九州大学学術情報リポジトリ
Kyushu University Institutional Repository

ニューラルネットワークと遺伝的プログラミングによるルール生成手法を用いたクレジット審査システムの構成

時永，祥三
九州大学大学院経済学研究院 経済工学部門 ： 教授

岸川，善紀
秋田県立大学システム科学技術学部 経営システム工学科 ： 助手

https://doi.org/10.15017/10593

出版情報：經濟學研究. 72 (5/6), pp.1-17, 2006-05-30. 九州大学経済学会
バージョン：
権利関係：

KYUSHU UNIVERSITY

ニューラルネットワークと遺伝的プログラミングによるルール生成手法を用いたクレジット審査システムの構成

時永 祥三　岸川　善紀

1　まえがき

情報システムにおける容量の拡大や処理速度の向上などにより, 個人や企業に関する詳細なデータが利用可能となっている。これにともなって各種の新しいデータ解析の手法が提案されているが, これらの中で資金貸付 (クレジット) を行う場合の判別・審査を自動化することは, 単に金融機関における経営の効率化の視点からだけではなく, 説明性の確保などからも必要とされている。従来の判別手法としては, 多変量解析, ニューラルネットワークなどの数値的な手法に偏っており, 判別のルールを基本としてクレジット拒否の理由を説明することが可能なシステムとしては構成されていない [1]-[4]。本論文では, ニューラルネットワークと遺伝的プログラミング (Genetic Programming:GP) によるクレジット判別ルール生成システムの構成について述べ, 公開されているデータを用いて検証する。

クレジット判別・審査について, 最近エントロピーに基づく決定木を構成する手法の開発など, 判別のルールを言語的に出力し, 貸付を拒否した場合の説明を行うこころみがなされている [5]-[9]。これらの中でも, ニューラルネットワークに基づく方法は, 関数近似能力と言語的なルール生成を統合したものである [6]-[9]。Nefclass は 3 階層構造の階層型ニューラルネットワークであり, ニューラルネットワークを調整する方法が用いられている [8]。しかし, 実現方法は簡単ではないうえに, ファジイ集合により表現されたルールを日常的な言語に変換することも, それほど単純ではない。Trepan は学習手法により簡単化されたニューラルネットワークの構成結果をもとにして, ルール抽出を行うものであるが, このような枝の生成におけるデータ利用が, いわゆる発見的に実施される問題がある [9]。Neurorule は簡単化されたニューラルネットワークからルールを抽出する方法であり, 入力信号と中間層の間の関係をルールにより表現し同時に, 中間層のユニットの信号レベルを用いて出力レベルとの関係もルールとして表現し, これらを調整して入力変数と出力レベルとの間の関係をルールとしてまとめる [7]。しかし, 操作が複雑であり, 命題が論理積で結合されて表現された形式でしかルールを表現できない問題がある。

本論文では, このプロセスをニューラルネットワークによる入力変数の離散化操作と,GP によるルール自動生成の性能を結合することにより簡素化している。われわれは, これまでの研究において GP 手法を関数近似やエージェントの学習と行動の記述, あるいは株価などの時系列認識に適用し, その有効性を確認している [12]-[21]。本論文では, この GP 手法を拡張し, クレジットにおけるルール生成へと適用している。具体的には, 数値型あるいはカテゴリ変数を入力とするニューラルネットワーク構成において, 入力を離散化し, かつこれらの結合の中で切断可能な結合を見出し, 本質的に必要な入力を求め簡素化する。次に, このように簡素化・離散化された入力を論理変数とする判別ルールを GP により自動生成する。このような GP 手法を適用することにより, ルール生成を自動化できるとともに, 適合度によりルールの推定精度を数値化できるメリットがある。

本論文のシステムを用いたクレジット判別の応用例として, 代表的なクレジット審査データを用いた検証と, 日本企業の格付問題などへとアルゴリズムを適用する。

以下,2. ではニューラルネットワークと GP によるルール生成の概要について述べ,3. ではに

ニューラルネットワーク手法の概要ついて示している。4. においては GP 手法の基本とルール生成について述べ,5. で応用例について述べている。

2 ニューラルネットワークとGPによるルール生成

2.1 クレジット判別・審査のモデル化

企業への資金貸付 (クレジット) においては, 対象となる企業の財務情報や経営に関する定性的な情報をもとに, 貸し付けた資金が回収できるかが検討される。クレジットと同様の概念ではないが, 企業の発行する債券をランク付けする, いわゆる格付は, 債券売買だけに限定されないで, 一般的に企業の社会的な信頼度をカテゴリ化するものとして認識されている [1][2][11]。格付けを実施する機関については, 米国の Standard and Poors や Moody などがよく知られているが, 日本では経済データの提供会社などが調査と合わせて副次的に公表することが多い。同様に, 個人へのクレジットについては, 個人の職業や預金などの属性を用いて貸付の可否の判別が行われる。

クレジットの適否の判別を数値的に行う方法は, さまざまに提案されており, 伝統的な手法としてはロジステック回帰分析や, 多変量解析における判別分析などがある [1][2]。金融機関が企業への貸付けを実施する場合, 対象企業の財務指標や借入における条件などを入力データとして, その信用リスクを計測することが重要となる。この手法に関しては, 古くはアルトマンらの線形判別関数法などが代表的であったが [1][2], 最近では, ニューラルネットワークによる手法がさまざまに適用されている [10][11]。あるいは,2 分木を用いる方法や, 近年行われているニューラルネットワークによる方法がある [10][11]。

ニューラルネットワークによる手法は, 基本的には, このシステムが任意の関数を近似する能力を持っていることに注目して, 非線形の判別関数を構成することに原理を置いている。この場合, ニューラルネットワークの構成における重みが, 逐次近似により求まること, その収束が保証されていることが, 大きな利点となっている [26]。

しかしながら, ニューラルネットワークによる手法は数値計算手法であり, 信用リスクを判断する場合の言語的な解釈や, 融資に当たっての細かな指導方針などを作成するには適当ではない。例えば, 銀行から見た場合の顧客である資金の借り手に対しては, 貸付ができないという結果だけではなく, 企業としての改善事項などを指摘することが, 将来の関係として望ましいであろう。また, 米国では, 1976 年に施行された法律 The Equal Credit Opportumities においては, 個人への貸付のついて, 性別, 年齢, 人種, 職業などを理由として差別してはならないことが明記されており, 根拠として言語的な説明が必要とされている。

このようなことから, 近年, 言語的な方法によるリスク分析手法が提案されている [4]-[9]。具体的には, カテゴリ変数により記述された対象を, プロダクションルールによる判別の 2 分木をもとにして判別をする方法である。代表的な事例として,Quinlan によるエントロピー法が良く知られており, 2 分木を構成する場合に, 最も有効性の高い変数からルールに組み入れていく方法である [5]。最初に提案された方法では, カテゴリ変数だけが対象であったが, その後, 数値型変数も組み入れる方法へと拡張されている。

しかし, このプロダクションルールによる判別基準の作成方法にも問題があり, 利用できるサンプル対象が, 変数利用が進むに従って少なくなることであり, 全部のサンプルを有効に利用することができない。また, エントロピーだけが基準となっており, 最適なルールへの改善の方法であるかどうかは判断できない。また, これと同じような問題であるが, ニューラルネットワークによる非線形の関数近似能力に相当するものが保証されていない。

2.2 Neurorule などの従来手法

以上のような問題を解決する 1 つの方法として, ニューラルネットワークの近似能力および数値的な判別システムを, カテゴリ化や簡単化するなどの方法を用いて修正することにより, 言語的な

解釈を生成する中間的な方法が提案されている [7]-[9]。これにより, ニューラルネットワークによる強力な近似能力と同時に, 言語的な解釈を与えることができる。

このような理由から, 数値的なクレジットの判断手法の開発に加えて, これらの結果をルールとして言語的に説明するシステムを開発する努力がなされている [5]-[9]。以下では, これまで提案されてきた, ニューラルネットワークを基礎とした言語的な判別ルールの生成手法の概要についてまとめる。これらの手法は, 次の 3 つのグループに集約される。

(1)Neurorule(Baesens,Setino,Vanthienen による手法)
(2)Trepan(Craven,Shavik による手法)
(3)Befcalss(Nauck による手法)

更に, これらの結果を決定テーブル (Decision Table) として表示する方法や, グラフィックスの手法により視覚的に表現する方法などが提案されている。

これらの 3 つの手法に共通する点は, 連続的な数値, あるいは複数のカテゴリ値として与えられている入力データを, できるだけ少ない離散化されたデータに変換して用いることにある。この離散化の手段として, ニューラルネットワークが用いられている。例えば, 預金額が 100 万円以下 100 万円以上 200 万円未満など, これらの金額をしきい値として 4 ランクにカテゴリ化する。このカテゴリ化した区分を 4 ビットの 2 進数で, 表現しこれを連続変数に代わる入力として用いる。

Nefclass は 3 階層構造の階層型ニューラルネットワークであり, その動作原理によりファジイタイプのパーセプトロンともよばれている [9]。多くのファジイシステムの場合と同様に, 中間層のユニットの数は多くなる傾向にあり, これにともなってファジイルールとして定義される決定ルールの数も多くなる。このような問題を解決するために, ルールの中で重要なものから順に k 個だけを選択しながら, 再度ニューラルネットワークを調整する方法が用いられている。このように,Nefclass の実現方法は簡単ではないうえに, ファジイ集合により表現されたルールを日常的な言語に変換することも, それほど簡単ではないことが問題となる。

Trepan は学習手法により簡単化されたニューラルネットワークの構成結果をもとにして, 従来から用いられている記号的な学習 (symbolic learning) 方法によるルール抽出を行うものである [8]。すなわち, 次々に判別を実施するための木構造 (枝) を生成していきながら, あらかじめ定めた評価基準を満たすようになった時点で, この生成の操作を終了する方法である。しかし,Trepan における大きな問題点は, このような枝の生成におけるデータ利用や, そのルールの整理方法が, 定められた方法によらずに, その場で適切なものを選択する, いわゆる発見的 (heuristic) な方法に基づいていることである。

Neurorule は分解手法 (decomposition approach) により簡単化されたニューラルネットワークからルールを抽出する方法であり, 学習後のニューラルネットワークの結合の重みと, ユニットのバイアスに注目する [7]。まず, 最初に数値として与えられている入力データを離散的な値に変換する操作を適用する。通常, ニューラルネットワークのユニットの間の結合の数は極めて多いので, これらの中で重要でないものは切断して除去する。このような切断は, ニューラルネットワークの能力が極端に低下しない範囲で実施される。このような学習のあとで, 更に, 中間層のユニットの信号のレベルも離散的な値により変換しておく。このようにして, ニューラルネットワークの全部の信号のレベルが離散的な値により表現される。次に, 入力信号と中間層の間の関係をルールにより表現しておく。同時に, 中間層のユニットの信号レベルを用いて出力レベルとの関係もルールとして表現しておく。このような 2 つのグループのルール集合ができあがったら, 最後に, これらを調整して入力変数と出力レベルとの間の関係をルールとしてまとめる。

以上のように Neurorule は, 確かにニューラルネットワークの内部構造をできるだけ忠実に残しながらルールを抽出する方法ではあるが, その方法は複雑である。更に, この手法で抽出されるルールは, 命題が論理積で結合されて表現された形式でしかルールを表現できないので, これ以外の形式への拡張性がない問題がある。

2.3 GPによるルール生成におけるニューラルネットワークの役割

本論文では, このような中間的な方法により, ニューラルネットワークによる非線形関数の近似と, 言語的な解釈を同時に利用するシステムにおいて, より適切な言語的な表現を得るための方法を提案する。具体的には,GP の方法により, 言語的解釈の部分を最適化する方法であり, 恣意的に人間が言語的な解釈を与えるよりは, 判別能力が向上することが期待できる。

以下の議論では, Neurorule で示されたルール生成の方法を簡単化するために GP を用いる。具体的には, 入力信号を 2 進数に符合化し, その入力が意味のあるものであるかどうかを検定するまでは, Neurorule における処理と同じことを行なうが, 言語的ルールを生成する部分を GP により置き換える。これにより, 中間層における 2 進数化の操作や, この中間層データと入力との関係を計算する面倒な操作が不要となる。これを手順として記述すると, 次のようになる。

(1) 入力変数を 2 進化し, これらのビット信号を入力変数としている。
(2) 重みの小さな枝を切断する。
(3) この結果として入力として採用される命題が求められる。
(4) 最後に, これらの命題を用いて GP によりルールを生成する。

3 ニューラルネットワーク手法の概要

3.1 基本システムの構成

ニューラルネットワークを構成するニューロンの動作と, ニューラルネットワークの設計の関係について以下に整理しておく。階層型ニューラルネットワークはニューロンが 1 つの階層としてグループをなし, これらの階層が複数個存在し, 階層の間では前段の階層のニューロンの出力が次の段階の階層のニューロンの全部 (あるいは一部でもよい) の入力としてに結合されている。この結合の大きさは重みとよび, ニューラルネットワークを学習する過程で調整されていくパラメータである。

最初の階層への入力端子は一般には複数あり (入力端子), 最後の階層 (出力層とよぶ) のニューロンは 1 つであるケースも, 複数であるケースも存在する。出力層の出力がニューラルネットワーク全体の出力端子とよぶ。階層型ニューラルネットワークの構成を図 1 に示す。示している。図 1 では, ユニットへの入力信号 x_i に対して重み係数 w_i をかけたものを加え u_i とし, 更に, これに対してしきい値演算をほどこしている。しきい値演算とは, 加算された結果 u_i がある値 h_i より大きければ出力を出し, そうでない場合には出力をゼロとする演算である。このような演算結果は, 更に, 次の段階にあるユニットへの入力として用いられる。

ニューロンの内部のしきい値演算は, 入力合計がある一定の数値を越えるまでは出力はゼロであり, これ以上になると一定の数値を出力としてもつ性質をもつ。このような特性を示すものを, しきい値素子とよんでいる。現実には, このようなしきい値素子をそのまま用いたのでは学習によりニューラルネットワークのパラメータを最適化する場合に不都合であるので, 緩やかに変化する関数 (シグモイド (sigmoid) 関数) を用いる。

階層型ニューラルネットワークにおいては, 入力端子への入力に応じて出力端子に望ましい出力が得られるように設計することが目的となる。ニューラルネットワークは, 最初, このように調整されていないので, 複数の学習データの入力と出力 (外的基準) のペアを与え, ニューラルネットワークに入力を加えた場合の出力が外的基準に近くなるように, ニューラルネットワークの重みを調整する。

以上のようにして準備された学習データを用いてニューラルネットワークを調整する手順 (重みの最適化) は, 逆伝幡法 (back propagation argorithm) として整理されている [Ru-E]。

いま, 図 1 に示すように, ユニットの集合からなる層を N 層作り, $n-1$ 番目の層に含まれる第 j 番目ユニットと, n 番目の層に含まれる第 i 番目のユニットの間に, ある重みをもった回線により結合されているとする。この結合は第 $n-1$ 層から第 n 層へと一方向であり, 上に述べた場合

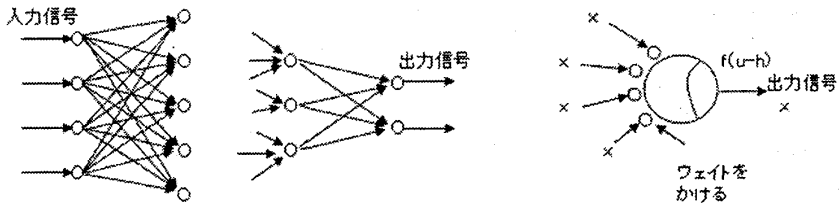

Figure 1: 階層型ニューラルネットワークの構成

に結合する回線の重みを $w_{i,j}^{n,n-1}$ とする。第 1 層はニューラルネットワークへ送られる入力信号を受け付ける入力端子であり, 最後の第 N 層はニューラルネットにより解析された出力信号を取り出す出力端子である。

いま, 第 $n-1$ 層の第 j 番目のユニットの出力信号を x_j^{n-1} とし, これが第 n 層の第 i 番目のユニットへの入力となっているので, 入力信号は以下のように記述される。

$$u_i^n = \sum w_{i,j}^{n,n-1} x_j^{n-1} \tag{1}$$

この入力信号の和はそのままユニット i の出力信号となるのではなく, 一定のしきい値 h_i を持った関数により制御されて出力される。第 n 層の第 i 番目のニューロンの出力値 x_i^n は次のように計算される。

$$x_i^n = f(u_i^n - h_i^n) \tag{2}$$

ここで, 関数 $f(.)$ はユニットの動作を規定する特性であり, 通常は数値計算上の収束などを考慮して次のシグモイド関数が用いられる。

$$f(x) = 1/]1 + \exp(-x)] \tag{3}$$

ニューラルネットワークにおける学習とは, ネットワークへ特定の入力 (第 1 層のユニットへの入力 x_i^1) を加えたときに, 対応して望ましい出力信号 (第 N 層の出力信号 x_i^N) が得られるように, ネットワークのパラメータ $w_{i,j}^{n,n-1}$ および h_i^n を調整していくことである。

出力層 N のユニット i から戻される学習信号は, 出力層からの出力値 x_i^N と教師信号 d_i^N との差を使って次のようになる。

$$\delta_i^N = (d_i^N - x_i^N) g(u_i^N), g(x) = f(x)(1 - f(x)) \tag{4}$$

第 n 層のユニット i から第 $n-1$ 層のユニットへ向かって戻される学習信号は

$$\delta_i^n = df(u_i^n)/du \sum_k \delta_k^{n+1} w_{k,i}^{n+1,n} \tag{5}$$

以上の量を用いて, 重みとしきい値との修正は, 計算の現在のステップと次のステップを, $t, t+1$ とすると, 次のようになる。

$$\Delta w_{i,j}^{n,n-1}(t) = \eta \delta_i^n x_j^{n-1} + \alpha \Delta w_{i,j}^{n,n-1}(t-1) \tag{6}$$

$$w_{i,j}^{n,n-1}(t+1) = w_{i,j}^{n,n-1}(t) + \Delta w_{i,j}^{n,n-1}(t) \tag{7}$$

$$\Delta h_i^n(t) = \eta \delta_i^n + \alpha \Delta h_i^n(t-1) \tag{8}$$

$$h_i^n(t+1) = h_i^n(t) + \Delta h_i^n(t) \tag{9}$$

ここで収束のためのパラメータ α, η は通常 1 以下であり, $\alpha = 0.5, \eta = 0.5$ 程度の値を用いる。

3.2 ニューラルネットワーク学習と離散化手法

ニューラルネットワークによる学習結果から言語的なルールを抽出する方法として, よく知られている Neurorule の方法の概要は, 次のようにまとめられる [Bae]。なお, ニューラルネットワークとしては3層構造の階層型ニューラルネットワークであり, 中間層が1つのものを用い, カテゴリごとに出力信号をユニットとして分離している。

ステップ1

ニューラルネットワークに対する入力と出力のペアを用いて, 通常の逆伝播法を用いて学習重みの最適化を行う。

ステップ2

ユニット間の重みを持つ結合 (枝) を適宜切断する。具体的には, 重みが小さい枝を無くする。しかし, その選択においては, もともとのニューラルネットワークの推定精度の極端な低下が起こらない範囲とする。

ステップ3

このように形成され, 簡単化されたニューラルネットワークに対して, 中間層の信号レベルを離散化する。

ステップ4

ネットワークの出力を, このような中間層の離散化された信号レベルを用いて記述する。この場合に, 得られる記述をルールとする。

ステップ5

更に, 中間層の離散化された信号レベルを, もともとの入力信号により記述する。なお, この入力信号も2進符号化されており, 離散化されていると仮定する。

ステップ6

以上の操作により得られるルールの2つを併合して,1つのルールを求める手順を繰り返す。これにより, 入力変数と出力のカテゴリとの間の対応関係を示すルールの集合を得ることができる。

なお, 以上の操作において入力信号は2進符号化されていることが仮定されている。例えば, 個人貸付の場合には, 貸し付け申請者の収入 W を $W > 3000, 2000 < W \leq 3000, 1000 < W \leq 2000, W \leq 1000$ のように4レベルに離散化し, これらの3ビットの符号化 (B_1, B_2, B_3) を与え, それぞれを $(1,1,1), (0,1,1), (0,0,1), (0,0,0)$ として表現しておく。

$W \leq 1000 \rightarrow (0,0,0)$

$1000 \leq W \leq 2000 \rightarrow (0,0,1)$

$2000 \leq W \leq 3000 \rightarrow (0,1,1)$

$W \geq 3000 \rightarrow (1,1,1)$

同様に, カテゴリカルデータについてもダミー変数を与えて, 離散化しておく。例えば, 企業の業種が A の場合 B の場合, これら以外を以下の2ビットを用いて表現する。

業種が A→ $(0,0)$

業種が B → $(0,1)$

業種がこれ以外 →(1,1)

従って, ニューラルネットワークへの入力は, もとの入力変数 W ではなく, このようなビット信号である。この信号の2進符号化が, どの程度最適であるかは議論の余地があるが, ここでは, 経験的に適切なレベルの境界値が与えられていると仮定する。

次のステップ2においては, ニューラルネットワークの入力層のニューロンから中間層のニューロンに向けてつながれている結合の, 数値の絶対値の小さいものから徐々に除去していく操作を適用する。この除去の過程において, もとのニューラルネットワーク結合を除去しないネットワークと比較して, 極端な予測精度の低下が起こる場合には結合を除去しない。また, すべての結合が除去された中間層のユニットは, ネットワークから取り除かれる。

ステップ3においては, 中間層のレベルの出力信号の値を, いくつかの段階レベルに離散化する。この過程で得られるカテゴリを, 発火レベル (activation level) とよんでいる。発火レベルの設定の仕方や, 近隣のどの数値を同じカテゴリにするかについては, さまざまな提案がある。クロ

ス度数表の検定に用いるカイ 2 乗検定の方法などが用いられる。このステップ 3 により, 出力信号すなわち, 学習データとして外生的に与えられたカテゴリと, 中間層の発火レベルとの関係を記述する関係が求まる。次のステップ 4 において, このルールが完成する。

次のステップ 5 においては, 中間層のレベルと発火レベルと, 入力データの関係を記述するルールを求める。ルールを求める対象は異なっているが, 方法はステップ 4 と同じである。

最後のステップ 6 では, ステップ 4 の結果であるルールと, ステップ 5 の結果であるルールを, 相互に比較して, 入力変数と出力レベルとの間のルールを検出する。

以上の説明からから分かるように, ニューラルネットワークからの言語的ルール抽出の代表的な方法で Neurorule は次のようなことが基本となっている [Bae]。

(1) 入力変数を 2 進化し, これらのビット信号を入力変数としている。
(2) 重みの小さな枝を切断する。
(3) 中間層の信号も離散化する。
(4) 最後に, これらの離散化された信号変数を用いてルールを記述する。

これらの基本操作の第 1 番目から第 3 番目は, 変数を離散化する前のニューラルネットワークの判別能力に比べて, 変数離散化を用いた場合のシステムの性能が極端に低下しないことに留意すれば, やや機械的な操作で実施することができる。最も面倒なのが, 第 4 番目のルールを生成する操作であろう。

3.3 GP によるルール生成

Neurorule においては, まず, 中間層の信号レベルを離散化し, できるだけルールの基礎となる部分を, この信号レベルで簡単化して記述し, そののちに, あらためて入力信号と中間層の信号レベルとの関係を記述している。また, この記述ルールを抽出する方法については,Quinlan らにより開発された従来の手法を拡張して, 多重に用いることを行っており, 極めて複雑な方法となっている。従って, この部分が自動化できれば, ニューラルネットワークからの言語的ルールの抽出も, 効率化されるであろう。

このような理由から, 本論文では, このルール抽出の部分を GP により実施するような拡張を行う。具体的には, Neurorule におけるステップ 4 からステップ 6 までの操作を省略し, 直接的に言語的ルールを GP により生成するものである。このような中間段階を省略することにより, Neurorule 適用における調整などの操作が不要となる。

ただし, GP はもともと 1 つの評価関数, 例えば与えられた数値の系列を近似する関数を求めるなどの最適化問題に適しており, ここでとり扱うような, 複数のルールを同時に生成することは基本的に含んでいない。従って, この部分を拡張する必要がある。この拡張については, GP における適合度の最大値をもとにした個体の選択において, 複数の個体を残すこと, 更に, 入力変数の個数を制限して, 複雑になるルールを除外することがある。この操作の概要を図 2 に示す。

例を用いて説明する。先ほど述べた個人貸付の場合に,$2000 < W \leq 3000$ などの 4 つのレベルに分割し, ビット列 (B_1, B_2, B_3) を割り当てた。新たに 3 つのビットが入力となる。これらのビットの 1 つだけについて, 中間層ユニットとの結合が無視できないと仮定する。例えば, これが B_2 である場合には, 命題として, $W > 1000$ のみが意味のあるものとして残る。

このように入力を離散化した結果を用いて, 意味のある結合だけを残していけば, 言語的ルールに用いることができる命題の集合を作成することができる。ただし, このような命題の抽出仮定では, あまり極端に命題数を制限すると, 離散化された入力を用いて結合係数の小さなリンクを切断したニューラルネットワークの判別能力が低下する。これを回避できるレベルの範囲で簡単化をはかる。

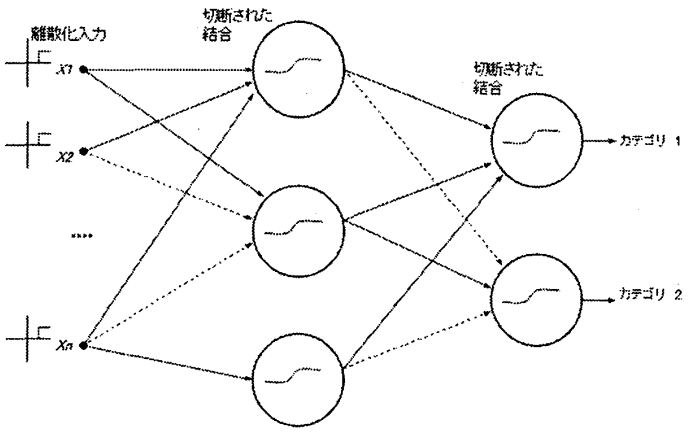

Figure 2: ニューラルネットワークによる入力離散化の概要

4 GP手法の基本とルール生成

4.1 GPによる関数近似

本論文で用いるGP手法については, すでに多くの記述があるので, 以下では要点のみを簡単にまとめておく [12]-[25]。なお, 説明を簡単にするため, 関数近似への応用例を最初に述べ, これをもとに論理式ルールを生成する方法へと拡張する。

GPはGA(Genetic Algorithm:遺伝的アルゴリズム) の1つの拡張であり, 個体はGAのようにビット列ではなく, 数学演算のための演算子, 変数からなっている。GPのシステムは基本的に, 3つの部分からなりたっており, その第1番目は個体の表現である。方程式は木構造で表現できるが, これを前置表現 (prefix representation) により置き換えておく。例えば, 方程式 $x(t) = [3x(t-1) - x(t-2)] \times [x(t-3) - 4]$ は, 次のように表現する。

$$\times - \times 3x(t-1)x(t-2) - x(t-3)4 \tag{10}$$

このような前置表現のそれぞれを個体とよぶ。

次に必要なのが, GPにより表現された個体の解釈である。これには式 (1) に示すような式により表現された方程式の右辺の形をもとにして, 関数の値を求めることである。個体により計算された関数の値 $\tilde{x}(t)$ と観測された時系列データ $x(t)$ との2乗誤差は近似度を与えるので, この逆数により個体の適合度を定義する。

第3番目に必要なのが, GPにより個体を性能の良いもの (今の場合には関数近似の能力の高いもの) に変換していく方法である。この方法の概要を図3に示す。

個体の集合 (プール) の能力をたかめることは, 個体に対して交差処理, 突然変異処理を行うことにより可能である。このため, *StackCount* というカウンタを用いる。

StackCount の値は, 前置表現で表現された個体のストリングを左側からサーチしていき演算記号に出会うとその数値を1つ増やし, 被演算子に出会うとその数値を1つ減らす操作を実施した結果である。個体のストリングの全体をサーチし終えたあとに, *StackCount* の数値は必ず1になる。従って, GPにおける初期個体を生成するときに, この条件を満足しないものは個体として採用しない。適合度に応じて2つの個体A, Bを選択したあと個体Aの交差位置を乱数により選択し, この位置における *StackCount* を計算しておく。次に, 個体Bについて同じ *StackCount* をもつ位置を検出し, 等確率で, ある1つの交差点を確定する。これらの位置を境界として, それぞれの個体の前半と後半を, 相互に交換した個体がoffspringとして生成される。

以上のような方法をアルゴリズムとしてまとめると, 次のようになる [1]-[10]。

(ステップ1)

乱数を用いて被演算子, 演算記号の並びからなる初期個体のプールを構成する。個体の表現の妥当性を, すでに述べた $StackCount$ を検査することにより行う。
(ステップ 2)
個体に表現された関数をもとに, それぞれの個体により得られる予測値を求める。これをもとにして, 個体 i における適合度 S_i を求める。
(ステップ 3)
次に示す適合度から変換された確率に応じて, 2 つの個体 i が選択される。

$$p_i = (S_i - S_{min}) / \sum_{i=1}^{N} (S_i - S_{min}) \tag{11}$$

ここで, S_{min} は適合度の最低値, N はプールの大きさである。この 2 つの個体に対して遺伝的操作を行い, 生成された新しい個体を次のステップにおける代替個体のプールである P-B に格納しておく。このような新しい個体の生成を, 規定回数繰り返す。新規個体の生成が終了したら, プール P-A の個体の中で, 相対的に適合度の低い個体を, プール P-B の個体により置き換える。
(ステップ 4)
ステップ 2 からステップ 4 までの交差処理を, 決められた個数の個体に適用し, 新しい個体のプールを作成したあとに, 次に示す突然変異を実施する。

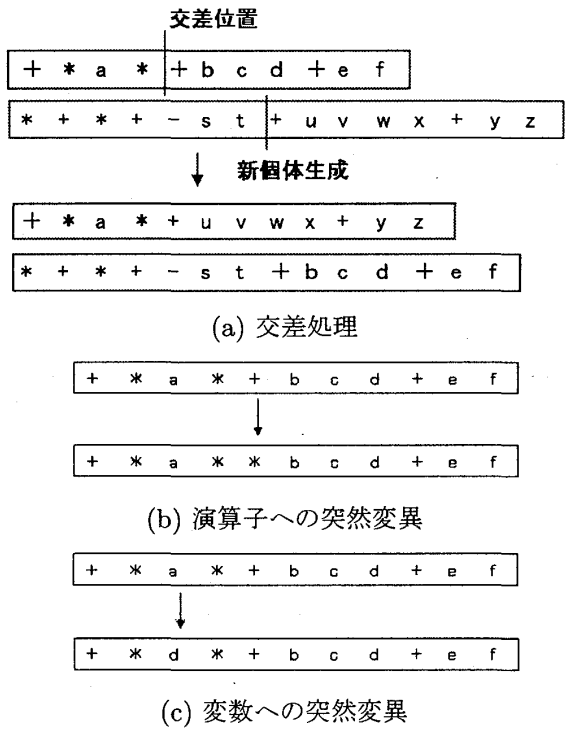

Figure 3: GP 手法における交差・突然変異処理の概要

G-突然変異:グローバルな突然変異を意味し,2 つの個体に対する交差処理である。ただし, 今の場合には, 個体 A は選択され突然変異を適用する個体であるが, 個体 B は, 初期個体の発生と同じ手順を用いて一時的に発生させた作業用の個体である。交差点を適切に決めたのちに, 個体 A の後半を個体 B の後半と交換する (個体 B に対しては, 特に何も操作や保存はしない)。
L-突然変異:任意に個体を選択して, この個体の被演算子, 演算記号の部分を, 任意に選択した被演

算子, 演算記号により置き換える。
(ステップ5)
ステップ2からステップ4までの操作を規定回数繰り返す。

4.2 GPによる論理式への遺伝的操作

これまでの研究においても,GP手法を論理式への遺伝的操作に適用し, プロダクションルールの改善をはかる方法論へ用いている。そのためには, 論理式の形式を2項演算の形で2つの命題を論理演算子で結合した場合に限定する必要があるが, 大きな制約ではない。従って, 基本的には論理式のレベルにおけるGPによる遺伝的操作は, 算術式における遺伝的操作と同様に実施できる。具体的には, 次のような置き換えを行う。
数値型入力変数 vi →　論理変数 Xi
算術演算子 $+, \times$ →　論理演算子 And,Or
ルールをGPにより学習・改善していく方法を用いる。これにはさまざまな方法が考えられるが, 以下では, 次のような比較的簡単な方法で, ルールを記述する論理式を表現する。
いま, 論理変数を含む論理式をルールであるとみなす。クレジット判別の予測として, 受け入れと拒否の分類を取り上げる。例を次に示す。
if x1 > a1 and x2 < a2 or x3>a3 and x1<a1 then reject
論理演算子で結合 (GPにおける前置表現ではすべて2項演算に分解されている) した論理式に書き換えられる。
X_i is True if $x_i \geq a_i$ otherwise False
それぞれの個体は, このような論理変数を含んだ論理式をクレジット判別のルールとして表現している。論理式表現における個体の適合度は, 次のようにして計算できる。
(1) 論理値の計算
それぞれのクレジット案件について論理変数 X_i の値を求める。
(2) 論理式の解釈
命題は1つの論理値として与えられるので, 論理演算子を考慮しながら, これを含む論理式の値を計算する。
(3) 適合度の計算
学習に用いているクレジット案件のデータに対して, 上のような論理式を用いたクレジット判別の予測をする。この予測結果と, 観測されたクレジット案件のデータ (実際に受け付けたか拒否したかどうか) とを比較する。予測と実際に発生したデータが同じなら, この個体の適合度を増加させる。

4.3 クラシファイヤシステムの有効性

本論文で用いるGPによるクレジット判別システムの特徴は, 相対的に大きな適合度をもつ個体に対しても近似を行うことができる候補として残し, 複数の論理式 (ルール) を個体プールの形で格納しておくことにある。これは, 変化する環境への適合を考慮したプロダクションシスGAにより構成する, いわゆるクラシファイヤシステム (classifier system) と同様の考え方を用いていることになる。本来の識別対象のデータの識別ルールが単独であるケースはまれであり, これからの変形や, ランダム性により生じるばらつきによる識別誤差をクラシファイヤシステムは軽減する効果がある [9][10]。
通常のGP手法では, 個体の適合度を最大化する方向に学習が行われるが, 本論文での適用では, 出現するパターンを可能なかぎりカバーするように, 多様な個体を準備することが目的とされる。具体的には, クレジットの可否を決定する場合に学習により生成されたルールの中で相対的に多数をしめるルールの決定に従うこととする。これにより個体, すなわちルール生成過程を近似する論理式が, 特定の学習データだけに適合することを避けることができ, しかも, 適合度も相対的に大きいものをプールに残すことができる。

5 応用例

5.1 企業倒産予測への応用

本論文の手法の有効性を確認するため, 最初に, クレジット審査の極端な事例として, 企業の倒産予測への応用を示す。企業倒産を予測する確率的な手法に関しては, これまでの多数提案されており, ロジットモデル (logit model), 多変量判別分析 (MDA:Multivariate Discriminant Analysis)) や, ニューラルネットワーク (NN) による手法などがある。これらの中でも,MDA やこれと類似した手法は, 最もよく知られた方法であり, 企業の経営状況を反映する財務指標を入力とする方法である。

以下では応用例として, 最初に, 本章で述べているニューラルネットワークと GP 手法に基づいたルール抽出の方法を用いて, 日本の企業における倒産予測の要因を説明するルール導出をとりあげる。これに関しては, すでに, もともとのニューラルネットワーク (財務指標を数値変数として用いる方法) による推定と, その比較分析の結果を公表しているが, ここでは, やや簡単化された例をとりあげる [11]。

このためには, 戦後日本の企業倒産の事例を検索し, この倒産時期における財務指標を収集するなどの作業が必要となる。具体的には, ある時期において債務不履行となった企業に注目しながら進めるが,(1) 観測期間は 1970 年から 1986 年までとする, (2) 小規模な企業を除外して考えいづれかの証券市場に上場されていること,(3) 製造業に属する企業であること, などを条件として検索する。

このような倒産企業だけではなく, この時期に健全 (非倒産) であった企業に関するデータも収集しておく必要がある。このようなデータをペアサンプル (pair sample) とよぶ。この手法は通常の MDA における操作と同様である。このように選択された企業は, 最終的には 26 の倒産企業と,26 の健全企業としてまとめられる。この場合, 健全企業は倒産企業のペアサンプルをなす必要があるので, 倒産企業が存続した時期に存続しており, 同じ産業分類に属して, 企業規模も類似しているような企業を選択している。これらの 26 企業の中で,15 企業がシステムを構成するための学習に利用され, 残りの 12 企業を用いてシステムの性能を検証している。

次に, これらの企業の財務データを収集する必要がある。最初に, 企業の決算データから計算される 62 個の財務指標を収集する。しかし, これらの全てを用いるわけではない。財務指標の分布が極めて偏っているものは, 統計的な倒産予測には適さないので, 入力データとして用いない。具体的には,t 検定の方法を用いて, 倒産企業と非倒産企業の 2 つのグループ間において平均値の差がない財務指標は除去し, 更に分布が正規分布から大きくはずれている財務指標を,Wilcoxson ランク検定により見出し, 入力データから除外している。

このような検定の結果として,13 個の財務指標を選択した。しかし, この 13 個という数値は倒産予測を言語的にルールで記述するには少し多いので, 更に数を減らす工夫をする。具体的には, ニューラルネットワークにおいて結合の重みの小さい部分を切断し, 入力数を減らす操作をする。最初に, それぞれの財務指標に対して 4 つのしきい値を設定する。これらの離散化された入力に対して, ニューラルネットワークによる倒産予測システムを構成する。更に, このニューラルネットワークの結合重みの切断を行う操作を適用して, 予測の能力を極端に低下させない範囲で, 簡単化されたネットワークを求める。

このような操作を実施した結果として, もともとは 13 個の指標を計算しているが, 最終的に次に示す 5 つの指標にまで縮約している。
判別に用いられる入力変数は次のように定義される。x1:使用総資本事業利益率 x2:固定資産回転率
x3:売上高経常利益率 x4:流動比率 x5:負債比率

これにより, ルール生成の準備が完了する。シミュレーションのための条件は以下のようにしておく。

データの利用:2/3 を学習データ,1/3 を検証用のデータとして利用

ニューラルネットワークの階層数:3
中間層におけるユニット数:10
出力層におけるユニット数:2
個体記述の配列の最大サイズ (これを M_s で表す)=10
個体の数プールの大きさ:1000
GP 実行の世代数:100

表 1 には適合度が最大になる 4 つの固体により得られる判別の結果について, 平均値を示している。これらの判別に用いる固体は, C4.5 などと異なり, さまざまな形状の木構造を示している。これらの木構造は, 個体の中で最大の適合度をもつものを選択することにより求めることができる。

表 2 には, これらの 4 つの個体の内容を前置表現のまま示している。なお論理変数 X_i は入力変数 x_i を離散化する場合の式として記述されていると仮定する。これらの 4 つの個体により, 判別の全部の内容を記述することができる。これらの判別結果は, 個体を記述する配列の最大の大きさを M_s=10 としているが, これをこれを拡大すると更に大きな判別率を得ることができる。しかし, 一方では, 個体の示す木構造はより複雑になる問題がある。

表 1　倒産予測の結果 (判別率の平均値を示す%)

カテゴリ	GP 手法	NN	MDA
failed	86%	96%	82%
sound	88%	90%	78%

表 2 生成されたルールの例

Or And X3 X1 Or X5 X4
Or X4 Or Or X1 X4 And X5 X4
Or X1 Or X4 Or X2 X4
Or X5 Or X4 And X3 X4

5.2 German Creit を用いた性能の相互比較

本論文で述べているルール抽出手法の有効性を確認するために, テスト用のデータとして公開されているデータに対する解析と, これを Neurorule を用いて実施した場合のルール構成手法との比較を行ってみる [7]。このデータはドイツの消費者ローン会社で実施された 1000 名を対象にした貸付審査の結果データであり, 貸付を拒否された 300 名のデータと, 貸付された 700 名のデータからなる。貸付を希望する個人のデータを入力して, これらの審査結果の違いを出力するような説明ルールを構成することが課題である。

データの項目は,7 つの数値データと,13 個のカテゴリカルデータとからなる。ニューラルネットワークへの入力として, これらの変数項目を利用する前に, これらを離散化しておく。その結果,45 個の離散的な入力変数が構成される。次に, これを用いて学習したニューラルネットワークにおいて, 重要でないユニット間結合を切断し, これにともなって必要ではなくなった入力をシステムから除去する。この結果,45 個の離散化入力から 39 個が除去され, 結果的にはシステムには 6 個の離散化データだけが入力変数として残ることが分かる。判別に用いられる入力変数は次のように定義される。
x1:手形口座開設の内容 (4 カテゴリ) x2:契約継続月数 (数値)
x3:クレジット履歴 (5 カテゴリ) x4:借入目的 (11 カテゴリ)
x5:預金口座内容 (5 カテゴリ)　x6:保証人の有無 (3 カテゴリ)

シミュレーションのための条件は, 以下のようにしておく。
ニューラルネットワークの階層数:3
中間層におけるユニット数:55

出力層におけるユニット数:2
個体記述の配列の最大サイズ:$M_s = 10$, および $M_s = 20$
個体の数プールの大きさ:1000
GP 実行の世代数:100

Baesen らにより実施された研究では,Neurorule によるルール生成と, これ以外の代表的な手法との比較が示されている [Bae]。この概要を表 3 に示している。なお, 表 3 において記号 GP-1, GP-2 は, ぞれぞれ,$M_s = 10$, および $M_s = 20$ の場合に対応している。この結果から分かるように,Neurorule は他のルール抽出方法より有効であることが理解できる。しかし, 以下のシミュレーションで示すように, 本章でわれわれが提案している方法は, この Neurorule の結果より少しではあるが, 改善されることが示される。

表 4 には適合度が最大になる 4 つの個体により得られる判別の結果について, 平均値を示している。この表において,P-NN はニューラルネットワークの構成において必要ない結合を切断した場合の結果を示している。これらの判別に用いる個体は, C4.5 などと異なり, さまざまな形状の木構造を示している。これらの木構造は, 個体の中で最大の適合度をもつものを選択することにより求めることができる。

表 5 には, これらの 4 つの個体の内容を前置表現のまま示している。なお論理変数 Xi は入力変数 xi を離散化する場合の式として記述されていると仮定する。これらの 4 つの個体により, 判別の全部の内容を記述することができる。これらの判別結果は, 個体を記述する配列の最大の大きさを M_s=10 としているが, これをこれを拡大すると更に大きな判別率を得ることができる。しかし, 一方では, 個体の示す木構造はより複雑になる問題がある。

表 3 判別結果の比較 (判別率の平均値%)

C4.5	P-NN	Neurorule	Trepan	Nefclass	GP-1	GP-2
74.25	77.84	77.25	73.95	73.65	77.23	79.98

表 4 生成されたルールの複雑さの比較

手法	複雑さ
C4.5	17 個の命題形式のルール
P-NN	6 個の入力変数を持つシステム
Neurorule	4 個の命題形式のルール
Trepan	11 個の葉と 21 個の節からなる木構造
Nefclass	14 個のファジイルール
GP-1	4 個の命題形式のルール
GP-2	10 個の命題形式のルール

表 5 生成されたルール

Or And X4 X1 Or X6 X4
Or X5 Or Or X1 X4 And X6 X4
Or X1 Or X4 Or X3 X5
Or X6 Or X5 And X4 X2

5.3 格付け問題への応用

第 3 番目の応用例として, 現在の日本企業の格付け問題をとりあげる [To-5]。本来なら, 金融機関のクレジット実績を入手し, 解析する必要があるが, 公表されたものがないことや, 以下で述べる格付データは, 実質的にクレジット審査に対応すると考えれるので, 格付に関する公表データを用いて, 本システムの性能を調べている。

企業の発行する債券に対して, ランクに相当する格付 (Bond Rating) を公表することは1990年代から実施されている。毎年新規に発行される債券に関して, Moody's や Standard & Poor などの格付機関が, 投資家が投資対象とするに値するかどうかを判断する材料としてデータ提供を行っている。この格付を用いることにより, 投資家は投資した資金を回収できるか, あるいは損失をこうむるかの予測をすることができる。格付は,Aa, A, Ba, B などの表現で, 数段階のランクで示される。以下では本章で述べているニューラルネットワークと GP 手法に基づいたルール抽出の方法を用いて, 日本の企業における格付を説明するルール導出をとりあげる。

最初に, 日本企業の中から 3 つのグループ, 具体的には 3 つの格付のグループを構成する。このための格付情報を,Moody's Japan が 2000 年に公表したデータを用いる。この Moody's Japan による格付は,AAA から CCC までの広い数段階の格付として公表されているが実際には極めて低い格付である C や CCC などは現れない。これは, 日本企業が, 少なくとも格付の対象となる企業の中には, 極端に経営状況が悪い企業は存在しないことを意味している。

更に, 本論文での分析手法の性質から, あまり多いランクを判別するのは適切ではないので, 以下のように複数の格付のランクを 1 つにまとめ, 企業を選択しておく。

(1) ランク A1,A2,A3,Aa1,Aa2 など 30 社をカテゴリ A としてまとめる
(2) ランク Baa1,Baa2,Baa3 など 32 社をカテゴリ B としてまとめる
(3) ランク Ba2,B2,B1,B2,Caa1 など 27 社をカテゴリ C としてまとめる

このように選択された合計 89 社の企業データの中から, その 2/3 をシステムを構成するための学習に用い, 残りの 1/3 をシステムの性能を評価する検証用に用いる。すなわち, 2000 年度における 89 社の決算データ (24 財務指標へ集約) と格付データを用いて, 本章の手法による格付の性能評価を行なう。すなわち, 外的基準として格付の値 (A, B, C の 3 つに集約) を与え, これに一致するように GP を用いて学習をすすめ, 最後に同じ学習データに対して求められた最も適合度の高いルールを適用する。その結果を外的基準と比較して, 一致度を求める。

なお, 用いる財務指標に関しては, 当初の 24 指標は簡潔なルールを生成するには多過ぎるので, ニューラルネットワークを用いた入力変数の縮約を行なう。ニューラルネットワークへの入力として, これらの変数項目を利用する前に, これらを離散化しておく。その結果,48 個の離散的な入力変数が構成される。次に, これを用いて学習したニューラルネットワークにおいて, 重要でないユニット間結合を切断し, これにともなって必要ではなくなった入力をシステムから除去する。この結果,48 個の離散化入力から 39 個が除去され, 結果的にはシステムには 9 個の離散化データだけが入力変数として残ることが分かる。次に示す主要な 9 指標だけを用いるように変更している。判別に用いられる入力変数は次のように定義される。

x1:株主資本営業利益率 x2:株主資本利払後事業利益率 x3:固定資産回転率 x4:使用総資本利益率 x5:増収率 (5 年間平均) x6:株主資本成長率 x7:営業キャッシュフロー対負債比率 x8:営業キャッシュフロー対流動負債比率 x9:株主資本利益率

生成する言語的ルールは, これらの 3 つのグループごとに作成する。例えば, グループ A の場合には, これに属するためのルールを生成し, これにより真となるものは格付 A と判断し, これ以外のものは B あるいは C とする。しかし, これが, B あるいは C であるかは, 同様に作成する B および C グループの判別ルールにより結果を用いる。

シミュレーションのための条件は以下のようにしておく。

ニューラルネットワークの階層数:3
中間層におけるユニット数:10
出力層におけるユニット数:2
個体記述の配列の最大サイズ (これを M_s で表す)=10
個体の数プールの大きさ:1000
GP 実行の世代数:100 表 1 には適合度が最大になる 4 つの固体により得られる判別の結果について, 平均値を示している。これらの判別に用いる固体は, C4.5 などと異なり, さまざまな形状の木構造を示している。これらの木構造は, 個体の中で最大の適合度をもつものを選択することにより求めることができる。

表 6 には GP により生成された言語的ルールによる倒産予測の結果を示している。比較のため,

同様に解析をニューラルネットワーク (NN) と多変量解析のパッケージを用いて実施し, 結果を表7.6にまとめている。ニューラルネットワークによる分析では, 格付けランクのA,B,Cに応じて, 出力の値を5/3, 1/2, 1/6として与え, 学習のための外的基準学習を行ない, 同じデータを用いて判別する。この場合, ネットワークの出力 y が, それぞれ, $2/3 < y \leq 1, 1/3 \leq y < 2/3, 0.0 \leq y < 1/3$ の範囲にある場合をA, B, Cとして判別している。

表6 債券格付判別の結果比較 (判別率の平均値%)

category	GP method	NN	MDA
A	83.33%	86.66%	76.66%
B	81.18%	84.84%	72.72%
C	81.48%	85.18%	70.37%

表7 生成されたルール

A	Or X8 Or Or X2 Or X6 X2 X3
	Or Or X3 Or X2 Or And X5 Or X9 X2 X6 X1
	Or Or X5 X2 Or Or X7 And X4 X9
	Or Or Or X4 X6 Or X6 X2 And And X9 X5 X3
B	Or Or X5 X2 Or Or X7 And X4 X9
	Or X8 Or Or X2 Or X6 X2 X3
	Or Or Or X4 X6 Or X6 X2 And And X9 X5 X3
	Or Or X3 Or X2 Or And X5 Or X9 X2 X6 X1
C	Or Or X6 And And X8 X6 X5 Or X7
	Or X8 Or Or X2 Or X6 X2 X3
	Or Or X5 X2 Or Or X7 And X4 X9 X7
	Or Or X1 Or X5 X4 Or X4 X7

表7には, これらの4つの個体の内容を前置表現のまま示している。なお論理変数 X_i は入力変数 x_i を離散化する場合の式として記述されていると仮定する。これらの4つの個体により, 判別の全部の内容を記述することができる。これらの判別結果は, 個体を記述する配列の最大の大きさを M_s=10としているが, これをこれを拡大すると更に大きな判別率を得ることができる。しかし, 一方では, 個体の示す木構造はより複雑になる問題がある。

これより分かるように, GPにより生成された言語的ルールによる判別結果は, 多変量解析による判別結果と, ほほ, 同等であることが分かる。数値としては, 判別分析の結果の方が良好であるが, 言語的なルールである利点はこれを上回る。

6 むすび

本論文では, ニューラルネットワークとGPによるクレジット判別ルール生成システムの構成について述べ, 公開されているデータを用いて検証を行い良好な結果であることを示した。ニューラルネットワークによる入力の離散化と,GPによるルール自動生成の性能を結合することにより手法を簡素化した。応用例として, 代表的なクレジット審査データを用いた検証などへと適用した。

今後, 企業の製品開発や立地分析などクレジット審査以外の分野に拡張するように試みる予定である。

References

[1] E.I.Altman, *Corporate bankruptcy in America*, D.C.Health and Company, 1971.

[2] B.Lev,*Financial Stament Analysis*, Prentice-Hall, 1974.

[3] Y.H.Pao,*Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Co.,Inc., 1989.

[4] C.M.Bishop,*Neural Networks for Pattern Recognition*, Oxford University Press, U.K., 1995.

[5] J.R.Quinlan,*C4.5 Programs for Machine Learning*,Morgan Kafumann, Chambery, France,1993.

[6] R.Andrews,J.Diederich and A.B.Tickle, "A survey and critique of techniques for extracting rules from trained neural networks, "Knowledge Based Systems, vol.8,no.6,pp.373-389,1995.

[7] B.Baesens,R.Setiono,C.Mues and J.Vanthienen, "Using neural network rule extraction and decision tables for credit-risk evaluation," Management Science, vol.49,no.3,pp.313-329,2003.

[8] M.W.Craven,J.W.Shavlik, "Extracting tree-structured representations of trained networks, D.Touretzky, M.Mozer,M.Hasselmo,ed. *Advances in Neural Information Processing Systems (NIPS)*,vol.8,pp.24-30,MIT Press, Cambridge,MA.

[9] N.Nauck , "Data analysis with neuro-fuzzy methods, Habilitation Thesis, University of Magdeburg, 2000.

[10] S.Dutta and S.Shekhar, "Bond rating:A non-conservative application of neural networks," discussion paper, Computer Science Division, University of California, Berkeley ,1989.

[11] 李 鋼浩, 時永祥三, "ニューラルネットワークによる経営情報解析-倒産分析と時系列解析," 経営情報学会論文誌,vol.1,no.2,pp.32-43,1991.

[12] Y.Ikeda and S.Tokinaga, "Approximation of chaotic dynamics by using smaller number of data based upon the genetic programming, IEICE Trans.Fundamentals, vol.E83-A,no.8, pp.1599-1607,2000.

[13] Y.Ikeda and S.Tokinaga, "Controlling the chaotic dynamics by using approximated system equations obtained by the Genetic Programming," IEICE Trans.Fundamentals,vol.E84-A,no.9, pp.2118-2127, 2001.

[14] 池田欽一, "共進化によるカオス常微分システムの推定,"電子情報通信学会論文誌,vol.J85-A, no.5,pp.548-559,2002.

[15] 矢加部正幸, 時永祥三, "遺伝的プログラミングを用いた CNN による拡散モデルの近似とその同期化への応用", 電子情報通信学会論文誌,vol.J85-A, no.5,pp.548-559,2002.

[16] X.Chen and S.Tokinaga,"Approximation of chaotic dynamics for input pricing at service facilities based on the GP and the control of chaos ,"IEICE Trans.Fundamentals,vol.E85-A, no.9,pp.2107-2117,2002.

[17] 池田 欽一, 時永祥三,"共進化遺伝的プログラミングによる社会学習を考慮したマルチエージェントシステムの構成とその人工株式市場分析への応用, 経営情報学会論文誌, vol.13,no.3,pp.17-35,2003.

[18] Y.Ikeda and S.Tokinaga, "Chaoticity and fractality analysis of an artificial stock market by the multi-agent systems based on the co-evolutionary Genetic Programming", IEICE Trans. Fundamentals, vol.E87-A,no.9,pp.2387-2394, 2004.

[19] 呂 建軍, 時永祥三, " 遺伝的プログラミングによる時系列モデルの集合的近似とクラスタリングへの応用", 電子情報通信学会論文誌, vol.J88-A,no.7,2005.

[20] 呂 建軍, 時永祥三, " 遺伝的プログラミングによる時系列セグメント識別を用いたカテゴリ記号表現に基づく 2 階層認識手法とその予測への応用,", 電子情報通信学会論文誌, vol.J88-A,no.11,2005.

[21] J.R.Koza: *Genetic Programming*, MIT Press, 1992 .

[22] J.R.Koza, " Genetic programming:A paradigm for genetically breeding populations of computer programs to solve problems", Report No.STAN-CS-90-1314, Dept.of.Computer Science Stanford University, 1990.

[23] J.Koza, " Evalution and subsumption using genetic programming," Proc of the First European Conference on Artificial Life, MIT Press,1991.

[24] J.R.Koza, " Genetic programming:A paradigm for genetically breeding populations of computer programs to solve problems," Report No.STAN-CS-90-1314, Dept.of.Computer Science Stanford University, 1990.

[25] M.J.Keith and M.C.Martin, " Genetic programming in C++: Implementation issues," in (ed) K.E.Kinnerar,Jr.,*Advance in Genetic Programming*, MIT Press, 1994.

[26] E.D.Rumelhart,G.E.Hinton and R.J.Williams, " Learning internal representation by error propagation, Paralell distributed processing", McClelland and the PDP Research Group ed. ,MIT Press ,1989.

時永　祥三　　(九州大学大学院・経済学研究院・経済工学部門・教授)
岸川　善紀　　(秋田県立大学・システム科学技術学部・経営システム工学科・助手)