

Kernel-based手法を用いた判別分析の高性能化とそのクレジット審査への応用

儲, 梅芬
九州大学大学院経済学府

<https://doi.org/10.15017/3000408>

出版情報 : 経済論究. 126/127, pp.79-89, 2007-03. 九州大学大学院経済学会
バージョン :
権利関係 :

Kernel-based手法を用いた判別分析の高性能化 とそのクレジット審査への応用

儲 梅 芬

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なく、説明性の確保などからも必要とされている。本論文では、Kernel-based手法を用いた判別分析の高性能化とそのクレジット審査への応用を示す。

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が数多く提案されている[1]-[3]。また一般的な非線形関数を実現する目的から、ファジィ推論システムの応用も試みられている[4][5]。このような判別関数による関数的なクラスタ推定とならんで、2分岐に基づく手法も展開され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ところみがなされている[6]。また、数値計算の手法とならんで、最近、ニューラルネットワーク手法をベースとして判別結果を言語的な形式で出力し、審査や勧告に利用する方法も開発されている[7]-[13]。

言語的な判別結果の表示を重視する場合には、数値的な分析手法より、一般に性能が低下することが示されている。このためクレジット審査に応用する場合には、注意が必要となる。同時に、与えられたサンプルに関する数値的な判別分析は、どこまで性能を向上させることができるかを見出すことは、すべての判別問題に共通するテーマでもある。

本論文では、判別分析の手法として、Kernel-based統計による方法を用い、入力変数に対する非線形変換関数を適用し、やや高次元の線形判別関数を用いた問題へと帰着させ、分析の精度を向上させる[14]-[16]。高次元の線形判別問題を解く場合に、このような非線形変換を直接取り扱う必要がないため、数値計算手法が単純化されている。得られたデータに対する線形判別関数の推定問題は、従来から存在する定式化手法により、最大固有値問題へと帰着される。このKernel-based手法は先行研究が存在する既存の方法であるが、これまでクレジット審査など経済分野への応用は示されておらず、適用可能性は未知であったと言える。また、他の判別分析手法との比較分析についても、統計パッケージなど一般的な手法にとどまっており、多くの判別分析のなかでも相対的に性能の良好な分析結果との比較も行われていない。このような意味で、あらためてクレジット審査の分野にKernel-based手法を適用し分析することは意義があると言えよう。応用例として、人工的なデータへの応用とクレジット審査問題への適用を示す。この結果、従来手法である多変量判別分析やニューラルネットワーク手

法に比べて、極めて性能が高く、高度化が達成できることが分かる。

以下では、2.においてはKernel-based手法による判別関数の推定の原理について述べる。3.では人工的データによる検証を示し、4.ではクレジット審査への応用を論じる。5.企業の格付問題と判別分析について論じる

2 Kernel-based手法による判別関数の推定

2.1 クレジット審査のモデル化

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ず、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]。格付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][17][18]。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、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[2]-[5]。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[6]-[13]。また、実際的な指導や勧告に判別結果を用いる目的で、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。

数値的な判別分析手法の中で、比較的性能が良好であるとされている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[3]。しかしながら、ニューラルネットワーク手法における非線形性には限界があり、関数の次数制限により分割される領域の形状が決定される。一方、次数の高い判別関数を効率よく計算する方法が見出せれば、このような非線形関数の形状による制限が緩和される見通しがある。本論文で検討を加えるKernel-based手法は、このような関数の次数を極端に増加させることにより、判別分析における領域の形状に自由度をもたせる方法であり、さまざまな理論的検討と応用が示されている。

多変量解析の1つの手法として判別分析は多くの分野で用いられており、現在でも金融業においては、企業への貸付や消費者ローンの審査において大きな位置を占めている。判別分析の基本は、観測されたサンプルごとに判別の基礎となるデータを入力変数 x_i (判別変数) として与えておき、同時に、このサンプルが所属する分類（これをクラスと呼んでおく）が与えられている場合に、クラスを推定

するための x_i に関する線形の関数を求めることである。このように準備されるデータを学習データと呼び、複数のクラスについて同時にデータを収集するので、ペアサンプルとも呼ばれる。

線形の判別関数を求める方法は、クラス内の分散とクラス間の分散の比を最小化するように、パラメータである線形判別関数の係数を求める問題に帰着される。しかしながら線形判別関数によるクラス分けの方法では、関数が線形であることに制約があるため、判別関数を非線形にまで拡張することで、より精度の高い判別を可能とする方法が提案されている。ニューラルネットワークは、このような問題を解決する方法の1つである。ただし、ニューラルネットワークを含めて、非線形判別関数を求める方法では、関数推定の方法論も非線形問題となるため、計算時間が多大となる問題がある。

このような課題を解決する1つの方法として提案された、代数的に非線形判別関数を求める方法がKernel-based手法である[14]-[16]。Kernel-based手法の特徴は、写像により判別変数の次元を増加させ、より精度の高い判別関数を構成することと、この次元の増加による計算量が抑制できることにある。

2.2 Kernel-based手法による判別関数の推定

Kernel-based手法とは、入力変数に対する非線形変換関数を適用し、やや高次元の線形判別関数を用いた問題へと帰着させる方法である。この計算の過程で、dot productという表現を用いる。2次元ベクトル (x_1, x_2) を3次元 $(x_1^2, \sqrt{2}x_1x_2, x_2^2)$ という2次モーメントへ変換する変換がある。このとき、2つのベクトル $x=(x_1^2, \sqrt{2}x_1x_2, x_2^2)$, $y=(y_1^2, \sqrt{2}y_1y_2, y_2^2)$ の要素の、すべての組み合わせの積和 $\sum_{i_1, i_2, j_1, j_2=1}^2 x_{i_1}x_{i_2}y_{j_1}y_{j_2}$ を計算することをdot productとよぶ。これを $(x \cdot y)^2 = k(x, y)$ として表現する。また、 $k(x, y) = \exp(-\|x - y\|/\sigma)$ の形の計算をする場合にはガウシアンKernelとよぶ。

Kernel-based手法においては判別分析の問題は、変数ベクトルの M 個のサンプル x_k , $k=1, 2, \dots, M$ から共分散行列 C を計算し、これを用いて固有値問題を解くことに帰着される。いま、判別すべきクラスの数 N とし、それぞれのクラス l には n_l 個のサンプルがあり、この合計を $M = \sum_{l=1}^N n_l$ とする。 j 番目のサンプルを表す変数ベクトルを x_j とする。次のような共分散行列を定義する。

$$C = M^{-1} \sum_{j=1}^M x_j x_j^T \quad (1)$$

いま、変換関数 $\phi(x)$ を用いて入力変数 x を変換する。この場合、変換されたデータ $\phi(x)$ の次元 n は、もとの変数の次元 m より多く設定されており(いわゆる高次元)、これにより、より自由度の高い判別関数の構成が可能となる。

$$\phi: R^m \rightarrow F, \quad x \rightarrow \phi(x) \quad (2)$$

変換されたデータについての共分散行列は、次のようになる。

$$V = M^{-1} \sum_{j=1}^M \phi(x_j) \cdot \phi(x_j)^T \quad (3)$$

クラス l のサンプルの入力変数ベクトルの第 k 番目の要素を x_{lk} としておく。この平均値 $\bar{\phi}_l$ を次のように定義し、この平均値に関する共分散行列 B を定義する。

$$\bar{\phi}_l = n_l^{-1} \sum_{k=1}^{n_l} \phi(x_{lk}) \quad (4)$$

$$B = M^{-1} \sum_{l=1}^N n_l \bar{\phi}_l \bar{\phi}_l^T \quad (5)$$

変換されたデータについての共分散行列 V は、また次のように書くこともできる。

$$V = M^{-1} \sum_{l=1}^N \sum_{k=1}^{n_l} \phi(x_{lk}) \cdot \phi(x_{lk})^T \quad (6)$$

線形判別関数を構成することにより、いくつかのクラスに分類する問題は、いわゆる Kernel Fisher Discriminant Analysis (KFD) と呼ばれるものであり、複数のクラスの平均値ができるだけ乖離しており、それぞれのクラスの分散は小さくなりように構成する。この問題は、次に示す Rayleigh coefficient を最大化することに帰着される [14]-[16]。計算のために、以下のようなデータを準備する。

2つのクラス p, q に関して、次のような Kernel 関数を定義する。

$$(k_{ij})_{pq} = \phi(x_{pi})^T \cdot \phi(x_{qj}) \quad (7)$$

これらを用いて、次の $M \times M$ 行列 K を定義する。

$$K = (K_{pq}), \quad p, q = 1 \sim N, \quad K_{pq} = (k_{ij}), \quad i = 1 \sim n_p, \quad j = 1 \sim n_q \quad (8)$$

K_{pq} は $n_p \times n_q$ 行列である。

次に、定数の要素を持つ行列 W (M times M) を定義する。行列 W は、その部分行列 W_l , $l = 1, 2, \dots, N$ がクラス l のサンプル数 n_l の逆数になるブロック対角行列である。

$$W = (W_l), \quad l = 1, 2, \dots, N, \quad W_l \text{ は } n_l \times n_l \text{ 行列である} \quad (9)$$

判別分析の問題は、以下の形式の固有値問題に帰着されることが示される。

$$\lambda V \nu = B \nu \quad (10)$$

すなわち、この式を固有値の最大化問題として見ると以下のようなになる。

$$\lambda = \frac{\nu^T B \nu}{\nu^T V \nu} \quad (11)$$

従って、KFD を解く問題と固有値問題が等価であることが示される。

ここで固有ベクトルは、空間 F の要素の線形結合となることを用いて、次の表現を得る。

$$\nu = \sum_{f=1}^N \sum_{g=1}^{n_f} \alpha_{fg} \phi(x_{fg}) \quad (12)$$

f はクラスを指す添え字であり、 g はクラス内のサンプルを示す添え字である。ベクトル α を、次のように定義しておく。

$$\alpha = (\alpha_{11}, \alpha_{12}, \dots, \alpha_{ij}, \dots, \alpha_{Nn_N}) \quad (13)$$

また、クラス l ごとに要素が $1/n_l$ に等しい部分行列 W_l からなる $M \times M$ ブロック対角行列 $W = (W_l), i = 1 \sim N$ を導入したことにより、式(11)の問題は次のようになる。

$$\lambda = \frac{\alpha^T K W K \alpha}{\alpha^T K K \alpha} \quad (14)$$

$$\nu^T \nu = \alpha^T K \alpha = 1 \quad (15)$$

この表現を更に簡単にするために、行列の固有値分解を行う。

$$K = U U^T, \quad U U^T = I \quad (16)$$

すなわち、行列 K を対角化し、この対角行列 Γ を用いてベクトル α の変換を行う。

最大固有値の問題は、次のように変換されている。

$$\lambda = \frac{(\Gamma U^T \alpha)^T U^T W U (\Gamma U^T \alpha)}{(\Gamma U^T \alpha)^T U^T U (\Gamma U^T \alpha)} \quad (17)$$

この式に、次を代入する。

$$\beta = \Gamma U^T \alpha \quad (18)$$

この結果、最大固有値問題は、次のようになる。

$$\lambda = \frac{\beta^T U^T W U \beta}{\beta^T U^T U \beta} \quad (19)$$

あるいは、次のような固有値問題に帰着することができる。

$$\lambda U^T U \beta = U^T W U \beta \quad (20)$$

これを整理すると、次の簡単な固有値問題になる。

$$\lambda \beta = U^T W U \beta \quad (21)$$

以上の準備のもとで、KFDアルゴリズムは、次のように整理される。

Step1 行列 K , W 計算

定義式にしたがって行列 K , W を計算する。Kernelの計算はガウシアンKernelの場合には指数関数の簡単な計算に帰着される。

Step2 固有値分解

固有値分解により行列 K を分解する。

Step3 固有ベクトル β と固有値

固有ベクトル β と固有値を計算する。

Step4 固有ベクトル ν 計算

係数 α を用いて、変換行列を介して固有ベクトル ν を計算する。ここで、 $\alpha = U \Gamma^{-1} \beta$ の関係を用いる。

Step5 クラスタに属するか判別

次の式を用いて未知の入力データである変数ベクトル z を有するサンプルが、どちらのクラスタに属するかを、次の関数により推定する。

$$(v^T \cdot \phi(z)) = \sum_{f=1}^N \sum_{g=1}^{n_f} \alpha_{fg} k(x_{fg}, z) \quad (22)$$

すなわち、学習データに対して上の計算を行い(変数 z に学習データ用いる)、それぞれのクラスタにおける上の計算式の値の平均値、あるいは分布を求めておく。この平均値の開きを、クラスタ分類のしきい値として用いる。

3 人工的データによる検証

3.1 人工的データの生成

本論文の手法を、実際のデータ集合のクラスタ分類に適用する前に、人工的に作成したデータに対するシステムの性能を調べる。具体的には、あらかじめ外的基準が定まっているクラスタ c を準備し、期待されたとおりにクラスタ分類がなされるかを調べる。

以下では、文献[16]で述べられている例を示すにとどめる。次のような2次元平面に配置される2つのクラスのデータを、それぞれについて200個ずつ人工的に生成する。

クラスタ1： $X \sim N(0, \sqrt{2})$, $Y_i = X_i * X_i + N(0, 0.01)$

クラスタ2： $X \sim N(2, 0.001)$, $Y \sim N(2, 0.001)$

この2つのクラスに対してKernel-based手法による判別分析を行う。最初的手法として、 $k(x, y) = (x, y)^d$, $d=2$ となるKernel functionを用いている[16]。ただし、以下の議論では、この場合を省略してガウシアンKernelの場合だけを示す。

学習データとして、それぞれから20サンプルを用い、判別の検証は200サンプルすべてについて行う。この結果、ガウシアンKernelを用いて判別を実施すると、すべてのサンプルが正確に判別されている。特に、Class 2についてはほとんど1つの点に変換されている。

3.2 判別分析の結果

図1には、生成されたデータを2次元平面上にプロットしている。この図より分かるように、これまでの非線形判別分析の手法を適用しても、領域の境界をクリアに描くことは極めて困難である。

図2には、判別結果をプロットしている。横軸方向がサンプル番号であり、縦軸には判別関数の値を示している。この結果から分かるように、すべてのサンプルが、もとの2つのクラスに正確に分類されている。

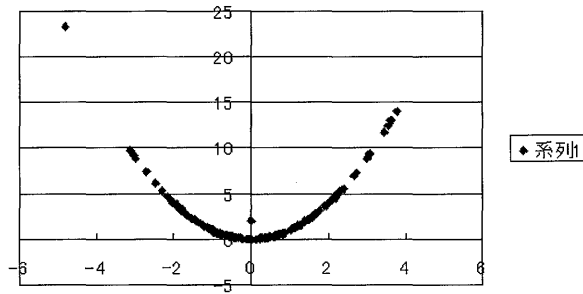


図1：データの2次元プロット

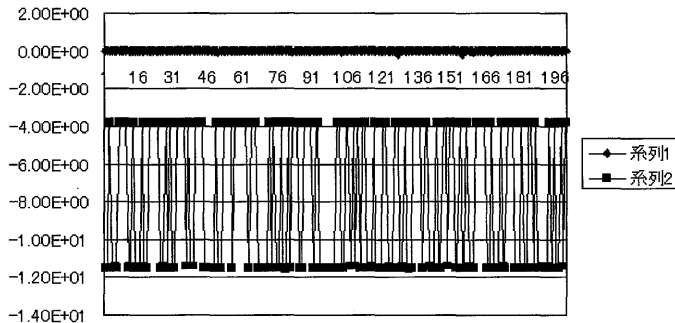


図2：判別関数の値の図示

4 クレジット審査への応用

4.1 German Credit概要

次に、やや実際のデータに対するクラスタ分類の例をとりあげ、本論文の手法の適用可能性を議論する。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。データの項目は、7つの数値データと、6個のカテゴリカルデータとからなる[19]。

このデータの本来の目的は、貸付審査の可否を決めるルールを求めることであるが、本論文で示すシミュレーションではクラスタを推定することに用いる。このデータセットを記述する数値型変数は次のようなものである。

- y1: クレジット期間 (クレジットによる完済までの期間で月数)
- y2: クレジット額 (貸付を希望している金額で単位はドイツマルクDM)
- y3: クレジット利率 (クレジットを受ける場合の利率)
- y4: 現住所での居住期間 (現在の住所に居住を始めてからの月数)
- y5: 年齢 (クレジット申請者の現在の年齢)
- y6: 当会社銀行でのクレジット開設数 (申請者がいくつのクレジットを同時に申し込んでいるか)
- y7: 扶養家族数 (申請者が扶養している家族数)

カテゴリ変数は、以下のような6個のカテゴリ変数となる。

- x1: 手形口座開設の内容 (マイナス, 小額200DM以下, 200DM以上, 口座なしの4カテゴリ)
- x2: 契約継続月数 (30月を境界とする数値を2カテゴリ化)
- x3: クレジット履歴 (履歴なし, 完済の履歴あり, 返済中あり, 返済遅延あり, 危険性ありの5カテゴリ)
- x4: 借入目的 (自動車購入など11カテゴリ)
- x5: 預金口座内容 (100DM以下100DM超500DM以下など5カテゴリ)
- x6: 保証人の有無 (なし, 共同申請, 保証人ありの3カテゴリ)

4.2 Kernel-based手法の適用

分析を分かりやすくするための、以下ではカテゴリ変数を判別分析の変数として用いた場合の結果を示す。このようなケースについては、以前から結果が示されているので、比較分析には都合が良い。

表1には、本論文で述べるKernel-based手法による判別結果、すなわち本来の2つのクラスタへの所属が、どの程度ただしく判別できたかを示している。比較の対象となった方法は、以下のようなものである。

C4.5: ID3を基本とする分岐の生成手法で枝切りがある

Neurorule: Baesenらにより実施された研究

P-NN: ニューラルネットワークで枝切りを実施

MDA：統計パッケージによる線形判別分析

ニューラルネットワークを用いて行う判別分析のシミュレーションのための条件は、以下のようしておく。

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

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[8]。この概要も表1に示している。

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、本論文で述べているKernel-based手法は、このNeuroruleの結果より、改善されることが示される。

表1 判別結果の比較（判別率の平均値%）

| C4.5 | P-NN | Neurorule | Trepan | Nefclass | Kernel-based手法 |
|-------|-------|-----------|--------|----------|----------------|
| 74.25 | 77.84 | 77.25 | 73.95 | 73.65 | 90.00 |

5 企業の格付問題と判別分析

5.1 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、本論文で述べているKernel-based手法に基づいた判別分析の方法を用いて、日本の企業における倒産をとりあげる。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[2]。

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ため、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。

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ため、入力データとして用いない。具体的には、*t*検定の手法を用いて、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、Wilcoxon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。

ニューラルネットワークを用いて行う判別分析のシミュレーションのための条件は、以下のようしておく。

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

表2には倒産予測の比較分析の結果を示している。これより分かるように、Kernel-based手法は、これまで先行研究において相対的に性能が良好であるとされているニューラルネットワーク手法より、性能が高いものとなっている。

表2 倒産予測の結果（判別率の平均値を示す%）

| カテゴリ | Kernel-based手法 | NN | MDA |
|------|----------------|-----|-----|
| 倒産 | 98% | 96% | 82% |
| 非倒産 | 98% | 90% | 78% |

5.2 債券格付問題への応用

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取りあげる[17][18]。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られるので、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。

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

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。

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多変量判別分析（MDA）を実施し、結果を表7.6にまとめている。NN手法の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

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3には、本論文で述べているKernel-based手法と他の手法との判別分析の結果を示している。これより分かるように、Kernel-bases手法による判別結果は、多変量解析による判別結果よりかなり良好であり、NNによる結果よりも優れている。したがって非線形判別の方法として最高レベルの結果を得ることができると言える。

表3 債券格付判別の結果比較（判別率の平均値%）

| クラス | Kernel-based手法 | NN | MDA |
|-----|----------------|--------|--------|
| A | 90.33% | 86.66% | 76.66% |
| B | 92.18% | 84.84% | 72.72% |
| C | 94.48% | 85.18% | 70.37% |

6 むすび

本論文では、本論文では、Kernel-based手法を用いた判別分析の高性能化とそのクレジット審査へ

の応用を示した。応用例として、シミュレーションにより本手法によるクレジット審査の有効性を議論し、現実の債券格付けなどとの対応関係について論じた。

今後の課題として、変動要因が更に多くなった場合の特性分析などがあり、検討を進める予定である。

参 考 文 献

- [1] E. I. Altman, *Corporate bankruptcy in America*, D. C. Health and Company, 1971.
- [2] 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.
- [3] 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.
- [4] K. Tan and S. Tokinaga, “Optimization of Fuzzy inference rules by using the Genetic Algorithm and its applications to the bond rating,” JORSJ, vol.42, no.3, pp.302-315, 1999.
- [5] K. Tan and S. Tokinaga, “The design of multi-stage fuzzy inference systems with smaller number of rules based on the Genetic Algorithm, Trans. IEICE Fundamentals, vol.E82-A, no.9, pp.1865-1873, 1999.
- [6] J. R. Quinlan, *C4.5 Programs for Machine Learning*, Morgan Kaufmann, Chambery, France, 1993.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] N. Nauck, “Data analysis with neuro-fuzzy methods, Habilitation Thesis, University of Magdeburg, 2000.
- [11] S. Tokinaga, J. Lu and Y. Ikeda, Neural network rule extraction by using the Genetic Programming and its applications to explanatory classifications,” IEICE Trans. Fuadamentals, vol.E88-A, no.10, pp.2627-2635, 2005.
- [12] J. Lu, S. Tokinaga and Y. Ikeda, “Neural network rule extraction by using the Genetic Programming and its applications to explanatory classification, Journal of Operations Research Society of Japan, vol.44, no.1, pp.222-223, 2005.
- [13] 呂 建軍, 岸川善紀, 時永祥三, “遺伝的プログラミングによるルール生成を用いたクラスタ特徴記述システムの構成とその応用”, 電子情報通信学会論文誌, vol.J89-A, no.12, pp.1142-1152, 2006.
- [14] B. Scholkopf and A. Smola, “Nonlinear component analysis as a kernel eigenvalue problem,” Neural Computation, vol.10, pp.1299-1319, 1998.
- [15] H. Kwon and N. M. Nasrabadi, “Kernel matched subspace detectors for hyperspectral target detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.28, no.2, pp.178-194, 2006.
- [16] G. Baudat and E. Anouar, “Generalized discriminant analysis using a Kernel approach,” Neural Computing, vol.12, pp.2385-2404, 2000.
- [17] 時永祥三, 譚康融, SASによる金融工学, オーム社, 2002.
- [18] 時永祥三編著, 複雑系とシステムリスク分析, 九州大学出版会, 2006.
- [19] <http://www.liacc.up.pt/ML/statlog/datasets/german/german.doc.html>