする。この時、g に対する述語確定度は f の g への射影と定義

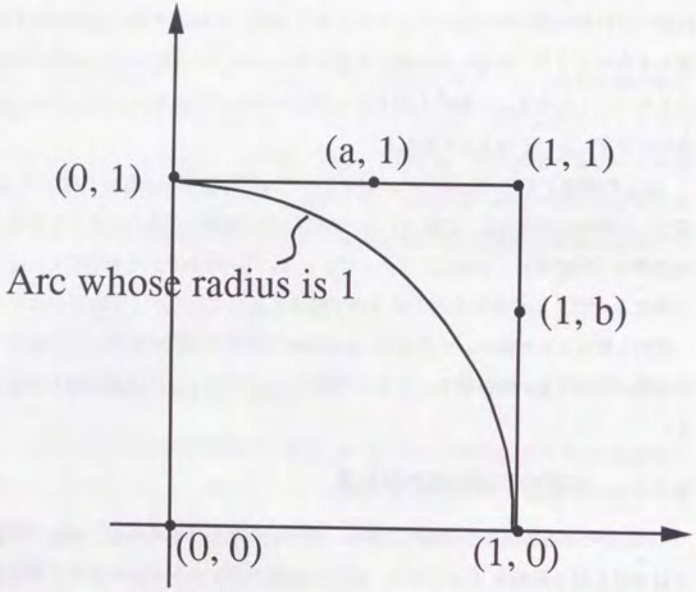

図 2.6: 非古典論理のベクトル（1 変数）

するのが自然であろう。即ち次式で定義される量が g に対する述語確定度と考えて良いであろう。分母は正規化のためである。

$$<f, g> / \| f \| \| g \|$$

以上の議論より、命題の不確実性を表す量は情報量と述語確定度と言うことになった。

2.9 おわりに

本章では 2 節でまず古典論理の代数モデルであるブール代数を初等代数で書き換えた。そして 3 節でそのモデルの定義域を [0,1] に拡張し、連続値論理を得た。この連続値論理は古典論理の全公理を満たした。ファジー論理との比較を行なった。4 節では古典論理の論理計算と独立事象の確率計算の関係を調べ、その差異が古典論理には独立変数に対する巾等律があるのに対し独立事象の確率計算には独立変数に対する巾等律がないことであることを明らかにした。そしてこの独立事象の確率計算と古典論理計算の対応関係を非独立事象の確率計算に拡張することによって、非独立事象の確率に対応した非古典論理のモデル、多重線形関数空間を得た。なお独立事象の確率計算から非独立事象の確率計算への拡張は干渉項を付加することによっておこなった。このような論理と確率を対置する考え方は論理確率二元論であり、従来の論理一元論とは異なるので、その辺りの議論を 5 節で簡単に行なった。

そして 6 節で非独立事象の確率に対応する非古典論理の関数空間である多重線形関数の空間を内積の導入によりユークリッド空間とし、論理命題がベクトル（論理ベクトル）して表現されることを述べた。このことは定義域が $\{0, 1\}$ でも [0,1] でも同じであった。そして 7 節でこの関数空間に情報量を定義した。この情報量は無差別原理を仮定すると通常の確率分布に定義された情報量と一致することを証明した。最後に 8 節でこの情報量を用いて命題の不確実性が情報量と述語確定度の二つの量で評価されることを見た。

第 3 章

定性的命題獲得のアルゴリズム

3.1 はじめに

本章では多重線形関数をブール関数で近似することを考える。多重線形関数空間は関数の定義域が $\{0,1\}$ でも [0,1] でも同様に扱え、かつ連続値論理関数が古典論理の公理を全て満たすので、定義域が [0,1] の連続値論理関数も、以降ブール関数と呼ぶことにする。多重線形関数空間はユークリッド空間であるから、関数の近似が可能である。任意の多重線形関数はそれにもっとも近いブール関数で近似することができる。この近似は情報量を用いると無差別原理の仮定のもとで準最尤法であることが証明できる。しかしこの近似法の計算量は指数オーダーであるため、変数の数が多くなると実際的でない。そこで回帰関数が線形である場合の効率的な近似アルゴリズムを提示する。このアルゴリズムでは関数（項）を低次からある次数まで生成する。ある次数で生成が終了すれば計算量は多項式オーダーになる。近似誤差に関する議論は [22] の結果を用いて行なう。このアルゴリズムは閾値論理での閾値論理関数をブール関数で近似するアルゴリズムの拡張になっている。2 節では近似の一般的議論を行なう。3 節ではこの近似法が準最尤法であることを証明するために必要な準備を行なう。4 節では 3 節で得られた変換式を用いた帰納推論について述べる。5 節では準最尤法の証明を行なう。6 節では効率的なアルゴリズムについて述べる。

なお以下で使う記号は次のとおり。 f, g...— 論理関数, f, g...— 論理ベクトル, p, q...— 確率ベクトル

3.2 近似の一般的議論

多重線形関数空間はユークリッド空間になるため多重線形関数はユークリッド空間内のベクトルで表現される。この論理ベクトルを f とし古典論理ベクトルを g とすると多重線形関数をブール関数で近似することは f を g で近似する事になる。即ち f に最も近い古典論理ベクトル g を捜すのである。f$(=(f_i))$ とし、g$(=(g_i))$ とすると、最小にしたい量は $\sum(f_i-g_i)^2$ となる。各項 $(f_i-g_i)^2$ は独立なので各項を最小にすることが $\sum(f_i-g_i)^2$ を最小にすることと同じになる。g_i は 1,0 のいずれかなので、$(f_i-g_i)^2$ を最小にする g_i は $f_i \geq 0.5$ ならば $g_i=1$ その他で $g_i=0$ となる。従って以下の通りとなる。

$$f_i \geq 0.5 \text{の時は} g_i = 1, \text{その他は} g_i = 0$$

例

$$z = 0.6x - 1.1y + 0.3$$

としよう。これを直交展開すると、

$$z = -0.2xy + 0.9x(1-y) - 0.8(1-x)y + 0.3(1-x)(1-y)$$

となり、ブール関数で近似すると

$$x(1-y), \text{即ち} X \wedge \overline{Y}$$

となる。

3.3 確率と論理の対応と変換式

3.3.1 論理ベクトルと確率ベクトルの基本的対応

情報量 0 の命題はトートロジである。例えば $X \vee \overline{X}$。この情報量とは式 2.5で定義した情報量である。$X \vee \overline{X}$ のベクトル表示は (1,1) である。一方、情報量 0 の確率分布は (1/2, 1/2) である。この情報量とは I である。 ここで、命題で最

も情報量の小さいものと事象で最も情報量の小さいものを対応させると、論理ベクトルの (1,1) と確率ベクトルの (1/2, 1/2) が対応することになる。この考えは無差別原理と同等のものである [18]。同様に考えると 2 変数では論理ベクトル (1,1,1,1) と確率ベクトル (1/4,1/4,1/4,1/4) が対応することになる。そして一般に n 変数のトートロジの場合には以下の対応が存在する事になる。

$$
\begin{aligned}
&2\ \ 変数\ \ (1,1,1,1) \Leftrightarrow (1/4,1/4,1/4,1/4)\\
&n\ \ 変数\ \ (1,...,1) \Leftrightarrow (1/(2^n),...,1/(2^n))
\end{aligned}
$$

同様にして以下の対応が得られる。

$$
\begin{aligned}
&x\ \ 1変数\ (1,0) \Leftrightarrow (1,0)\\
&x\ \ 2変数\ (1,1,0,0) \Leftrightarrow (0.5,0.5,0,0)
\end{aligned}
$$

ここで、m 個の要素が 1 であるような論理ベクトル $\mathbf{f} = (f_i)(i = 1 \sim n)$ とそれに対応する確率ベクトル $\mathbf{p} = (p_i)(i = 1 \sim n)$ 考える。この場合上記の議論に従えば以下の対応が成立することになる。

f_i が 1 ならば p_i は $1/m$

f_i が 0 ならば p_i は 0

具体例は以下の通りである。但し左が論理ベクトルで右が確率ベクトルである。

$$
\begin{aligned}
&(1,1,1,0) \ \Leftrightarrow\ (1/3,1/3,1/3,0)\\
&(0,0,1,1) \ \Leftrightarrow\ (0,0,1/2,1/2)\\
&(1,0,0,0) \ \Leftrightarrow\ (1,0,0,0)
\end{aligned}
$$

これは f は p と方向が同じだがノルムが違うベクトルであることを意味する。以降この関係を非古典論理にも拡張して使用する。

3.3.2 変換式

前章の情報量の議論と、前項の議論をまとめると以下の二つになる。

1. f と p のベクトルの情報量は同じである。

2. f と p のベクトルの向きは同じである。

上記の 1 より、$H = I$ となり、これより、

$$N_r = 2^{H_e/2}$$

となる。なぜなら

$$
\begin{aligned}
H &= I\\
\rightarrow H &= n - H_e\\
\rightarrow -log_2(2^{-n/2}N_r)^2 &= n - H_e\\
\rightarrow 2^{-n/2}N_r &= 2^{H_e/2-n/2}\\
\rightarrow N_r &= 2^{H_e/2}
\end{aligned}
$$

だからである。又、2 よりベクトルの向きが同じなため、変換式は以下のようになる。

$$\mathbf{f} = (2^{H_e/2}/|\mathbf{p}|)\mathbf{p} \tag{3.1}$$

ここで、

$$H_e = -\sum_1^{2^n}(p_i log_2 p_i)$$

$$|\mathbf{p}| = (\sum_1^{2^n} p_i^2)^{1/2}$$

$$\mathbf{p} = (p_1, ..., p_{2^n})$$

である。この変換式を用いて確率ベクトルから論理ベクトルに変換できる。即ち頻度等の確率データから論理命題を得ることが出来るわけである。

3.4 確率分布から得られる命題について

上記の変換式を用いることによって確率分布（ベクトル）から論理命題（ベクトル）を得ることができる。

表 3.1: 観測例

事象	雨	晴れ／曇り	日数	論理関数
E1	雨	曇り	23	xy
E2	雨	晴れ	2	$x\overline{y}$
E3	雨でない	曇り	30	$\overline{x}y$
E4	雨でない	晴れ	45	$\overline{xy}$

3.4.1 例

表 3.1は天気に関する観測例である。例えば $E1$ は雨で曇っていた日が 23 日あったと言うことを意味する。この例における確率ベクトル p は $(0.23, 0.02, 0.3, 0.45)$ である。但し、第 1 成分は xy に対応する事象 $E1$ の確率、第 2 成分は $x\overline{y}$ に対応する事象 $E2$ の確率、第 3 成分は $\overline{x}y$ に対応する事象 $E3$ の確率、第 4 成分は $\overline{xy}$ に対応する事象 $E4$ の確率である。($\overline{x}, \overline{y}$ は $1-x, 1-y$ の略記。) これを式 3.1で論理ベクトル f に変換すると $(0.69, 0.09, 0.99, 1.35)$ となる。この論理ベクトルの情報量は 0.32 であまり情報がないことがわかる。また述語確定度は基準ベクトルを $(1,0,1,1) = \overline{x} \vee y (= x \to y)$ とすると 0.97 となり、この論理ベクトルが「雨ならば曇っている。」という命題で良く近似されることを示している。従って古典論理に近似する前の論理ベクトル $(0.69, 0.09, 0.99, 1.35)$ は近似すると「雨ならば曇っている。」に近似されるような命題を意味する。

3.4.2 各種確率概念との関係

確率の概念は種々あるが [72]、本論文の確率ベクトルは頻度確率に当たり、論理ベクトルの述語確定度は主観確率に相当する。従って式 3.1は頻度確率から主観確率を求める式とみなすことが出来、更に無差別原理がその変換式の基礎であるといえる。また述語確定度は信念の度合 [1] に相当し、式 3.1は頻度確率から信念の度合を求める変換式であるともいえる [2]。

3.4.3 論理ベクトルの真理値について

前項での簡単な例では、論理ベクトル f が $(0.69, 0.09, 0.99, 1.35)$ となった。ここで 4 番目の真理値が 1 を越えていることが問題であるが、以下ではこれについて説明する。いま実際には 2 変数で記述されるべき事実があったとしよう。すなわちその独立変数を x, y とすると $f(x,y)$ と記述されるべきであると言うことである。そしてたとえば確率ベクトルが $(1/3, 1/3, 1/3, 0)$ であったとしよう。これを論理ベクトルに変換すると $(1,1,1,0)$ となり、$X \vee Y$ と言う命題が得られる。しかしこれを x のみで表現すると確率ベクトルは $(2/3, 1/3)$ となり、これを論理ベクトルに変換すると $(1.23, 0.61)$ となり、その真理値が 1 を越えることになる。

この例からも分かるように対象を表現するのに適切な変数の組を選べば古典論理で表現される場合でもそれよりも小さい規模の変数の組合せを選べば古典論理で表現されないと同時に真理値が 1 を越えることになるのである。従って真理値が 1 を越えると言う現象は適切な変数を選んでいないと言うことに起因すると言える。現実の問題を考えるならば、人間は対象を支配している独立変数をすべて枚挙することは通常は不可能である。従ってその対象を説明するために全独立変数の内から恣意的にいくつか選び、対象を記述せざるを得ない。それゆえ上記のように、通常は真理値が 1 を越えることになるのである。即ち存在の構造と認識の構造が一致した時に完全な記述が可能となり、それ以外の時は不完全な記述になると言うことである。

3.4.4 確率データからの帰納推論アルゴリズム

アルゴリズムは以下の通りである。

1. 確率データを確率ベクトルに変換する。

   確率データが (a_i) の時は確率ベクトルは (p_i) $(p_i = a_i / \sum a_i)$ となる。

2. 確率ベクトルを論理ベクトルに変換する。

   式 3.1 f= $(2^{H_s/2}/|\mathrm{p}|)\mathrm{p}$ を使う。

3. 論理ベクトルを古典論理ベクトルで近似する。

この変換式で得られた論理ベクトル f を古典論理ベクトル g で近似する

$f_i \geq 0.5$ の時は $g_i = 1$, その他は $g_i = 0$

4. 古典論理ベクトルを古典論理命題に変換する。

5. この論理命題を簡単化する。

この段階ではクワインーマクラスキー法等を使用することになる。尚,今までは2値の場合で議論してきたが、1から3のステップは2値、多値にかかわらず同様の処理であり、4,5では多値用のクワインーマクラスキー法を開発したので、多値でもこのアルゴリズムは使用できる。

論理ベクトルを古典論理ベクトルで近似することによって小さい頻度は捨てられる。これは真理値に対して0.5のしきい値を用いているのと同じである。真理値での0.5に対応する値を確率側で求めるのは非常に困難である。この困難さは確率ベクトルから論理ベクトルへの変換式 f= $(2^{H_e/2}/|\mathrm{p}|)\mathrm{p}$ をみれば容易にわかるであろう。又この近似は確率側で言えば適当なしきい値でそれより小さい値を切り捨てて例外扱いにしていることになる。

なお多値の場合でも1または0を要素とする論理ベクトルを古典論理ベクトルと呼ぶことにする。

3.4.5 例

このアルゴリズムを表3.1の例で説明する。但し確率データを (250,20,400,330) とすると、確率ベクトルは最初の手順で (0.25, 0.02, 0.4, 0.33) となる。論理ベクトルは式3.1より (0.774, 0.062, 1.239, 1.022) となり、これに最も近い古典論理ベクトルは (1, 0, 1, 1) となる。これを命題に変換すると $XY \vee \overline{X}Y \vee \overline{XY}$ となり、簡単化すると $\overline{X} \vee Y (= X \rightarrow Y)$ となる。

得られた結果は「雨ならば曇っている」と言う命題になる。天気雨の日数が他に比べて少ない為、これを例外として処理し、残った部分を命題にしている。

その確率は 0.98(=250+400+330/250+20+400+330) となり、妥当な結果である。人間はこの様な確率データから蓋然的な命題（規則）を帰納的に導いているが、それと同様のことをしていると言える。データが変われば当然結果も変わり、例えば、天気雨の日数が増えて、他と同じくらいになれば、「雨ならば曇っている。」と言う命題が得られなくなるがこれは晴れていても雨が降るのであるから当然と言える。得られた命題より弱い命題も成立するため一個の確率データから一般には複数の命題が得られることになる。例えば Y であると言う結果が得られたとする。即ち「曇っている」と言う結果が得られたとする。この場合に言えることはこれだけではなく、図1.2よりわかるように Y より弱い命題 $X \vee Y, \overline{X} \vee Y$ も言えるのであり、これらの命題をどう使うかは問題毎に違うと思われる。尚、A ならば B の様な規則に変形する方法も何通りもあり、どの様に変形するかは利用方法に依存する。なおこのアルゴリズムはノイズに強いアルゴリズムであることが確認されている [37][77][78]。

3.5 準最尤法

本節では3.2節の近似法が無差別原理を仮定すれば準最尤法であることを示す。

いま論理ベクトル f と古典論理ベクトル g が十分近いと仮定する。即ち

$$f_i \simeq g_i, (i = 1, ..., n)$$

とする。この時3.2節の近似は

$$\mathrm{f} - \mathrm{g} \rightarrow min$$

であり、今 $f_i \simeq g_i, (i = 1, ..., n)$ であるから上式は

$$||\mathrm{f}| - |\mathrm{g}|| \rightarrow min$$

となる。$|\mathrm{f}|, |\mathrm{g}|$ を前節までの議論に基づいて $2^{H_e/2}, 2^{H'_e/2}$, とおくと上式は

$$|2^{H_e/2} - 2^{H'_e/2}| \to min$$

となり、

$$|H_e - H'_e| \to min$$

となる。$H'_e > H_e$ の場合を考えると

$$H'_e - H_e \to min$$

となり、$H_e = -\sum_1^{2^n}(p_i log_2 p_i), H'_e = -\sum_1^{2^n}(q_i log_2 q_i)$ とおけば、

$$\sum_1^{2^n}(p_i log_2 p_i) - \sum_1^{2^n}(q_i log_2 q_i) \to min$$

$f_i \simeq g_i, (i = 1, ..., n)$ を仮定しているから $p_i \simeq q_i, i = 1, ..., n$ であり、対数関数が定義域 [0,1] で急降下関数であることを考えると式は

$$\sum_1^{2^n}(p_i log_2 p_i) - \sum_1^{2^n}(p_i log_2 q_i) \to min$$

と変形できる。これは K-L 情報量であり、この近似法は最尤法となる [54]。従って $f_i \simeq g_i, (i = 1, ..., n)$ を仮定すれば最尤法になるので、3.2 節の近似法は準最尤法と言える。

3.6 効率的なアルゴリズム

前節までの議論では、特に論理式の制約はなく、回帰式も多重線形関数であった。しかしその近似方法の計算量は指数オーダーであり、変数の数が多い場合は実際的とは言えない。本節では回帰式が線形の場合には計算量が多項式オーダーの効率的なアルゴリズムが存在することを述べる [65] [79]。表現形としては DNF(Disjunctive Normal Form) 式を考える。なお定義域が [0,1] の場合も DNF 式と言う表現を使う。なおこのアルゴリズムは閾値論理で提示された閾値論理関数から論理式を求めるアルゴリズムの拡張になっている [43]。

3.6.1 回帰式と直交表現の係数の関係式

ここで重回帰の回帰式である線形関数を

$$p_1 x_1 + ... + p_n x_n$$

とし、それの直交関数による展開式を

$$a_1 x_1 \cdot\cdot x_n + ... + a_{2^n}(1 - x_1)\cdot\cdot(1 - x_n)$$

とすると、

$$p_1 x_1 + ... + p_n x_n = a_1 x_1 \cdot\cdot x_n + ... + a_{2^n}(1 - x_1)\cdot\cdot(1 - x_n)$$

となる。a_i は次の通りである。

$a_1 = p_1 + ... + p_n,$

$a_2 = p_1 + ... + p_{n-1},$

...

$a_{2^{n-1}} = p_1,$

$a_{2^{n-1}+1} = p_2 + p_3 + ... + p_{n-1} + p_n,$

...

$a_{2^n-1} = p_n,$

$a_{2^n} = 0$

各係数 a_i はそれに対応する直交関数の値のみ 1 にし、他の直交関数の値を 0 にするような変数の値の組合わせを代入することによって簡単に求められる。例えば a_1 ならばこれに対応する直交関数 $x_1 \cdot\cdot x_n$ の値を 1 にする変数の値の組合わせは $x_1 = x_2 = \cdot\cdot = x_n = 1$ である。これを代入すると右辺は a_1 となり、左辺は $p_1 + ... + p_n$ となり、$a_1 = p_1 + ... + p_n$ が得られる。同様に $x_1 = x_2 = \cdot\cdot = x_{n-1} = 1, x_n = 0$ を代入すると $a_2 = p_1 + ... + p_{n-1}$ となる。他の a_i も同様にして計算できる。

3.6.2 近似後のブール関数に x_i 等が存在する条件

ここで x_1 に注目する。ブール関数に近似された後で x_1 が存在することは

$x_1x_2 \cdot\cdot x_n$,

$x_1x_2 \cdot\cdot (1-x_n)$,

...

$x_1(1-x_2)\cdot\cdot(1-x_n)$.

の全ての直交関数が存在することである。即ち各直交関数の係数, $a_1, a_2, ..., a_{2^{n-1}}$ が全て 0.5 以上のことである。即ち $min\{a_i\} \geq 0.5(i = 1 \leq i \leq 2^{n-1})$ （以下 $mina_i$ と略記）である。

ところで $a_i(1 \leq i \leq 2^{n-1})$ は

$a_1 = p_1 + ... + p_n$,

$a_2 = p_1 + ... + p_{n-1}$,

...

$a_{2^{n-1}} = p_1$,

であり、$a_i(1 \leq i \leq 2^{n-1})$ は全て p_1 を含んでいる。従ってもし p_i が全て非負ならば $a_{2^{n-1}} = p_1$ が最小になる。なぜならばその他の a_i は他の p_i を含んでいるから $a_{2^{n-1}} = p_1$ より大きくなるからである。一般には p_i は全て非負ではない。その場合には負である p_i を全て含んでいる a_i が最小になる。即ち $mina_i = p_1 + \sum_{1\leq j\leq n, j\neq 1, p_j \leq 0} p_j$ となる。そしてこのような a_i は必ず $a_i(1 \leq i \leq 2^{n-1})$ の中に存在する。なぜならば $a_i(1 \leq i \leq 2^{n-1})$ は $p_1 + (p_j(2 \leq j \leq n)$ の任意の組合せ) であるからである。以上の議論より

$$mina_i = p_1 + \sum_{1\leq j\leq n, j\neq 1, p_j\leq 0} p_j$$

となる。従って、近似後のブール関数に x_1 が存在する条件は

$$p_1 + \sum_{1\leq j\leq n, j\neq 1, p_j\leq 0} p_j \geq 0.5$$

となる。回帰式 $p_1x_1 + ... + p_nx_n$ は x_i に関して対称であるから以上の議論は他の x_i についても成立する。即ち

$$p_i + \sum_{1\leq j\leq n, j\neq i, p_j\leq 0} p_j \geq 0.5$$

ならば x_i が近似後のブール関数に存在する。同様に考えると $\overline{x}_i$ が近似後のブール関数に存在する条件は

$$\sum_{1\leq j\leq n, j\neq i, p_j\leq 0} p_j \geq 0.5$$

となる [56]。今までの議論は最も低次の項（$x_i, \overline{x}_j$）についての議論であったが同様の議論が高次の項（x_ix_j 等）の存在についても可能であり、その結果一般に $x_{i_1}\cdot\cdot x_{i_k}\overline{x}_{i_{k+1}}\cdot\cdot\overline{x}_{i_l}$ が近似後のブール関数に存在する条件は

$$\sum_{i_1}^{i_k} p_{i_j} + \sum_{1\leq j\leq n, j\neq i_1,...,i_l, p_j\leq 0} p_j \geq 0.5$$

になる。

3.6.3 DNF 式の生成

上記の判定条件を用いて低次の項から生成して行き、その項（ブール関数）を論理和で接続して最終的なブール関数を得る。低次の項で存在が確認された変数に関してはその項より高次の項での存在の判定の必要がない。簡単な例で言えば x が存在することが分かれば xy, xz 等は存在する（$x = x \vee xy \vee xz$）ので xy, xz の存在の判定は必要ないのである。従ってこのアルゴリズムは簡単化を同時に行なっている。このブール関数同定アルゴリズムの例を以下に挙げる。$y = 0.65x_1 + 0.23x_2 + 0.15x_3 + 0.20x_4 + 0.02x_5$ を重回帰分析の回帰式とする。まず x_i に関しては x_1 の存在条件 p_1 のみが 0.5 以上なので x_1 が存在することになる。つぎに x_1 を除外した x_ix_j に関してはどの p_i と p_j の和も 0.5 を越えないので x_ix_j と言う項は存在しないことになる。そして次に x_1 を除外した $x_ix_jx_k$ に関しては $x_2x_3x_4$ の存在条件 $p_1 + p_2 + p_3$ が 0.5 以上なので $x_2x_3x_4$ が存在することになる。x_1, x_2, x_3, x_4 を除外すると x_5 が残るが、これ一つでは更に高次の項は作れないのでここでアルゴリズムは終了する。従って存在するのは x_1 と $x_2x_3x_4$ となり、その論理和 $x_1 \vee x_2x_3x_4$ が求めるブール関数となる。

3.6.4 計算量と誤差評価

まず重回帰分析の計算量は回帰関数の変数（＝項）の多項式である。これはブール関数の変数の数 n の多項式である。従って重回帰分析の計算量はブール関数の変数の数 n の多項式であると言える。重回帰式からブール関数を求めるアルゴリズムの計算量は簡単な計算より多項式であることが分かる。但し高次の項まで同定しようとすると計算量が増加して行き最高次まで同定するとその計算量は指数関数になる。従って計算量を多項式にするには項の生成をある次数で止めねばならない。ある次数までの論理式で近似する場合の誤差評価に関して [22] の結果が有効である。Linial は一様分布の仮定のもとで以下の式を提示した。

$$\sum_{|S|>k} \hat{f}(S)^2 \leq 2M2^{-k^{1/2}/20},$$

ただし f はブール関数である。S は項（もしくは項の集合）、$|S|$ は項（もしくは項の集合）の次数であり、k は整数、$\hat{f}(S)$ は f の S 次フーリエ変換であり、M は回路のサイズである。

上式は高次の項は非常に少ないパワーしかなく、低次の項に情報が集中していることを意味している。即ちある次数までの項の近似で十分良い近似になっていることになる。即ち若干の誤差を付加するだけで計算量を多項式に落せることを意味する。更に詳細な誤差解析は今後の課題である。

3.7 おわりに

本章では多重線形関数をブール関数で近似することを考えた。この近似は情報量を用いると無差別原理の仮定のもとで準最尤法であることが証明できた。しかしこの近似法の計算量は指数オーダーであるため、変数の数が多くなると実際的でない。そこで回帰関数が線形である場合の効率的な近似アルゴリズムを提示した。近似誤差に関する議論は [22] の結果を用いて行なった。2節では近似の一般的議論を行なった。3節ではこの近似法が準最尤法であることを証

明するために必要な準備を行なった。4節では3節で得られた変換式を用いた帰納推論について述べた。5節では準最尤法の証明を行なった。6節では効率的なアルゴリズムについて述べた。

第 4 章

応用

4.1 はじめに

前章の近似手法を具体的な 3 種類の問題に適用する。最初の 2 種類の問題は以下の通りである。

1. 帰納学習 [40][79]

   多重線形関数空間は $\{0,1\}$ データと $[0,1]$ データを扱えるが、更にダミー変数を導入することによって離散データを $\{0,1\}$ データに還元することによって帰納学習で通常扱うデータを扱えるようにする。このようにして重回帰分析を行ない多重線形関数を得る。この関数をブール関数で近似することによって論理命題を得る。回帰関数が線形の場合に前章の効率的なアルゴリズムを使用し、C4.5 と比較実験を行なう。誤差解析は [15] と [22] の結果を用いて行なう。

2. 数値データ [38][67]

   数値データを重回帰分析して多重線形関数を求め、その関数をブール関数で近似することによって論理命題を求める。

上記の問題のアルゴリズムの一般形は以下の通りである。

1. データを 1 次多重線形関数 (= 線形関数) で重回帰分析して回帰関数を得る。残差誤差が一定値以下になるまで高次の項を追加して重回帰分析を行なう。

2. 得られた回帰関数をブール関数で近似する。

回帰関数が線形関数でも帰納学習の代表的なアルゴリズム C4.5 より良い結果が出るのを実問題で確認しているので本論文では回帰関数が線形関数の場合に議論を限定している。

またもう一つの問題は ニューラルネットワークの構造分析である。[68][71] ニューラルネットワークの各素子は線形関数と出力関数から構成されている。この出力関数を線形関数で近似し、得られた線形関数を古典論理関数で近似することによってニューラルネットワークの構造分析を行なう。本論文では排他的論理和（連続値も含めて）の例を示す。

4.2 帰納学習

本節では帰納学習へ前章のアルゴリズムを適用する。

4.2.1 人工知能における帰納学習関連の研究の概要

近年、人工知能の分野で帰納推論、帰納学習と呼ばれる研究が盛んに行なわれている。人工知能では「帰納推論」は主に形式言語、オートマトン、帰納関数等を正負の事例から求める推論のことを意味するが [47]、これは本論文とはあまり関係ないのでここでは省略する。これに対し、「帰納学習」は各種データから論理式、決定木等を求める推論のことを意味する。即ち帰納学習とは、記号表現された概念集合（例えば命題論理の命題のあるクラス）の中から与えられた事例（例えば表 3.1）を基にしてある概念を得ることである [49]。

帰納学習は教師あり学習と教師なし学習に分類される。教師あり学習とは与えられた事例がその概念を満足する正例かその概念を満足しない負例に分類されている場合である。多変量解析で教師あり学習に相当するのは被説明変数のある重回帰分析、判別分析や、外的基準のある数量化 1 類、2 類である。これに対して教師なし学習とは正例、負例に分類されてなく、存在する事例から適当な概念を学習する（見つける）場合である。多変量解析で教師なし学習に相当するのは被説明変数のない主成分分析や外的基準のない数量化 3 類である。

表 4.1: 例

事例	A	B	C	クラス
E1	a_1	1.1	?	1
E2	a_1	?	c_2	2
E3	a_2	2.3	c_1	1
E4	a_1	1.0	c_1	1
E5	a_3	1.2	c_2	2
E6	a_2	0.0	c_1	2
E7	a_3	2.2	c_2	1

教師なし学習は概念を事例から形成するので概念形成とも呼ばれる。従って本論文が扱う問題は人工知能での帰納学習の教師あり学習に相当する。

4.2.2 学習アルゴリズム

ここでは、命題獲得と予測のアルゴリズムについて述べる。このアルゴリズムの概要を以下に示す。

1. 事例表現 — 欠損値補填、連続属性と離散属性の表現、変数の削減を行なう
2. 重回帰分析 — 重回帰分析を行なう
3. 予測 — テスト事例を回帰式に代入して予測を行なう
4. 命題生成 — 回帰式から論理ベクトルを得、命題を生成する

ここでは例として表 4.1 を用いて説明していく。このデータは、属性として $\{A, B, C\}$ の 3 つがあり、A は $\{a_1, a_2, a_3\}$ の 3 値をとる離散属性、B は連続属性、C は $\{c_1, c_2\}$ をとる 2 値の離散属性である。クラスは $\{1, 2\}$ の 2 値である。また、「?」は属性値が不明である欠損値を表している。

欠損値補填

実問題の中には、さまざまな理由によりある属性の値が不明である事例が存在する。この不明である値を欠損値 (missing value , unknown attribute value) と呼ぶ。欠損値を含む場合には、欠損値を含む属性や事例を使用しないという方法も考えられるが、欠損値の多いデータでは多くの情報を切捨ててしまうことになるので、適当な値を補填して使用することとした。欠損値の補填は次のように行なう。

まず、欠損値を持つ事例のクラスと同じクラスである事例の集合を作り、欠損している属性の値に注目する。このとき、

離散属性 : クラスが同じ事例の集合の中で最頻の値
連続属性 : クラスが同じ事例の集合の中の平均値

によって欠損値を補填することとする。表 4.1 のデータでは、E1 のクラスが 1 なので、同じクラスである {E3, E4, E7} という部分集合を考え、その部分集合の中で属性 C に注目すると c_1 が 2 事例、c_2 が 1 事例なので、c_1 を E1 の属性 C の値とする。E2 の属性 B は連続値をとるので、クラス 2 である部分集合 {E5, E6} の属性 B の値の平均値をとり、0.6 を E2 の属性 B の値とする。

欠損値の補填の方法としては、このほかにも全事例中最頻の値を用いる、「欠損値」という属性値を考える、類似の事例の値を用いる、欠損値をクラスとして推定する、等が考えられる [32]。どのような方法が最適であるかは今後調べる必要がある。

連続属性の表現

連続値をとる属性に対しては、属性値の $[0, 1]$ への正規化を行なう。これは、その属性がクラスの決定に与える影響の相対的な度合いを取り出すためである。この相対的な影響度は、後に命題を生成する際に重要となる。正規化は次のように行なう。

今、ある連続属性の最小値を min 、最大値を max とすると、値 x は

$$\frac{x - \min}{\text{max-min}}$$

によって、変換される。表 4.1 の属性 B については、

(1.1, 0.6, 2.3, 1.0, 1.2, 0.0, 2.2)

↓

(0.478, 0.261, 1.000, 0.435, 0.522, 0.000, 0.957)

となる。

離散属性の表現

離散値をとる属性に対しては、ダミー変数によって $\{0, 1\}$ の 2 値で表現する。ダミー変数とは質的データを数値的に表現するための変数で、事例に対して次のように値を定める。

$$x_{ij} = \begin{cases} 1 & \text{属性 } i \text{ について値 } j \text{ を持つとき} \\ 0 & \text{その他のとき} \end{cases}$$

例えば、属性 A の属性値が $\{a_1, a_2, a_3\}$ のとき、属性値「a_2」は「0, 1, 0」と表される。このとき、各属性の属性値の数をそれぞれ n_i とすると、ダミー変数の数は $\sum n_i$ になる。また、クラスに対してもダミー変数による表現を用いて、クラス毎に事例集合を生成する。表 4.1 の例ではクラスは $\{1, 2\}$ なので事例集合が 2 つ生成される。表 4.1 に対して、以上の欠損値補填、離散属性、連続属性の表現を施した結果を 表 4.2 に示す。

変数削減

ダミー変数によって表現されたデータに対し、変数の削減を行なう。これはダミー変数の導入による線形従属性（例えば $a_1 + a_2 + a_3 = 1$ である。）を消去するためである。ここでは、各離散属性に対応するダミー変数の組（属性値に相当する）から一つずつ変数を削除する。削除される変数はクラスとの負の相関が最も高いものである。例として表 4.2-a における属性 A に注目してみる。属性 A の 3 つのダミー変数の値とクラスの値を比較して一致する数を求める。この場合は、a_1 に対しては 4 事例、a_2 に対しては 3 事例、a_3 に対しては 2 事例である。従って、最も負の相関が高い a_3 を削除することとする。同様に属性 C に対しては、c_2 が削除される。また、表 4.2-b に対しても同様であるが、クラ

表 4.2: 欠損値、離散値、連続値、処理済み

表 4.2-a: クラス 1 に対して

	A			B	C		
事例	a_1	a_2	a_3		c_1	c_2	クラス 1
E1	1	0	0	0.478	1	0	1
E2	1	0	0	0.261	0	1	0
E3	0	1	0	1.000	1	0	1
E4	1	0	0	0.435	1	0	1
E5	0	0	1	0.522	0	1	0
E6	0	1	0	0.000	1	0	0
E7	0	0	1	0.957	0	1	1

表 4.2-b: クラス 2 に対して

	A			B	C		
事例	a_1	a_2	a_3		c_1	c_2	クラス 2
E1	1	0	0	0.478	1	0	0
E2	1	0	0	0.261	0	1	1
E3	0	1	0	1.000	1	0	0
E4	1	0	0	0.435	1	0	0
E5	0	0	1	0.522	0	1	1
E6	0	1	0	0.000	1	0	1
E7	0	0	1	0.957	0	1	0

スの値が異なるので表 4.2-a と表 4.2-bとでは異なる変数が削除されることに注意されたい。以上の事例表現の全ての処理を施した結果を、表 4.3 にまとめておく。

表 4.3: 事例表現の全ての処理を終了

表 4.3-a: クラス 1 に対して

	A		B	C	
事例	a_1	a_2		c_1	クラス 1
E1	1	0	0.478	1	1
E2	1	0	0.261	0	0
E3	0	1	1.000	1	1
E4	1	0	0.435	1	1
E5	0	0	0.522	0	0
E6	0	1	0.000	1	0
E7	0	0	0.957	0	1

表 4.3-b: クラス 2 に対して

	A		B	C	
事例	a_1	a_2		c_1	クラス 1
E1	0	0	0.478	0	0
E2	0	0	0.261	1	1
E3	1	0	1.000	0	0
E4	0	0	0.435	0	0
E5	0	1	0.522	1	1
E6	1	0	0.000	0	1
E7	0	1	0.957	1	0

重回帰分析

欠損値補填、連続値の正規化、ダミー変数表現、変数の削減、の処理を終えたデータに対して重回帰分析を行ない、回帰式を求める。このとき、データはクラス毎に生成されているので、重回帰分析もクラス毎に行ない、回帰式を得る。表 4.1 に表されるデータはすでに述べたように 表 4.3 に変換されている。表 4.3-a と 表 4.3-b に対して得られる回帰式はそれぞれ次のようになる。

$$0.0765x_1 + (-0.4759x_2) + 1.2048x_3 + 0.7645x_4 - 0.3909 \tag{4.1}$$

$$0.5524x_1 + 0.0765x_2 + (-1.2048x_3) + (-0.7645x_4) + 0.5500 \tag{4.2}$$

論理命題獲得

前章の効率的なアルゴリズムを式 4.1と 式 4.2 に適用すると、それぞれ

$$\bar{x_2}x_3 \vee x_3x_4 \tag{4.3}$$

$$\bar{x_3}\bar{x_4} \tag{4.4}$$

が得られる。但し、ここで x_i はダミー変数表現、変数の削減を行なった後の変数を表している。すなわち表 4.3 に表されている変数を順に x_1, x_2, $\cdots$ とした。これは、元のデータに合わせると、

$$\bar{a}_2B \vee Bc_1 \rightarrow \text{クラス } 1$$

$$\bar{B}\bar{c}_2 \rightarrow \text{クラス } 2$$

となる。ここで、連続属性である B は、否定の場合は値が小さいこと、そうでない場合は値が大きいことを表していると考えられる。

4.2.3 予測

まず予測をする事例 (テスト事例と呼ぶ) を訓練事例の表現と合わせる必要がある。すなわち、欠損値、連続値正規化、ダミー変数、変数削減等の処理である。

訓練事例に対する処理と同様に、連続属性は [0,1] に正規化し、離散属性はダミー変数を用いて表す。但し、欠損値に関しては補填はせずに、該当する値を 0 とおく。すなわち、欠損値を持つ属性が連続属性である場合にはそれを 0 とし、離散属性である場合にはダミー変数に展開した後その属性に関するダミー変数 (3 値なら 3 変数) の値を全て 0 とする。これにより、値が欠損している属性に関する情報は使わないことになる。変数の削減に関しては、訓練事例で削除したのと同じ変数を削除する。

回帰式による予測

表現し直したテスト事例を回帰式に代入し、値を求める。回帰式はクラスごとに得られているので、クラスごとに代入値が求まる。得られた値のうち最大値に対応するクラスが、このテスト事例に対して予測されるクラスである。なぜなら、重回帰分析を行なう前に、クラスを $\{0,\ 1\}$ としてクラスごとに事例の集合を生成しているため、新事例のクラスは最大値のクラスである可能性が最も高いと考えられる。

表 4.1 の例で、例えば $(a_3,\ 2.2,\ c_2)$ というテスト事例 (E7 と同じ事例である) に対しては、表現し直した値 $(0,\ 0,\ 0.957,\ 0)$ を式 4.1に、$(0,\ 1,\ 0.957,\ 1)$ を式 4.2に代入すると

$$0.0765 \cdot 0 - 0.4759 \cdot 0 + 1.2048 \cdot 0.957 + 0.7645 \cdot 0 - 0.3909 = 0.7621$$

$$0.5524 \cdot 0 + 0.0765 \cdot 1 - 1.2048 \cdot 0.957 - 0.7645 \cdot 1 + 0.5500 = -1.2910$$

となり、クラス 1 に対応する式 4.1のほうが大きいのでこの事例のクラスは「1」であると予測される。

命題による予測

回帰式による予測と同様に、クラス毎の命題に事例の値を代入したときの最大値に対応するクラスが予測されるクラスである。前に述べた例では、回帰式の直交関数展開が、

$$z = 0.84xy + 0.26x(1-y) + 0.61(1-x)y + 0.03(1-x)(1-y)$$

であり、各項の係数を $\{0,1\}$ に近似すると

$$z = 1\,xy + 0\,x(1-y) + 1\,(1-x)y + 0\,(1-x)(1-y)$$

であった。これは計算すると $z = xy + (1-x)y = y$ である。従って、「命題に事例の値を代入する」とはこの場合変数 y の値そのものになるわけである。

ところで、実際には直交関数展開は行なわず、前章の効率的なアルゴリズムを使用している。この場合には第 2 章で述べた通り、ブール代数を初等代数で書き換え、事例の値を代入すれば良い。例えば、$XY \vee Z$ という命題は書き換えにより、$xy + z - xyz$ になる。これにより、各クラスに対して値が求まり、そのうちの最大値に対応するクラスが予測されるクラスとなる。

表 4.1 の例で考える。$(a_1,\ 1.1,\ c_1)$ というテスト事例に対して、表現し直した値 $(0,\ 0,\ 0.957,\ 0)$ を式 4.3に代入すると

$$\begin{aligned}
\text{式4.3} &= \tau((1-x_2)x_3 + x_3x_4 - (1-x_2)x_3x_3x_4) \\
&= (1-x_2)x_3 + x_3x_4 - (1-x_2)x_3x_4) \\
&= (1-0) \cdot 0.957 + 0.957 \cdot 0 \\
&= 0.957
\end{aligned}$$

となる。同様にしてクラス 2 に対して $(0,\ 1,\ 0.957,\ 1)$ とを式 4.4 に代入すると

$$\begin{aligned}
\text{式4.4} &= \tau((1-x_3)(1-x_4)) \\
&= (1-x_3)(1-x_4) \\
&= (1-0.957)(1-1) \\
&= 0.000
\end{aligned}$$

クラス 1 の方が値が大きいのでクラス 1 と予測される。

4.2.4 実験

ここでは、このアルゴリズムで獲得された命題と予測の正解率の実験について述べる。実験の方法は次の通りである。

1. データより訓練事例をランダムに選択

2. 訓練事例により学習 (回帰式獲得)

3. 訓練事例とは別にテスト事例を選択

また、比較のために機械学習アルゴリズムの一つである C4.5 [33] によっても同一のデータの学習を行なった。C4.5 は決定木学習アルゴリズムとして有名な ID3 [31] の拡張である。決定木の各節点には属性が対応しており、属性値ごとに枝に分かれている。また、先端の葉にはクラスが対応している。まず、訓練事例を用いて決定木を生成する。このとき、各節点では最も良くクラスを判別するような属性が選択される。その基準には情報量を用いている。テスト事例の予測の際には、決定木の根から順にその事例を表す属性値の枝をたどり、最後にたどりついた葉のクラスが予測されるクラスである。また、決定木より複数個のルールを生成することができ、そのルールを用いた予測も可能である。

実験に用いたデータは UCI (The University of California at Irvine) の Machine Learning database から入手したものである。以下に、各データの説明と得られたルールを示す。予測の正解率については最後に表 4.4 にまとめた。

mushroom

mushroom が食べられる (edible) か食べられないか (poisonous) をクラスとしている。属性は mushroom の形状 (例えば cap-shape が bell あるいは flat 、など) に関して 22 個あり、全て離散値である。事例は全部で 8124 事例、欠損値も存在する。得られたルールの例を次に示す。

(odor:creosote)(gill-color:green) or (odor:creosote)(stalk-root:rooted でない) or
(odor:creosote)(stalk-surface-below-ring:scaly) or (odor:fishy)(gill-color:green) or
(odor:fishy)(stalk-root:rooted でない) or (odor:fishy)(stalk-surface-below-ring:scaly) or
(odor:foul)(gill-color:green) or (odor:foul)(stalk-root:rooted でない) or
(odor:foul)(stalk-surface-below-ring:scaly) or (odor:pungent)(gill-color:green)

or
(odor:pungent)(stalk-root:rooted でない) or (odor:pungent)(stalk-surface-below-ring:scaly) or
(odor:spicy)(gill-color:green) or (odor:spicy)(stalk-root:rooted でない) or
(odor:spicy)(stalk-surface-below-ring:scaly) or (gill-color:green)(stalk-root:rooted でない) or
(gill-color:green)(stalk-surface-below-ring:scaly) or (stalk-root:rooted でない)(stalk-surface-below-ring:scaly)
→ poisonous

(odor:almond)(gill-color:green でない)(stalk-root:rooted) or
(odor:almond)(gill-color:green でない)(stalk-surface-below-ring:scaly) or
(odor:almond)(stalk-root:rooted)(stalk-surface-below-ring:scaly) or
(odor:anise)(gill-color:green でない)(stalk-root:rooted) or
(odor:anise)(gill-color:green でない)(stalk-surface-below-ring:scaly) or
(odor:anise)(stalk-root:rooted)(stalk-surface-below-ring:scaly) or
(odor:none)(gill-color:green でない)(stalk-root:rooted) or
(odor:none)(gill-color:green でない)(stalk-surface-below-ring:scaly) or
(odor:none)(stalk-root:rooted)(stalk-surface-below-ring:scaly) or
(gill-color:green でない)(stalk-root:rooted)(stalk-surface-below-ring:scaly)
→ edible

[C4.5]

(odor:foul) or (odor:spicy) or (odor:fishy) or (odor:pungent) or (odor:creosote)
→ poisonous

(odor:none) or (odor:almond) or (odor:anise)
→ edible

本論文のアルゴリズムで得られたルールは C4.5 に比べ複雑なルールに見えるが、C4.5 よりもクラスの判定のための条件が多いだけで、基本的な内容は同じである。予測の正解率は C4.5 よりも良い (表 4.4)。

voting-records

1984 年の米国議会議員の記録である。属性と属性値はいくつかの政策に関する議員の賛否であり、クラスは議員の所属する政党で democrat と republican のどちらかである。事例数は 435、欠損値は存在する。得られたルールの例は次の通り。

(water-project-cost-sharing:y)(physician-fee-freeze:n)(el-salvador-aid:y)
or
(adoption-of-the-budget-resolution:y)(physician-fee-freeze:n)(el-salvador-aid:y) or
(adoption-of-the-budget-resolution:y)(physician-fee-freeze:n)(anti-satellite-test-ban:n) or
(adoption-of-the-budget-resolution:y)(physician-fee-freeze:n)(mx-missile:y)
or
(adoption-of-the-budget-resolution:y)(physician-fee-freeze:n)(superfund-right-to-sue:n) or
(physician-fee-freeze:n)(el-salvador-aid:y)(anti-satellite-test-ban:n) or
(physician-fee-freeze:n)(el-salvador-aid:y)(mx-missile:y) or
(physician-fee-freeze:n)(el-salvador-aid:y)(education-spending:n) or
(physician-fee-freeze:n)(el-salvador-aid:y)(superfund-right-to-sue:n)

or
(physician-fee-freeze:n)(el-salvador-aid:y)(crime:n)
→ democrats

(adoption-of-the-budget-resolution:n)(physician-fee-freeze:y)(el-salvador-aid:n)
→ republican

[C4.5]

(physician-fee-freeze:n) or (adoption-of-the-budget-resolution:y)(synfuels-corporation-cutback:y)
→ democrats

(adoption-of-the-budget-resolution:n)(physician-fee-freeze:y) or
(physician-fee-freeze:y)(synfuels-corporation-cutback:n)
→ republican

mushroom と同様クラス判定のための条件が厳しいが、基本的には同じルールが得られた。正解率は C4.5 よりも良い。

zoo

生物の分類をする問題で、クラスは 「ホニュウ類、鳥類、魚類、ハチュウ類、昆虫、両性類、甲殻類」の 7 値。属性はその生物の特徴として hair の有無、eggs かどうか 等全て離散属性で 16 個ある。101 事例あり、欠損値は存在しない。

(milk : yes) → ホニュウ類
(feathers : yes) → 鳥類
(feathers : no)(toothed : no)(backbone : yes) →ハチュウ類
(hair : yes)(eggs : no)(milk : no)(toothed : yes)(breathes : no) →
魚類

(eggs : yes)(milk : yes)(bacbone : no)(legs : 6) → 昆虫

(eggs : no)(milk : no)(backbone : no)(legs : 6 でない)((toothed : no)or(breathes : no)or(venomous : no)or(fins : no)) → その他

[C4.5]

(milk : yes) → ホニュウ類

(feathers : yes) → 鳥類

(milk : no)(fins : yes) → 魚類

(feathers : no)(milk : no)(airborne : yes) → 両性類

(airborne : no)(backbone : no) → その他

両アルゴリズムとも、あるクラスについてはルールが得られていない。これは事例の数に偏りがあるため、事例の数が少ないのクラスについて良く学習できなかったと考えられる。全部で 101 の事例のうち 41 が「ホニュウ類」に属している。また、最も事例が少ないクラスは「両性類」で 4 事例しかない。この他にも「ハチュウ類」が 5 事例であるなど事例の数に偏りが大きく、そのため事例の少ないクラスについては命題を生成できなかったと思われる。またその影響を受けて命題による予測の正解率は他に比べて低くなっている。

iris

iris(あやめ) の種類分けする問題で、クラスは「Setosa、Versicolour、Virginica」の 3 値。属性はがく片の長さ・幅、花弁の長さ・幅、の連続属性が 4 個である。事例数は 150 で欠損値は存在しない。

(petal length: 小) → Setosa

(petal length: 大)(petal width: 小)

(sepal width: 小)(petal length: 大)

(sepal width: 小)(petal width) → Versicolour

(petal width : 大) → Virginica

[C4.5]

(petal length $\leq$ 1.9) → Setosa

(petal length > 1.9)(petal length $\leq$ 5.3)(petal width $\leq$ 1.7) → Versicolour

(petal width > 1.7)

(petal length > 4.9) → Virginica

iris の属性は全て連続値のデータであるが、このアルゴリズムは連続値で中間の値をとるクラスについては命題の学習が難しい。このことが *iris* の結果に現れている。すなわち、ある属性に関して値が大きいクラスはほとんど 100% の正解率で学習しているが、中間の値をとるクラスの正解率が表 4.4 に示すように全体の正解率を下げている。

表 4.4: 実験結果

データ名	本アルゴリズム		C4.5	
	回帰式	命題	決定木	ルール
mushrooms	99.3%	99.3%	98.7%	98.7%
voting-records	97.0%	97.0%	93.0%	93.5%
zoo	93.1%	84.9%	90.0%	90.0%
iris	84.7%	92.0%	98.7%	97.3%

4.2.5 *iris* について

前述の *iris* の実験結果について若干詳細に述べる。この *iris* のデータでは、がく片の長さ (= a と表すことにする)、がく片の幅 (= b とする)、花弁の長さ (= c とする)、花弁の幅 (= d とする)、という 4 つの属性で事例を表し、クラスは iris(あやめ) の種類で、*setosa*、*versicolour*、*virginica* の 3 個である。たとえば、事例は以下のように表されている。このような事例の集合より、表 4.5 のような命題が得られた。

a	b	c	d	クラス
7.0	3.2	4.7	1.4	versicolour

表 4.5: *iris* の命題

クラス	命題	正しさ
setosa	$\bar{c}$	100%
versicolour	$c\bar{d} \vee \bar{b}c \vee \bar{b}\bar{d}$	76.2%
virginica	d	99.8%

属性は全て連続値をとるので、否定はその属性の値が小さいことを表していると考えられる。すなわち、*setosa* に関しては属性 c (花弁の長さ) が小さいと読むことができる。このことを確認するために、このデータの各事例の分布を図 4.1 に示した。図 4.1 は、X 軸に *iris* の事例をクラスごとにまとめて配置し、Y 軸に各属性の値をとったグラフである。これにより、実際 *setosa* の属性 c が他の 2 つのクラスの属性値に比べて小さいことが分かる。次に、これらの命題がどの程度あてはまっているかを次のような方法で調べた。

(1) あるクラスに属する事例と属さない事例を選択

(2) 事例の属性値を比較

(3) (2) の結果はそのクラスに関する命題を満たしているか

例えば、*setosa* である事例 $e_1 = (5.1, 3.5, 1.4, 0.2)$ と、*setosa* でない事例 $e_2 = (7.0, 3.2, 4.7, 1.4)$ を比較すると、属性 c の値は e_1 のほうが小さいので *setosa* に関する命題 $\bar{c}$ は満たされている。このような実験を行なった結果は表 4.5の通りである。

versicolour の正しさが低いことについては、図よりわかるように他のクラスの属性値と比較して *versicolour* の属性値が中程度であるため、ブール関数で表現することが難しいと考えられる。これについては今後の課題であるが、現在真理値関数に正規分布を仮定して、最尤推定で多次元正規分布を求め、それを 1 次元正規分布の論理関数で近似することを考えている。この際正規分布間の距離が必要であり、それは甘利の情報幾何学 [46] が有効であると考えている。

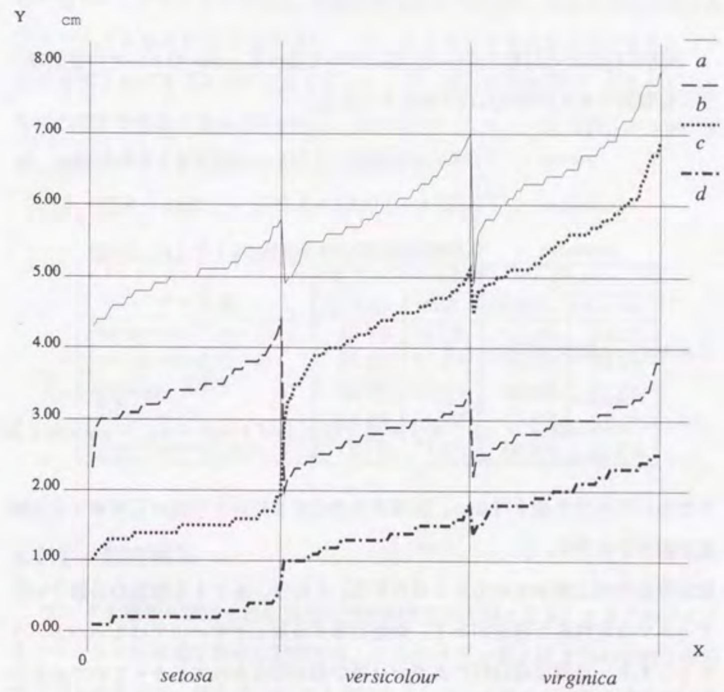

図 4.1: *iris* データの分布

表 4.6: 近似後の関数の予測正解率

クラス	近似しない	近似後
setosa	100%	100%
versicolour	68%	72%
virginica	86%	90%

次ぎに近似後の関数を用いた予測について述べる。*iris* のデータの重回帰分析の結果得られた回帰式は以下の通りである。

$$
\begin{aligned}
setosa &: \ 0.236\,a + 0.582\,b - 1.314\,c - 0.152\,d + 0.661 \\
versicolour &: \ -0.078\,a - 1.058\,b + 1.289\,c - 1.160\,d + 0.759 \\
virginica &: \ -0.159\,a + 0.476\,b + 0.025\,c + 1.311\,d - 0.420
\end{aligned}
$$

これをブール関数で近似すると

$$
\begin{array}{lll}
setosa & : 1 - c & \rightarrow \bar{c} \\
versicolour & : (1-b)(1-d) + (1-b)cd + bc(1-d) & \rightarrow c\bar{d} \vee \bar{b}c \vee \bar{b}\bar{d} \\
virginica & : d & \rightarrow d
\end{array}
$$

となる。これで予測を行ない、回帰式をそのまま用いた予測の正解率との比較表を表 4.6 に示す。

近似後の方が上がっていることがわかる。これは、もともと植物の分類というデータでは定性的な構造があり、命題の形で表現しやすいのではないかということ、あるいは重回帰分析のままでは過学習が行なわれてしまったのではないかということが考えられる。これについては、さらに他のデータに対して調べる予定である。以上のあやめの例は、多重線形関数空間が数値データ（非記号、パターン）から定性的な命題を生成するのに有効であることを示している。またこれは同時に記号学習がパターン学習の近似であることを意味している。

4.2.6 連続属性の離散化

今までは連続属性をそのまま扱っていたが、ここでは連続属性を離散化して扱う [70]。連続属性の離散化は C4.5 のアルゴリズムを流用した。これはアルゴリズムの性能を C4.5 と比較するため、（非本質的な）連続属性の（離散化）での性能を同じにするためである。実験結果は表 4.7 に示す。表より分かる通り本アルゴリズムの方が C4.5 より良い。C4.5 はさまざまな改良工夫がなされているのに本アルゴリズムはそのような改良工夫（例えば変数選択）がなされていないのでさらに良くなると思われる。なお表中の $(2, m, c-2)$ は属性が 2 値、多値、連続でクラスが 2 値であることを意味する。

表 4.7: 実験結果

データ名	本アルゴリズム		C4.5	
	回帰式	命題	決定木	ルール
mushrooms(2,m—2)	99.3%	99.3%	98.7%	98.7%
voting-records(2—2)	97.0%	97.0%	93.0%	93.5%
zoo(2,m—m)	93.1%	84.9%	86.9%	87.8%
iris(c—m)	94.1%	94.1%	93.6%	93.6%
heart-disease(2,m,c—2)	77.1%	70.2%	66.9%	71.6%

4.2.7 誤差解析

データを離散に限定すれば誤差の理論的解析が可能になる。このアルゴリズムのサンプル計算量を簡単に説明する。ここで k 次の項とは k 個の変数を含んでいるものを言う。例えば xyz は 3 次である。また k 次までの項を含んでいる項を k- 多重線形関数と呼ぶ。仮説空間を k- 多重線形関数空間にした場合の正解と経験的な仮説間の誤差は経験的な仮説と最適な仮説の誤差と最適な仮説と正解の間の誤差の和となる。

まず経験的な仮説と最適な仮説の誤差は疑似次元（pseudo dimension[15]）で評価されることを述べる。この疑似次元は VC 次元を拡張したものである。

VC 次元は値域が $\{0,1\}$ の関数空間に対して定義されている [4]。Haussler は VC 次元を、値域が実数の関数空間に拡張して、疑似次元を定義した。Haussler の定理は次の通りである [15]。

d を仮説空間の疑似次元とする。h_t と h_e を各々最適な仮説と経験的な仮説とする。m をサンプル数とする。$|\bullet|$ をノルムとする。ϵ と δ を実数とする。もし重回帰に使われる多重線形関数の値域が [0,1] ならば，任意の確率分布に対して以下の式が成立する。

Probability$(|t-h| \geq \epsilon) \leq \delta$ if $m \geq \frac{a}{\epsilon^2}(2d log_2 \frac{b}{\epsilon} + log_2 \frac{c}{\delta})$,
但し a, b, c は係数である。

n 変数の場合、 k- 多重線形関数空間の疑似次元は n の多項式である [8]。 それゆえ、サンプル計算量は n, $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$ の多項式になる。

次に正解と最適な仮説との誤差は [22] によって評価される。

f をブール関数とする。一様分布の場合、 $\| f - g \| < \epsilon$ を満たす k- 多重線形関数 g が存在する。但し k は高々 $O(log(n/\epsilon)^2)$ である。

この式はブール関数が低次元の多重線形関数で良く近似されることを示している。誤差解析に関して簡単に述べたが、さらに詳細な解析は将来の課題である。

4.2.8 おわりに

本節では、多重線形関数のブール関数による近似に基づいたアルゴリズムを提示し、実験によってその有効性を確認した。今後の課題としては、回帰式から DNF 式を求めるとき k 次までで止めた場合の誤差の解析がある。また精度を向上させるために、変数選択や重回帰分析の回帰式に高次の項を追加する (例えば、2 変数では $ax + by$ に xy を追加して $ax + by + cxy$ とする) ことを検討する必要がある。

4.3 数値データから論理命題を見つける

4.3.1 はじめに

発見に関する研究はいくつかあるが [21]、数値データから論理命題をシステマティックに発見すると言うものはほとんど見当たらない。一方数値データを取り扱ってきたのは主に統計学であるが、そこでは数式による近似であるとか分類等であり論理命題を導くという研究はほとんどない。本節では数値データから論理命題を発見する手法を提示する。ここで取り扱う数値データは重回帰分析が取り扱う数値データと同じである。

歴史上、特に科学において、人間は数値データを検討することによってそのデータに伏在する様々な関係式を発見してきた。多くの科学的発見には必ずといって良いほどこのようなデータからの関係式の発見と言う作業が伴っている。その関係式は簡単な比例式から複雑な微分方程式までいろいろあるが、その様な式を導く各段階で、何を変数に選ぶとか、どのような定性的な関係がその変数間に成立しているとかの仮説を作る作業が必要になる。筆者はその定性的な関係の発見を自動的に行う手法の研究を行っているが、ここでは最も基本的な定性的関係である論理的関係を表現する論理命題を自動的に発見する手法について述べる。BACON[21] も数値データをとり扱っているがそこでは簡単な関係式をヒューリスティックスを用いて再発見している。ここでは論理命題をヒューリスティックスを用いずに発見する。

アルゴリズムの概略は以下の通りである。

1. データの [0,1] への正規化

2. 重回帰分析

3. 直交関数展開

4. ブール関数による近似

5. 命題化

6. 簡単化

またこの論理関数列が直交していることからこの手法はフーリエ解析の論理版ともみなせる。渡辺 [41] は論理スペクトルと言う概念を導入しているが、本論文ではその論理スペクトルを数値データから実際に求めていると言える。この重回帰分析の結果得られた論理関数は一般には非古典論理の命題に対応するが、ここではこれを古典論理の命題で近似し, 簡単な例で妥当な結果を得ることを確認する。さらにこの手法によって得られる命題が他の手法では必ずしも得られないことを述べる。このアルゴリズムにより数値データから論理命題を発見できることを実験で確認するが、時には適切な論理命題を発見できない場合もあり、そのような場合には直接、直交関数による重回帰分析が有効であることを述べる。このような手法により数値データから論理命題を自動的に発見することが可能になり、定性的な関係を発見すると言う作業を支援できるものと考える [63][67]。

4.3.2 重回帰分析によって得られる関数より論理命題を求める

重回帰分析によって得られる関数より論理命題を求める方法について

通常の重回帰分析の関数は定義域も値域も R である。今まで述べてきた拡張モデルの関数は定義域は $[0,1]$ で値域は R である。従って通常の重回帰分析の関数の定義域を適当な方法で $[0,1]$ にすることができれば、今までの説明により、通常の重回帰分析によって得られる関数が本論文での拡張モデル L に属すことになり、何らかの命題を表現していると解釈できるわけである。

本節では重回帰分析で得られた関数をそれに最も近いブール関数で近似して命題を得ることを考える。最も近いというのは 2 章で導入したノルムに関してである。このノルム以外にも関数の近さを評価する量は存在するであろう。関数解析においてもヒルベルト空間、バナッハ空間等の関数空間があり、関数の近さを評価する量はそれぞれ異なる。したがって 2 章で導入したノルムはそれらいくつか考えられる内の一つであるが、2 乗ノルム (ユークリッドノルム) はわれわれの直観に最も近いので最初にとりあげるにふさわしいノルムであると考える。他のノルムとの比較は将来の課題としたい。

重回帰分析で得られた関数 (f) に最も近いブール関数 (g)、即ち $\| f - g \|$ を最小にする g_i を求める。結果は $f_i \geq 0.5$ ならば $g_i = 1$ その他で $g_i = 0$ となる。

手順

前項議論より、手順は以下の通りである。

1. データの $[0,1]$ への正規化
2. 重回帰分析
3. 直交関数展開
4. ブール関数による近似
5. 命題化
6. 簡単化

(1) データの $[0,1]$ への正規化

重回帰分析の結果である関数を L に属させるために行う。正規化するに当たり以下の 3 つの場合が考えられる。

1. 上限、下限が存在する場合

   上限下限が存在する場合とは例えば 100 点満点の試験の場合である。この場合の上限は 100 であり、下限は 0 である。こう言う場合にはその上限、下限で正規化出来る。

2. 上限、下限は存在しないが、データの発生源の確率分布が分かっている時

   この場合はその既知の確率分布 (例えば正規分布等) を当てはめて上限下限を求めることができる。例えば上方 $100\alpha\%$ 点を上限、下方 $100\alpha\%$ 点を下限とする方法も考えられる。

3. 上限、下限が存在せず、データの発生源の確率分布も分からない時

この場合でももっともあり得そうな確率分布を当てはめて上限下限を求めることもできるが、与えられたデータの最大値、最小値をそれぞれ上限下限とする方法もある。ここでは最後の与えられたデータの最大値を上限に、最小値を下限にすることとする。データ (x) の最大値、最小値をそれぞれ a, b とすると正規化されたデータは $y = (x-b)/(a-b)$ となる。

(2) 重回帰分析

この処理は通常の重回帰分析である。

(3) 直交関数展開

2 章参照

(4) ブール関数による近似

3 章参照

(5) 命題化

得られた古典論理ベクトルを命題に直す。

(6) 簡単化

得られた命題を簡単化することであり、クワイン - マクラスキ法等様々な手法が提案されている。

簡単な例による説明

以下の例を考える。これは文献 [48] より引用した。表 4.8は金属のある性質（配向度）が時間と温度によりどう変化するかという表である。

(1) データの $[0,1]$ への正規化

表 4.8のデータを正規化すると表 4.9となる。

(2) 重回帰分析

これを重回帰分析すると

$$z = 0.46x - 0.41y + 0.38$$

表 4.8: ある金属の配向度

試料番号	温度（℃）(X)	時間（分）(Y)	配向度（%）(Z)
1	1700	30	36
2	1800	25	39
3	1800	20	44
4	1850	30	44
5	1900	10	59
6	1930	10	51

表 4.9: ある金属の配向度（正規化後）

試料番号	温度（℃）(X)	時間（分）(Y)	配向度（%）(Z)
1	0.000	1.00	0.000
2	0.435	0.75	0.130
3	0.435	0.50	0.348
4	0.652	1.00	0.348
5	0.87	0.00	1.000
6	1.000	0.00	0.652

となる。

(3) 直交関数展開

直交関数展開すると

$$z = 0.43xy + 0.84x(1-y) - 0.03(1-x)y + 0.38(1-x)(1-y)$$

になる。

(4) ブール関数による近似

これをブール関数で近似すると

$$z = 0xy + 1x(1-y) + 0(1-x)y + 0(1-x)(1-y) = x(1-y)$$

となる。

(5) 命題化

命題では $X \wedge \overline{Y}$ となる。

(6) 簡単化

簡単化してもこのままである。これを普通の表現を用いると「温度が高く時間が短ければ配合度が高い」となる。このようにして数値データから命題を得ることができる。

通常の重回帰分析との比較

ここでは、通常の重回帰分析とはデータを $[0,1]$ に正規化しない重回帰分析のことを指す。これに対しデータを $[0,1]$ に正規化した重回帰分析を正規化した重回帰分析と呼ぶ。本論文のアルゴリズムで得られた命題と通常の重回帰分析の結果とを比較する。通常の重回帰分析では

$$z = 0.04573x - 0.469y - 28.4$$

が得られる。この式は配合度が温度に比例し、時間に反比例していることを示している。従って本論文のアルゴリズムの結果である「温度が高く時間が短ければ配合度が高い」と同じことを意味している。この事から通常の重回帰分析

で得られる関数の各係数の符号の組合せから本論文のアルゴリズムで求めたのと同じ命題を得ることができると思われるかも知れないが、必ずしもそうとは限らない。この場合は簡単化が実質的になされていないから通常の重回帰分析からでも命題が得られるのである。簡単化をして得られる命題は通常の重回帰分析で得られる関数の各係数の符号の組合せからは得られない。係数の符号の組合せはこの例で言えば $++, +-, -+, --$($++$ は $ax+by+c$ において $a \geq 0, b \geq 0$ を意味する。)であり、命題で言えば $XY, X\overline{Y}, \overline{X}Y, \overline{XY}$ のブール代数の原子に相当する命題だけであり、$+$ と $-$ が簡単化できないため X のような簡単化がなされたような命題は得られない。例えば正規化したデータによる重回帰分析で

$$z = 0.5x - 0.2y + 0.3$$

が得られたとしよう。これを直交関数展開すると

$$z = 0.6xy + 0.8x(1-y) + 0.1(1-x)y + 0.3(1-x)(1-y)$$

となり、ブール関数で近似すると

$$xy + x(1-y) = x$$

となるが、これは通常の重回帰分析によって得られた関数の係数の符号の組み合わせからは得られない。

それでは、符号のみでなく係数の数値自体にも注目すれば命題の簡単化に相当する処理(大きい係数の変数のみ残し、小さい係数の変数は落す等)が可能であると思われるかも知れないが、それは各変数のレンジが同じでないので比較できないから、無理である。次にデータを $[0,1]$ に正規化した場合について述べる。

正規化した重回帰分析について

この場合も得られた関数の係数の符号の組合せに関しては通常の重回帰分析と同様に、得られた関数の係数の符号に注目してその符号の組合せから得られる命題はブール代数の原子に相当する命題だけである。しかしデータを $[0,1]$ に正規化しているため、得られた回帰式の係数を 0 か 1 で近似することによって命題が得られると思われるかも知れないが、以下のことより必ずしもうまく行かない。

(1) 得られた関数の係数を 0 か 1 で近似して得られるのは説明変数の組合せに限定される。例えば 2 変数で

$$z = 0.6x + 0.2y + 0.4$$

が得られたとしよう。この場合近似をすれば

$$z = x$$

となる。この様に説明変数とその選言で論理命題を表現をすることになるが、これではすべての論理命題を表現できない。2 変数の場合には $X, Y, \overline{X}, \overline{Y}$ とそれの選言の命題しか表現できず、$\overline{X}Y \vee X\overline{Y}$ 等の命題は表現できない。

(2) またその様な近似は粗過ぎる。この近似で得られたブール関数は x であり、x と $0.6x+0.2y+0.4$ との距離は $\sqrt{0.56}$ となる。但し関数 f, g 間の距離 $\| f-g \|$ は 2 章の定義より

$$\| f-g \| = \sqrt{<f-g, f-g>} = 2^2 \int_0^1 \int_0^1 \tau((f-g)(f-g))dxdy$$

となり、この場合

$$f = x,\ g = 0.6x + 0.2y + 0.4$$

として計算する。一方

$$0.6x + 0.2y + 0.4$$

を直交関数展開すると

$$z = 1.2xy + 1.0x(1-y) + 0.6(1-x)y + 0.4(1-x)(1-y)$$

となり、ブール関数で近似すると

$$z = xy + x(1-y) + (1-x)y$$

となる。この場合

$$0.6x + 0.2y + 0.4 (= 1.2xy + 1.0x(1-y) + 0.6(1-x)y + 0.4(1-x)(1-y))$$

と

$$xy + x(1-y) + (1-x)y$$

の距離は $\sqrt{0.36}$ となる。したがって正規化した重回帰分析によって得られた関数の係数を 0 か 1 で近似して得られる古典論理関数は必ずしも最も近い論理関数とはならないのである。さらに表 4.8の例では $z = 0.46x - 0.41y + 0.38$ が得られるが、この式の x, y の係数をそのまま 1,0 で近似しても得られるのは 0 となり、近似の粗さが、明らかにおかしい結果を生ずることになる。

即ち、表現力が限定されていることと近似が粗い事により必ずしも妥当なブール関数が得られないのである。以上により、正規化した重回帰分析により得た関数に最も近いブール関数が通常の重回帰分析や正規化した重回帰分析の係数からは必ずしも得られないことを述べた。

4.3.3 実験

表 4.8の例の通常の重回帰分析の結果が $z = 0.04573x - 0.469y - 28.4$ であり、この式が、配合度が温度に比例し、時間に反比例していることを示しているのは既に述べた。これはブール関数による近似の結果である「温度が高く時間が短ければ配合度が高い」と同じことを意味している。従って本論文の古典論理関数による近似が妥当であることが分かる。結果の妥当性は (1) 通常の重

回帰分析と比較できる範囲で、結果が同じである。(2) ブール関数による近似がユークリッドノルムに基づいているから理論的に妥当であろう。と言う 2 点であるが、この項では本アルゴリズムの性能に関する実験を行う。

実験 1

測定されるのは数値で、その背後に存在する法則、規則が論理的、定性的であるような例としては非線形素子で構成される神経回路、電子回路が考えられる。これらの非線形素子は種々の関数で近似できるがここでは閾値関数を考える。そこで閾値関数で論理的に結合されているような以下の関数を考える。

$$w(x,y,z) = P((f(x) \vee \overline{g}(y)) \wedge h(z))$$

ここで、 $f(x), g(y), h(z)$ は以下の通りである。

$$f(x) = 1(x \geq 0.5),\ = 0(x < 0.5)$$

$$g(y) = 1(y \geq 0.6),\ = 0(y < 0.6)$$

$$h(z) = 1(z \geq 0.4),\ = 0(z < 0.4)$$

また $P(t)$ は $t = 1$ で 0.5 と 1 の間の数値を、$t = 0$ で 0 と 0.5 の間の数値を、確率的に対応させる関数とする。但し、 $0 \leq x, y, z, w \leq 1$ である。閾値は f, g, h で各々 0.5, 0.6, 0.4 と異なる値にしてある。

そこで実験は x, y, z に適当な数値 (表 4.10) を与え上記の関数に従って w を生成しその数値から論理命題（この場合は $W = (X \vee \overline{Y}) \wedge Z$）を求めることになる。

これを重回帰分析すると

$$w = 0.2x - 0.2y + 0.4z + 0.2$$

となる。これを直交展開すると

表 4.10: 実験 1

x	y	z	w
0.9	0.6	0.6	0.5
0.7	0.9	0.1	0.2
1.0	0.5	1.0	0.7
0.6	0.5	0.2	0.3

$$0.6xyz + 0.2xy(1-z) + 0.8x(1-y)z + 0.4x(1-y)(1-z) + 0.4(1-x)yz + 0.0(1-x)y(1-z) + 0.6(1-x)(1-y)z + 0.2(1-x)(1-y)(1-z)$$

となり、ブール関数で近似すると

$$1.0xyz + 0.0xy(1-z) + 1.0x(1-y)z + 0.0x(1-y)(1-z) + 0.0(1-x)yz + 0.0(1-x)y(1-z) + 1.0(1-x)(1-y)z + 0.0(1-x)(1-y)(1-z)$$

なり、命題にすると

$$XYZ \vee X\overline{Y}Z \vee \overline{XY}Z = (X \vee \overline{Y}) \wedge Z$$

となる。従って目的の論理関数が得られたことになる。このような論理命題が数値データの背後に存在することは重回帰分析の式を見てもわからない。しかしこの方法で必ずしも正しい論理命題が求められるわけではない。次の実験でそれを示す。

実験 2

実験 2 では求めるべき論理命題を $XY \vee \overline{Y}Z$ としよう。そして表 4.11 の数値が与えられたとしよう。数値の発生の仕方は実験 1 と同様である。特に 10 番目のデータは正解の命題の反例であり、一種の雑音といえる。

これを重回帰分析すると

$$w = 0.28x - 0.21y + 0.23z + 0.27$$

となり、直交展開し古典論理関数で近似して論理命題を求めると

表 4.11: 実験 2

x	y	z	w
0.8	0.6	0.9	0.6
0.7	1.0	0.2	0.7
0.9	0.2	0.7	0.6
0.8	0.1	0.3	0.3
0.4	0.7	0.8	0.1
0.2	0.8	0.3	0.1
0.1	0.1	0.9	0.8
0.3	0.2	0.2	0.2
0.7	1.0	0.2	0.5
0.7	1.0	0.2	0.4

$$XYZ \vee X\overline{Y} \vee \overline{XY}Z$$

となり、正しい論理命題が得られない。このように本方式によって必ずしも正しい論理命題が得られるわけではない。正しい論理命題が得られるかどうかは求める論理命題と与えられた数値データに依存する。即ち線形関数による近似は荒いので、論理命題が「複雑」だと求められないし、「複雑」でなくても与えられた数値データが「悪い」と間違った論理命題が求められることになる。本手法の実験的、理論的分析は今後の課題としたい。ここでは本手法がうまく機能しない場合にも、直接、直交関数で重回帰分析するとうまくいくことを確認する。直接、直交関数で重回帰分析するとは 3 変数で言えば数値データを

$$ax + by + cz + d$$

で近似するのではなく

$$axyz + bxy(1-z) + cx(1-y)z + dx(1-y)(1-z) + e(1-x)yz + f(1-x)y(1-z) + g(1-x)(1-y)z + h(1-x)(1-y)(1-z)$$

で近似することである。このためにまず xyz から $(1-x)(1-y)(1-z)$ の 8 個の値を x, y, z から計算して重回帰のための式を作成する。例えば 1 番めのデータからは

$$0.432a + 0.048b + 0.228c + 0.032d + 0.108e + 0.012f + 0.072g + 0.008h$$

が得られる。このような処理をして重回帰分析を行なうと

$$w = 1.05xyz + 0.67xy(1-z) + 0.76x(1-y)z + 0.05x(1-y)(1-z) - 1.2(1-x)yz + 0.34(1-x)y(1-z) + 0.94(1-x)(1-y)z + 0.0(1-x)(1-y)(1-z)$$

が得られ、ブール関数で近似して論理命題に直すと

$$XY \vee \overline{Y}Z$$

となり、正しい論理命題が得られる。このような直交関数での近似は、線形関数による近似よりきめが細かいので正しい論理命題が得られるが、説明変数が n 変数の時に重回帰式の項の数は 2^n となるので、計算量の大きくなると言う問題がある。これは今後解決しなければならない問題である。

4.3.4 おわりに

表 4.8の例では配向度の説明変数として温度と時間を選んだ。その結果 (0.43, 0.84, -0.03, 0.38) と言うベクトルを得た。そしてこのベクトルを古典論理ベクトルで近似して (0, 1, 0, 0) を得た。この 2 つのベクトル間の距離は約 0.63 である。この距離は近似誤差と考えて良く、この距離が小さいほど良い近似であると言うことになる。表 4.8の例で言えば、もしこの距離が極めて小さければ、配向度は温度と時間で確定的に説明できることを意味し、逆にこの距離が大きいならば配向度は温度と時間だけでは確定的に説明できず、他の説明変数（例えば圧力とか湿度）を必要とするか、もしくはこの説明変数の組合せが適当でないことを意味する。従って例えば温度と圧力を説明変数にした時の距離が 0.2 であるならば、温度と圧力の組合せの方が温度と時間の組合せより配向度を良く説明する、即ちより強い因果関係があることになる。このようにこの距離は説明変数の選択基準になり得る。

本節で提案した手法はブール代数の原子に対応する論理関数が直交しているためフーリエ解析の論理版と見なすこともできる。従って、各論理関数の係数はいわば周波数成分であり、一種のスペクトル分析をしているとも見なせる [41]。

このような視点から直交関数展開の結果を見ると、得られたベクトルは被説明変数と説明変数間の因果関係の重みを意味しており、古典論理ベクトルに近似して使用しなくともそのまま利用可能であると思われる。

本節では、データを $[0,1]$ に正規化した重回帰分析により得られた関数がある非古典論理のモデルに属していると言う事実に基づき、その関数をブール関数で近似することによって、数値データから論理命題を発見する手法を提示した。今後この手法を更に発展させるとともに、判別分析等の他の多変量解析にも適用する予定であり、数値データと質的データが混在するようなデータベースから一般的規則を発見する手法を確立したいと考えている。これにより数値データから規則性や関係式を発見する作業の支援が可能となるであろう。なお今後の課題としては (1) 本手法の理論的、実験的分析 (2) ユークリッドノルム以外のノルムとユークリッドノルムの比較 等がある。

4.4 ニューラルネットワークの構造分析

4.4.1 はじめに

人工知能、認知科学等においていくつかのパラダイムが存在するが、記号主義とコネクショニズムが二大潮流であるといえよう。記号主義は、記号の形式的操作、即ち計算によって知能を実現しようとする主義である。常識推論等を古典論理を拡張することによって実現しようとするいわゆる論理主義 [23] はその典型である。記号主義に対しては人工知能、認知科学、哲学等からさまざまな批判がなされている。その多くは記号主義が状況依存性、身体依存性、パターン等を扱えないことに関するものである。これに対しコネクショニズム（ここではニューラルネットワーク、PDP のアプローチと同義に用いる。）[74] は主に神経回路網をモデルにしたネットワークを用いて各種計算を実行することによって知能を実現しようとする主義である。両者はそれぞれ長所短所があり、その統合の必要性は多くの研究者によって指摘されてきたし、またいくつかの試みもある [25][51]。[51] は記号とパターンの統合を実現する柔らかな論理を目

指しているが、ここで提示するパラダイムは基本的には柔らかな論理である。

記号主義とコネクショニズムの統合や対応の方式はいくつかあるが [74]、本論文で提示した多重線形関数空間は記号主義の基本部分である古典論理とコネクショニズムの基本部分であるニューラルネットワークを統一的に表現できるパラダイムである。多重線形関数空間を用いると種々の応用が可能と思われる。例えばニューラルネットワークをブール関数で（近似）表現したり、その逆であるブール関数をニューラルネットワークで表現したりすることが可能となる。本節ではまず非線形関数である出力関数の（近似）表現について述べ、ニューラルネットワークのブール関数による（近似）表現の具体的例として排他的論理和を学習したニューラルネットワークの構造分析について述べる。なおニューラルネットワークの構造分析に関しては [10] 等いくつかの方法が提案されているが本論文の方法はそれらとは異なる。

4.4.2 出力関数について

ニューラルネットワークは基本的には線形関数と出力関数で表現される。閾値関数にはステップ関数、シグモイド関数等各種ある。この出力関数に関してはここでは線形近似で対応し別の対応に関しては将来の課題とする。但し入力が $\{0,1\}$ のステップ関数の場合はブール関数近似がそのまま出力関数となる。以下にそれを示す。ステップ関数 $1(x)$ は 以下の通りである。

$$\begin{aligned} 1(x) &= 1(x>0) \\ &= 0(x \le 0) \end{aligned}$$

まず最初にステップ関数の出力を計算する。今ある素子の関数を

$$1(p_1x_1 + \ldots + p_nx_n - h)$$

としよう。但し h はバイアスである。$p_1x_1 + \ldots + p_nx_n$ の直交関数展開を

$$a_1x_1 \cdot \cdot x_n + \ldots + a_{2^n}(1-x_1)\cdot\cdot(1-x_n)$$

とすると、素子の関数は

$$1(a_1x_1\cdot\cdot x_n + ... + a_{2^n}(1-x_1)\cdot\cdot(1-x_n) - h)$$

となる。入力が $\{0,1\}^n$ であるから、その数値が入力された時は、上式の $a_i(i = 1,..,2^n)$ のうちどれか唯一つの a_i が残ることになる。従って上式は $1(a_i - h)$ となり, その出力は

$$a_i > h \rightarrow 1$$

$$a_i \leq h \rightarrow 0$$

となる。次にブール関数近似の出力を計算する。但しブール関数近似の際の閾値を 0.5 固定ではなく h とする。直交関数展開は以下の通りである。

$$a_1x_1\cdot\cdot x_n + ... + a_{2^n}(1-x_1)\cdot\cdot(1-x_n)$$

ブール関数近似は各 a_i に対して $a_i > h$ ならば 1、$a_i \leq h$ ならば 0 とするから、$\{0,1\}^n$ が入力された時はこの入力に対応した a_i が $a_i > h$ ならば出力は 1, そうでなければ、即ち $a_i \leq h$ ならば出力は 0 となる。これはステップ関数の出力と同じである。従って入力が $\{0,1\}$ の時のステップ関数はブール関数近似と同じとなる。以上の議論より入力が $\{0,1\}$ の時、出力関数がステップ関数ならば出力関数も多重線形関数空間の中で処理可能であることがわかった。従って出力関数がステップ関数もしくはそれに良く近似される関数である場合は、線形関数をブール関数で近似する際の閾値は 0.5 ではなく上記の h を用いて行なうことによって各素子を多重線形関数で表現できる。

4.4.3 排他的論理和の例

排他的論理和は $X\overline{Y} \vee \overline{X}Y$ である。まず排他的論理和を学習するニューラルネットワークの構成について述べ、次に学習結果の分析について述べる。

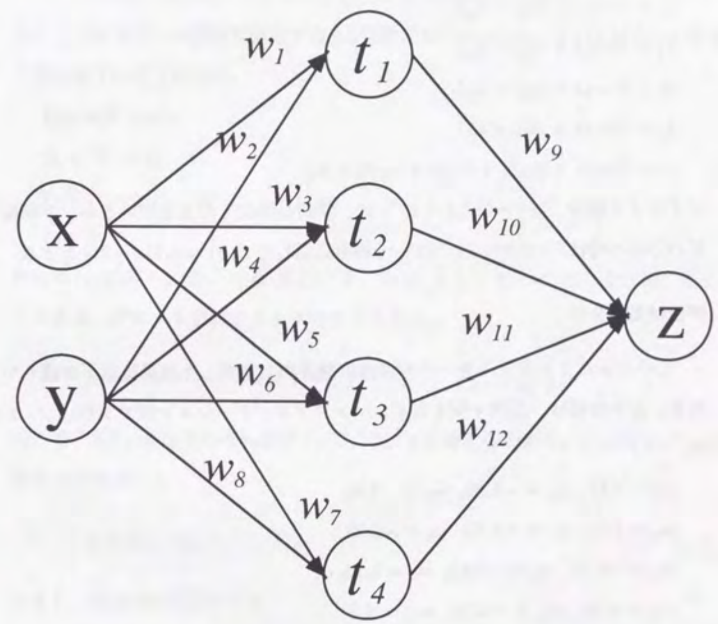

図 4.2: ニューラルネットワーク

ニューラルネットワークの構成

いま図4.2のようなニューラルネットワークの構成を考える。但し x, y は入力で、z は出力、t_i は中間出力である。w_i は重み係数である。中間出力、出力はシグモイド関数 $S(x) = 1/(1+e^{-x})$, バイアス h_i を用いて以下のように計算される。

$$t_1 = S(w_1x + w_2y + h_1),$$
$$t_2 = S(w_3x + w_4y + h_2),$$
$$t_3 = S(w_5x + w_6y + h_3),$$
$$t_4 = S(w_7x + w_8y + h_4),$$
$$z = S(w_9t_1 + w_{10}t_2 + w_{11}t_3 + w_{12}t_4 + h_5)$$

シグモイド関数 $S(x) = 1/(1+e^{-x})$ は、学習の時はそのまま用いるが、学習結果の分析の時は $(0, S(0)), (1, S(1))$ で線形近似して $S(x) = 0.23x + 0.5$ とする。

学習結果の分析

このニューラルネットワークで排他的論理和を学習した結果は以下の通りである。なお学習は、正例4例を与え、バックプロパゲーション法で1000回行なった。

$$w_1 = 2.51,\ w_2 = -4.80,\ w_3 = -4.90,$$
$$w_4 = 2.83,\ w_5 = -4.43,\ w_6 = -4.32,$$
$$w_7 = -0.70,\ w_8 = -0.62,\ w_9 = 5.22,$$
$$w_{10} = 5.24,\ w_{11} = -4.88,\ w_{12} = 0.31$$
$$h_1 = -0.83,\ h_2 = -1.12,\ h_3 = 0.93,$$
$$h_4 = -0.85,\ h_5 = -2.19$$

t_1, t_2, t_3, t_4 は以下のようになる。

$$t_1 = 0.58x - 1.1y + 0.31$$
$$t_2 = -1.1x + 0.60y + 0.25$$
$$t_3 = -1.0x - 0.99y + 0.70$$

$$t_4 = -0.16x - 0.14y + 0.30$$

これを直交展開すると各々以下のようになる。

$$t_1 = -0.22xy + 0.89x(1-y) - 0.79(1-x)y + 0.31(1-x)(1-y)$$
$$t_2 = -0.28xy - 0.88x(1-y) + 0.85(1-x)y + 0.25(1-x)(1-y)$$
$$t_3 = -1.3xy - 0.30x(1-y) - 0.29(1-x)y + 0.70(1-x)(1-y)$$
$$t_4 = 0.0xy + 0.14x(1-y) + 0.16(1-x)y + 0.30(1-x)(1-y)$$

そしてこれをブール関数で近似する (各係数が0.5以上で1、0.5以下で0とする) と各々以下のようになる。

$$t_1 = x(1-y)$$
$$t_2 = (1-x)y$$
$$t_3 = (1-x)(1-y)$$
$$t_4 = 0$$

従って t_1 は $X \wedge \overline{Y}$ を、t_2 は $\overline{X} \wedge Y$ を、t_3 は $\overline{X} \wedge \overline{Y}$ を、t_4 は 0 を学習したことになる。次に z を計算すると次のようになる。

$$z = 1.2x(1-y) + 1.2(1-x)y - 1.1(1-x)(1-y) + 0$$

但し第1,2,3,4項はそれぞれ素子1、2、3、4の寄与分である。この式をブール関数で近似すると

$$z = x(1-y) + (1-x)y$$

となり、論理式で表現すると

$$X\overline{Y} \vee \overline{X}Y$$

となり排他的論理和となる。また素子1 (t_1), 素子 $2(t_2)$ を外すと図4.2のニューラルネットワークは排他的論理和として機能しないが、素子3 (t_3)、素子 $4(t_4)$ は外しても図4.2のニューラルネットワークは排他的論理和として機能する。従って図4.2のニューラルネットワークは4個の素子の冗長性を活かした学習をしていないことになる。例えば素子3の学習結果の古典論理関数近似が $t_3 = x(1-y)$

で素子 4 の学習結果のブール関数近似が $t_4 = (1-x)y$ ならば 4 個の素子の冗長性を活かした学習と言えよう。以上のようにニューラルネットワークの構造分析がブール関数近似を用いてできるのである。上記の例はニューラルネットワークが学習した対象が論理関数であったが、一般には学習対象は論理関数ではない。また入力、出力が $\{0,1\}$ ではなく実数である。その場合には入出力は適当に正規化することによって $[0,1]$ とし、上記のような構造分析を行なう。その結果得られるものは学習対象の定性的構造である。ここで言う定性的構造とは例えば「温度が高い時は品質が悪い」等の命題である [67]。

4.4.4 学習の経過の分析

ここでは排他的論理和 $x\overline{y} \vee \overline{x}y$ を図 4.2のニューラルネットワークが学習する場合、各素子の学習した関数が学習回数とともにどう変わるかを見てみる。実験結果を表 4.12に示す。表より以下のことが分かる。

1. 学習回数 700 回で正解である排他的論理和を学習している。
2. 200 回、300 回で学習した関数 $\overline{xy}$ は、100 回で「学習した」関数 0 より、正解である排他的論理和 $x\overline{y} \vee \overline{x}y$ から遠い。従って 400 回は「無駄な学習」をしていると言える。
3. （表 4.12だけからでは分かりにくいが）700 回以降は冗長性がある。例えば素子 3 は冗長である。

4.4.5 連続関数の学習

ここでは $[0,1] \to R$ の関数の学習例を簡単に述べる。関数はいままでと同様の排他的論理和 $x\overline{y} \vee \overline{x}y$ であるが、定義域が $[0,1]$ である。従って学習に使う例は乱数で発生させた $[0,1]$ 内の実数とその関数の値である。またニューラルネットワークも今までと同様の中間素子 4 個のものとする。30 個の事例を 600 回学習させた結果、中間素子は

表 4.12: 学習した関数

回数	素子 1	素子 2	素子 3	素子 4	全体
100	0	$\overline{xy}$	0	0	0
200	0	$\overline{xy}$	0	0	0
300	0	$\overline{xy}$	0	0	0
400	$\overline{xy}$	$\overline{xy}$	0	xy	0
500	$\overline{xy}$	$\overline{y}$	0	xy	$x\overline{y}$
600	$\overline{xy}$	$\overline{y}$	0	xy	$x\overline{y}$
700	$\overline{xy}$	$\overline{x} \vee \overline{y}$	$\overline{x}y$	xy	$x\overline{y} \vee \overline{x}y$
800	$\overline{xy}$	$\overline{x} \vee \overline{y}$	$\overline{x}y$	xy	$x\overline{y} \vee \overline{x}y$
900	$\overline{xy}$	$\overline{x} \vee \overline{y}$	$\overline{x}y$	xy	$x\overline{y} \vee \overline{x}y$
1000	$\overline{xy}$	$\overline{x} \vee \overline{y}$	$\overline{x}y$	xy	$x\overline{y} \vee \overline{x}y$

$$\overline{xy}, x\overline{y}, \overline{x}y, 0$$

を学習し、全体としては

$$x\overline{y} \vee \overline{x}y$$

を学習した。従ってニューラルネットワークの構造分析が実数領域でも有効であることを実験的に確認したことになる。

4.4.6 おわりに

本節では多重線形関数空間が記号主義の基本部分である古典論理とコネクショニズムの基本部分であるニューラルネットワークを統一的に表現できると言う事実に着目して、学習されたニューラルネットワークの構造分析が古典論理関数で近似可能であることを排他的論理和の例で示した。今後は出力関数の処理等の改良を加え、大規模なニューラルネットワークの構造分析等の応用を行ないたい。

4.5 予測アルゴリズム

4.5.1 スペクトル

本節では Linial[22] が提示した直交関数系とその [0,1] への拡張について述べる。スペクトル技術は信号処理を始めとする多くの理工学の分野で使われている。しかし、論理回路の分野でもその技術は発展してきた [16]。Linial [22] はそのスペクトル理論に基づく予測アルゴリズムを提示した。その際用いた直交関数を簡単に紹介する。関数空間は本論文での多重線形関数空間であり、関数は $f:\{0,1\}^n \to \mathbf{R}$ である。内積、ノルムは重み係数を無視すれば同じである。直交関数は次の通りである。

$$\chi_S(x_1,...,x_n) = \begin{cases} +1 & \text{if } \sum_{i \in S} x_i \text{ が偶数} \\ -1 & \text{if } \sum_{i \in S} x_i \text{ が奇数} \end{cases}$$

但し、S は変数の添字の部分集合である。なお n 変数のブール関数の値域は $\{-1,1\}$ である。 $f:\{0,1\}^n \to \{-1,1\}$.

4.5.2 [0,1] への拡張

まず最初に直交関数 χ_S を本論文の初等代数モデルで書き換えると以下のように表現される。

$$\chi_S = \prod_{i \in S}(1-2x_i)$$

そしてこの直交関数系はそのまま [0,1] に拡張できる。即ち次の直交関数は直交性と単位性を満たす。

$$< \chi_i, \chi_j >= \begin{cases} 0(i \neq j), \\ 1(i = j), \end{cases}$$

但し内積は次のように定義される。

$$< f,g >= \int_0^1 \tau(fg)dx.$$

4.5.3 予測アルゴリズム

ここでは [22] の予測アルゴリズムを簡単に説明し、そのアルゴリズムを [0,1] に拡張する。確率分布 D に対して、あるアルゴリズムが次の条件を満たせば、そのアルゴリズムを (ϵ,δ,D) 予測アルゴリズムと呼ぶ。

$\hat{f}$ が f と入力の ϵ 部分以上で不一致になる確率が δ 以下である。但し $\hat{f}$ は f の予測値である。

[22] は深さ d サイズ M の回路の (ϵ,δ,U) 学習アルゴリズムを提示した。但し U は一様分布である。ある d に対して以下のアルゴリズムは ϵ,δ, and M の準多項式時間アルゴリズムである。

学習段階

アルゴリズムは f の値を m 個のランダムなサンプル点 $x_1,...,x_m$ で観察する。但し $m = 4(2n^k/\epsilon)\log(2n^k/\delta)$、 $k = (20\log(2m/\epsilon))^d$。その F の S 番めの係数は

$$\alpha_S = \frac{\sum_{i=1}^m f(x_i)\chi_S(x_i)}{m}$$

である。但し $|S| > k$ を満たす全ての S に対して $\alpha_S = 0$ である。

予測段階

f の予測値は以下の通りである。

$$\hat{f} = sign(\sum_{|S| \leq k} \alpha_S \chi_S(x))$$

既に述べた通り直交関数 χ_S を本論文の初等代数モデルで書き直すと

$$\chi_S = \prod_{i \in S}(1-2x_i)$$

となり、これをこのまま [0,1] に拡張することができる。従って [22] の予測アルゴリズムは [0,1] に拡張しても有効であると言える。但しブール関数の表現が本論文では $f:\{0,1\}^n \rightarrow \{0,1\}$ であるのに対し [22] では $f:\{0,1\}^n \rightarrow \{-1,1\}$ である。それゆえそのまま使えるわけではなく、重み係数の調整が必要である。当然の事ながら [0,1] に拡張しても (ϵ, δ, U) algorithm であるかは保証されていない。従ってその理論的検討は今後行なわねばならない。

第 5 章

結論

学習は大きく二つに分けると、一つは記号学習であり、もう一つはパターン学習である。記号学習とは人工知能の帰納学習等の主に非数値的データから定性的命題を獲得する学習である。パターン学習とは多変量解析やニューラルネットワーク等を用いて主に数値データを学習して予測するものである。記号学習の欠点の一つはクラス（被説明変数）が連続値の場合には有効ではないと言うことであり、パターン学習の欠点の一つは予測式のみを求めて定性的命題を求めてないということである。また記号学習とパターン学習は原理的にも異なる。記号学習の原理は、もちろん種々あって一概に要約するのは難しいが、抽象化が主であるのに対し、パターン学習の原理は基本的には誤差等の評価量を最小にすることである。

本論文では上記の記号学習とパターン学習の欠点を解決することを目指した。即ちクラス（被説明変数）が連続値の場合にも有効であり、ニューラルネットワークからや回帰式から定性的命題を獲得できるアルゴリズムを提示した。また原理面に関して記号学習のアルゴリズムがパターン学習の原理で得られることを示した。即ちパターン学習の原理が記号学習でも有効であることを述べた。

本論文の定性的命題獲得アルゴリズムの基本は多重線形関数をブール関数で近似することであった。そこで 2 章でまず多重線形関数について説明した。それは次の手順で行なった。

1. 古典論理の初等代数モデルを提示した。

2. これを [0,1] に拡張することによって連続値論理を得た。

3. 関数空間を拡張することによって多重線形関数空間を得た。

4. この関数空間が非独立事象の確率関数の関数空間に対応していることを示した。

5. この関数空間を内積の導入によりユークリッド空間にした。

6. 情報量を導入し、このユークリッド空間内での論理命題の表現を調べた。

3 章では多重線形関数のブール関数による近似手法について述べたが、最初に基本的なアルゴリズムを提示してその手法が準最尤法になっていることを説明した。次に回帰関数が線形の場合の効率的なアルゴリズムを提示した。多重線形関数をデータから求める手法は種々存在するが、それを重回帰分析で求めるとアルゴリズムは以下のようになった。

1. データを重回帰分析して多重線形関数を得る。

2. 多重線形関数をブール関数で近似する。

重回帰分析は最小 2 乗法であり、多重線形関数のブール関数近似は準最尤法であるから、これらパターン学習の原理で記号学習のアルゴリズムが得られたことになる。

定性的命題獲得の研究としては人工知能の帰納学習があるが、それと比較して本アルゴリズムの優位性は以下の通りであった。

1. 性能が良い。

2. クラス (被説明変数) が連続値の時でも有効である。

1 項に関しては人工知能の帰納学習の代表的なアルゴリズムである C4.5 と実問題で比較実験して本アルゴリズムの方が優秀であることを示した。2 項に関しては人工知能の帰納学習ではクラス（被説明変数）が連続値の時には有効では

ないが、本アルゴリズムは有効であり、それを簡単な例で示した。また上記の応用はブール関数で近似する多重線形関数を重回帰分析で求めたのであるが、多重線形関数をニューラルネットで求めることも考えられる。この場合にはニューラルネットワークの構造分析と言う応用になり、本アルゴリズムがニューラルネットワークの構造分析にも有効であることを簡単な例で確認した。

謝辞

本論文を作成するにあたり多くの方から支援を頂きました。特に、御指導して頂きました東京大学電子情報工学科の石塚満先生に感謝します。 また、専門分野が違うにもかかわらず、議論をして頂き、有益な助言を頂きました東京大学計数工学科の北森俊行先生（現在、法政大学）に感謝します。

参考文献

[1] F. Bacchus: *Representing and Reasoning with Probabilistic Knowledge*, The MIT Press, 1990.

[2] F. Bacchus, A. Grove, J.Y. Halpern and D. Koller: From statistics to beliefs, *Proceedings of 10th AAAI*, pp.601-608, 1992.

[3] G. Birkhoff and T.C. Bartee: *Modern Applied Algebra*, McGraw-Hill, 1970（一松訳，現状応用代数，新曜社，1973）.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler and M. K. Warmuth: Learnability and the Vapnik-Chervonenkis dimension *J. ACM*, Vol.36, No.4, pp.929-965, 1989.

[5] A. Bundy: Incidence calculus, *Edinburgh Research Paper*, No.497, 1990.

[6] R. Carnap: *Logical Foundations of Probability*, The University of Chicago Press, 1950.

[7] R. Carnap: *Philosophical Foundations of Physics*, Basic Books, 1966 (沢田他訳，物理学の哲学的基礎，岩波書店，1968）.

[8] R. M. Dudley: Central limit theorems for empirical measures, *Ann. Prob.*, Vol.6, No.6, pp.899-929, 1978.

[9] C. Elkan: The paradoxical success of fuzzy logic, *Proceedings of 11th AAAI*, pp.698-703, 1993.

[10] J.L. Elman: Finding structure in time, *Cognitive Science*, Vol.14, pp.179-211, 1988.

[11] R. Fagin and J.Y. Halpern: A logic for reasoning about probabilities, *Proceedings of 3rd IEEE annual symposium on Logic in Computer Science*, pp.410-421, 1988.

[12] G. Frege: Ueber Sinn und Bedeutung, In P. Geach and M. Black eds, *Translations from the Philosophical Writings of Gottlob Frege*, Blackwell, Oxford, 1952.

[13] J.Y. Girard: Linear logic, *Theoretical Computer Science*, 50, pp.1-102, 1987.

[14] V.N. Grisin: Predicate and set theoretic calculi based on logic without contraction, *Math. USSR Izvestija 18*, pp.41-59, 1982.

[15] D. Haussler: Decision theoretic generalization of the PAC Model for neural net and other learning applications , *Information and computation*, pp.78-150, 1992.

[16] S.L. Hurst, D.M. Miller and J.C. Muzio: *Spectral Techniques in Digital Logic*, Academic Press, 1985.

[17] I. Kant: *Kritik der Reinen Vernunft*, 1787.

[18] J.M. Keynes: *A Treatise on Probability*, Macmillan, London, 1921.

[19] S. Koppelberg: *Boolesche Algebren* (現代のブール代数, 広瀬監訳, 共立出版, 1986).

[20] S.A. Kripke: *Naming and Necessity*, Harvard University Press, 1980 (八木沢, 野家 訳, 名指しと必然性, 産業図書, 1985).

[21] P. Langley, G. Bradshaw and H.A. Simon: BACON.5:The discovery of conservation laws,*Proceedings of the Seventh International Joint Coference on Artificial Intelligence*, pp.121-126, 1981.

[22] N. Linial, Y. Mansour and N. Nisan: Constant depth circuits, Fourier transform, and learnability, *Journal of the ACM*, Vol.40, No.3, pp.607-620, 1993.

[23] J. McCarthy: Applications of circumscription to formalizing common-sense knowledge, *Artificial Intelligence*, Vol.28, pp.89-116, 1986.

[24] D. McDermott and J. Doyle: Non-monotonic logic I, *Artificial Intelligence*, Vol.13 1/2, pp.41-72, 1980.

[25] M. Minsky: Logical versus analogical or symbolic versus connectionist or neat versus scruffy, *AI MAGAZINE*, pp.34, Summer, 1991 (「論理」対「類推」, あるいは「記号的」対「コネクショニスト」, あるいは「整然とした」対「雑然とした」について, 日経 AI 別冊 1991 秋号, pp.156-173, 1991).

[26] N. Nilsson: Probabilistic logic, *Artificial Intelligence*, Vol. 28, pp.71-87, 1986.

[27] H. Ono and Y. Komori: Logics without the contraction rule, *J. Symbolic Logic* 50, pp.169-201, 1985.

[28] J. Pearl: Probabilistic semantics for nonmonotonic reasoning:A survey, *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pp.505-516, 1989.

[29] H. Putnam: Meaning and reference, *Journal of Philosophy*, 70, pp.699-711, 1973.

[30] W.V.O. Quine: *From a Logical Point of View*, Harper & Row, 1963.

[31] J.R. Quinlan: Induction of decision tree, *Machine Learning 1*, pp.81-106, 1986.

[32] J.R. Quinlan: Unknown attribute values in induction, *Proceedings of the 6th International Workshop on Machine Learning*, pp.94-98, 1989.

[33] J.R. Quinlan: *C4.5: Programs for machine learning*, Morgan Kaufmann Pub., 1993.

[34] R. Reiter: A logic for default reasoning, *Artificial Intelligence*, Vol.13 1/2, pp.81-132, 1980.

[35] C.E. Shannon and W. Weaver: *The Mathematical Theory of Communication*, Univ. III. Press, 1949 (長谷川, 井上 訳, コミュニケーションの数学的理論, 明治図書, 1969).

[36] P.F. Strawson: *Individuals-An Essay in Descriptive Metaphysics*, Methuen, London, 1959 (個体と主語, 中村秀吉訳,

みすず書房, 1978）.

[37] H. Tsukimoto and C. Morita: The discovery of propositions in noisy data, *Machine Intelligence 13*, pp.143-167, Oxford University Press, 1994 .

[38] H. Tsukimoto: The discovery of logical propositions in numerical data, *AAAI'94 Workshop on Knowledge Discovery in Databases*, pp.205-216, 1994.

[39] H. Tsukimoto: On continuously valued logical functions satisfying all axioms of classical logic, *Systems and Computers in Japan*, A John Wiley & Sons, Inc. Company, 1994 (To appear).

[40] H. Tsukimoto and C. Morita: Efficient algorithms for inductive learning-An application of multi-linear functions to inductive learning, *Machine Intelligence 14*, Oxford University Press, 1995 (To appear).

[41] S. Watanabe: *Knowing and Guessing*, John Wiley and Sons, 1969 （知識と推測, 村上, 丹治訳, 東京図書, 1975）.

[42] A.N. Whitehead and B. Russel: *Prinpicia Mathematica*, Vol.1, 2nd edition, Cambridge, 1927.

[43] R.O. Winder: Single state threshold logic, *Switching Circuit Theory and Logical Design*, AIEE Special Publication Vol.S-134, pp.321-332, 1961.

[44] L. Wittgenstein: *Tractatus Logico-Philosophicus*, London, 1921.

[45] L.A. Zadeh: Fuzzy algorithms, *Information and Control*, Vol.12, pp.94-102, 1968.

[46] 甘利 俊一, 長岡 浩司: 情報幾何学の方法, 岩波書店, 1993.

[47] 有川 節夫, 西野 哲朗: 学習における計算論的アプローチ, 情報処理, Vol.32, No.3, pp.217-225, 1991.

[48] 有馬 哲, 石村 貞夫: 多変量解析のはなし, 東京図書, 1987.

[49] 安西 祐一郎: 認識と学習, 岩波書店, 1989.

[50] 飯田 隆: 言語哲学大全 2, 勁草書房, 1989.

[51] 大津 展之, 田村 浩一郎: 柔らかな論理をめざして, 情報処理 Vol.28, No.5, pp.629-636, 1987.

[52] 奥野 忠一, 久米 均, 芳賀 敏郎, 吉沢 正: 多変量解析法, 日科技連, 1971.

[53] 小野 寛: 非標準論理の現状とその展望, 情報処理, Vol.30, pp.617-625, 1989.

[54] 坂元 慶行, 石黒 真木夫, 北川 源四郎: 情報量統計学, 共立出版, 1983.

[55] 高島 文次郎, 森田 千絵, 月本 洋: 論理の幾何的モデルに基づいたルールと事例の協調推論の枠組, 情報処理学会研究報告, 93-AI-88, 88-1, 1993.

[56] 高島 文次郎: 私信, 1994.

[57] 月本 洋: 連続値論理数学について, 情報処理学会研究報告, 88-AI-60, 60-5, 1988.

[58] 月本 洋: 命題論理の幾何的モデル, 情報処理学会論文誌, Vol.31, pp.783-791, 1990.

[59] 月本 洋: 確率論理の幾何的モデルと帰納推論への応用, 情報処理学会研究報告, 91-AI-75, 75-1, 1991.

[60] 月本 洋: 確率データからの帰納学習 - その理論的基礎について -, 人工知能学会研究会資料, SIG-FAI-9201, pp.11-20, 1992.

[61] 月本 洋: 確率データからの帰納学習, 人工知能学会誌, Vol.7, No.5, pp.870-876, 1992.

[62] 月本 洋: 情報量に基づく不確実な命題の表現 - 常識推論の為の新しい論理的枠組の一応用 -, 人工知能学会研究会資料, SIG-F/H/K/S/I-9201, pp.57-64, 1992.

[63] 月本 洋: 数値データからの帰納学習 - 論理関数による重回帰分析 -, 情報処理学会全国大会講演論文集, Vol.3, pp.39-40, 1992.

[64] 月本 洋: 古典論理の全ての公理を満たす連続値論理について, 多値技報, Vol.MVL-93, No.1, 電子情報通信学会, pp.26-33, 1993.

[65] 月本 洋: ブール関数の多項式時間学習アルゴリズム, 情報処理学会研究報告, 93-AI-87, 87-12, 1993.

[66] 月本 洋: 不確実な知識を表現する非古典論理のモデル, 人工知能学会誌, Vol.8,No.3, pp.367-376, 1993.

[67] 月本 洋: 数値データからの論理命題の発見, 人工知能学会誌, Vol.8, No.6, pp.752-759, 1993.

[68] 月本 洋: 記号主義とコネクショニズムの統合パラダイム, 電子情報通信学会技術研究報告, NC93-53, 1993.

[69] 月本 洋: 古典論理の全ての公理を満たす連続値論理について, 電子情報通信学会論文誌, Vol.J77-D-I No.3, pp.247-252, 1994.

[70] 月本 洋, 下郡信宏, 森田千絵: 回帰分析に基づく帰納学習アルゴリズム, 情報処理学会研究報告, 94-AI-96, 96-2, 1994.

[71] 月本 洋: パターン処理の近似としての記号処理, 電子情報通信学会論文誌, 1994 (掲載予定).

[72] 永井 成男, 大窪 徳行: 帰納的確率と様相の論理, 早稲田大学出版部, 1986.

[73] 林 知己夫, 樋口 伊佐夫: 情報処理と統計数理, 産業図書, 1970.

[74] 宮田 義郎: 認知科学における PDP 的アプローチ, 人工知能学会誌, VOl.7, No.5, pp.779-785, 1992.

[75] 向殿 政男: Fuzzy 論理関数について, 別冊「数理科学」ファジー理論への道, サイエンス社, pp.85-92, 1988.

[76] 本橋 信義: 現代論理学入門, 岩波書店, 1989.

[77] 森田 千絵, 月本 洋: 確率データからの帰納学習のアルゴリズムについて, 人工知能学会第 6 回全国大会論文集, Vol. I, pp.153-156, 1992.

[78] 森田 千絵, 月本 洋: 情報量に基づく帰納学習アルゴリズムの実験的解析, 人工知能学会研究会資料 SIG-F/H/K/S/I-9201, pp.49-56, 1992.

[79] 森田 千絵, 月本 洋: 多項式時間帰納学習アルゴリズム, 情報処理学会研究報告, 93-AI-90, 90-1, 1993.