

特別研究報告書

サポートベクターマシンによる倒産予測

指導教官

福嶋 雅夫 教授
山下 信雄 助手

情報学科数理工学コース 平成9年度入学

山口 貴大

平成13年1月

サポートベクターマシンによる倒産予測

山口 貴大

摘要

1990年代初頭のバブル経済の崩壊により我が国の経済は長期間の低迷に陥っている．それに伴い企業の倒産件数が急激に増加している．これにより経営者や投資家のみならず，研究者の間でも企業の信用リスクの計測に高い関心が寄せられている．そして，多くの研究者たちによって大量に蓄えられた倒産データから倒産企業を予測する方法の開発が進められてきている．

一方，人工知能の分野では大量に蓄えられたデータの中から有用な指標を抽出するデータマイニングと呼ばれる手法が近年活発に研究されている．特に，サポートベクターマシン (Support Vector Machine: SVM) と呼ばれる手法が，現実の問題への応用において優れた性能を持つことが報告されている．

本報告書では，SVM を用いて企業の倒産を予測する方法を提案する．さらに SVM の学習結果を用いて企業の倒産確率を推定する方法を提案する．また実際に倒産した企業の財務データに対して提案した方法を適用した数値実験を行い，実務上よく用いられる判別分析との比較を行った．その結果，SVM の識別率は判別分析の識別率に比べて優れていることが確認された．また認識率において優れているだけでなく，倒産確率の推定においても SVM の学習結果は有用であることが分かった．

目次

1	序論	1
2	サポートベクターマシンの原理	1
2.1	パターン認識とは	1
2.2	線形サポートベクターマシン	2
2.2.1	分離可能な場合	2
2.2.2	分離不可能な場合	4
2.3	非線形サポートベクターマシン	6
2.3.1	カーネル関数	6
3	倒産予測への応用	7
3.1	非線形サポートベクターマシンによる倒産・非倒産の判別	7
3.2	倒産確率の推定	8
4	数値実験	9
4.1	数値実験で用いたデータ	9
4.2	実験環境	9
4.3	実験結果	10
4.3.1	倒産・非倒産の判別結果	10
4.4	確率密度関数の推定結果	10
4.5	結果の考察	10
5	結論	13
A	グラフ	16
A.1	学習結果を用いて推定された倒産確率	16
A.2	推定確率を表す関数により算出された倒産確率に対する企業数の分布	19

1 序論

1980年代に絶頂期を迎えた我が国の経済は1990年代に入り一転して長期間の低迷に陥った。それに伴い、企業の倒産件数が急激に増加し、企業は生き残りをかけて抜本的な経営改善を行わざるを得なくなった。このような状況の中で、多くの企業の間で、より良い経営判断のための材料として取引企業の信用リスク計測への関心が高まっている。

一方、情報技術の発達により記憶装置が大容量になり、またネットワークが整備され、その結果膨大な量のデータが蓄積されるようになった。このように無秩序に集められたデータから有用な情報を取り出すことは困難な作業である。1990年代中頃より、データマイニングと呼ばれる、膨大なデータから人間には簡単には分からない規則性を発見したり、未来を予測したりする手法が研究されている。

近年、パターン認識の分野でサポートベクターマシン (Support Vector Machine: SVM) という手法が、ニューラルネットワーク、nearest neighbor 等の他の認識手法と比較して優れた性能をもつことが示され、多くの研究者たちの注目を集めている。SVMは、1960年代にVapnikらによって提案された超平面による特徴空間の分割法に端を発する [4]。これを特に線形サポートベクターマシン (Linear Support Vector Machine: LSVM) と呼ぶ。LSVMは2クラスが超平面で分割できる (線形分離可能な) 場合には良い認識率を達成できるが、一般に線形分離可能である場合は少ない。1990年代に入って、カーネル関数と呼ばれる非線形関数を導入することによって線形分離可能でない場合に対しても良い認識率を示す手法が提案された [2]。これにより SVMは一気に注目を集めることになった。カーネル関数を用いて拡張された手法を特に非線形サポートベクターマシン (Nonlinear Support Vector Machine: NSVM) と呼ぶ。SVMの長所の一つとして学習が容易であることが挙げられる。SVMの学習は凸二次計画問題を解くことによって行うことができ、パーセプトロンなどに見られるような局所解の問題も存在しない。

本報告書では、SVMを用いて企業の倒産を予測することを試みる。しかしながら倒産するかしないかを正確に判別するのは一般には難しい。そこで本報告書では、SVMの学習によって得られた結果に基づいて倒産確率を推定する手法を提案する。さらにこの手法を用いて実際の企業の財務データに対して数値実験を行い、SVMの有用性について考察する。

本報告書はまず第2節でSVMの原理について説明する。次に第3節でSVMを用いて倒産を推定する方法を提案する。第4節では数値実験を行った結果を報告し、それに対する考察を述べる。最後に第5節で結論を述べる。

2 サポートベクターマシンの原理

この節では、まずパターン認識について説明し、次にSVMの数学的な原理を説明する。

2.1 パターン認識とは

ここでは、SVMについて説明する前にまず本報告書で扱うパターン認識とはどのようなものであるのかを説明する。

パターン認識とは認識対象がいくつかの概念に分類できるとき、観測されたパターンをそれらの概念のうちの一つに対応させることである [9]。この概念をクラスと呼ぶ。なお本報告書では倒産、非倒産の2クラスパターン認識を扱う。

l 個の観測されたデータが与えられているとする。各々の観測データは、特徴ベクトル $\boldsymbol{x}_i \in \mathbf{R}^n$, $i = 1, \dots, l$ とそれに割り当てられたクラス $y_i \in \{-1, 1\}$ の組からなる。 \boldsymbol{x}_i の各要素は計測された各特徴量を表す（本報告書では、 i が企業を表し、 \boldsymbol{x}_i がその企業の財務指標、 $y_i = 1$ が倒産、 $y_i = -1$ が非倒産を意味する。）。さらに関数 $f: \mathbf{R}^n \rightarrow \mathbf{R}$ が次の条件を満たすものとする。

$$\begin{aligned} y_i = 1 \text{ のとき } f(\boldsymbol{x}_i) &> 0 \\ y_i = -1 \text{ のとき } f(\boldsymbol{x}_i) &< 0 \end{aligned}$$

$f(\boldsymbol{x}) = 0$ を満たす点の集合は二つのクラスの境界面をなす。このような f を識別関数と呼ぶ。

パターン認識は二つの段階からなる。まずクラスが既知の観測データから識別関数 f を求め（学習段階）、次にクラスが未知のデータを f の正負により二つのクラスのどちらかに分類する（識別段階）。新たなデータ \boldsymbol{x} に対してクラス y は

$$y = \text{sgn}[f(\boldsymbol{x})]$$

で推定される。ここで $\text{sgn}[\cdot]$ は

$$\text{sgn}[\alpha] = \begin{cases} 1 & \alpha \geq 0 \text{ のとき} \\ -1 & \alpha < 0 \text{ のとき} \end{cases}$$

なる関数である。未知のデータを正しく分類した割合（識別率）が高い識別関数を求めることがパターン認識の目標である。

2.2 線形サポートベクターマシン

空間上の 2 種類のクラスのデータを超平面で分離することを考える。このとき、観測データを超平面で誤りなく完全に分離できる場合とそうでない場合がある。まず分離できる場合の手法を与え、次にそれを分離できない場合に拡張する。

2.2.1 分離可能な場合

まず、与えられた次の観測データが超平面で誤りなく分離できる場合を考える。

$$\begin{aligned} \{\boldsymbol{x}_i, y_i\}, i = 1, \dots, l &: \text{学習用観測データ} \\ \boldsymbol{x}_i \in \mathbf{R}^n &: \text{学習用観測データの座標} \\ y_i \in \{-1, 1\} &: \boldsymbol{x}_i \text{ に割り当てられたクラス} \end{aligned}$$

正のサンプルと負のサンプルを分離する超平面（分離超平面）の方程式を

$$H_0: \boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

とする。但し $\boldsymbol{a} \cdot \boldsymbol{b}$ はベクトル $\boldsymbol{a}, \boldsymbol{b}$ の内積を表す。ここで \boldsymbol{w} は超平面の法線ベクトルで、 b は定数項である。次に d_+ (d_-) を分離超平面から最も近い正（負）のサンプルまでの最短距離とする。このとき $d_+ + d_-$ を分離超平面のマージンと呼ぶ。線形分離可能な場合は SVM はマージン最大の分離超平面を求める問題として以下のように定式化できる。

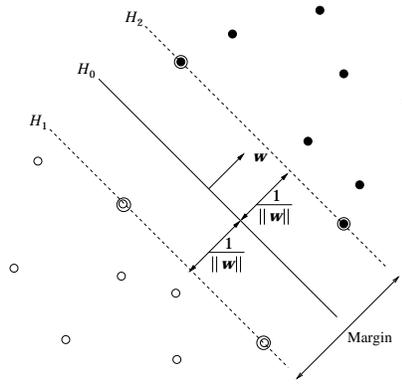


図 1: 線形分離可能な場合の分離超平面

すべての観測データは H_0 によって分離されるため、次の制約条件を満たさなければならない。

$$\mathbf{x}_i \cdot \mathbf{w} + b > 0 \quad y_i = +1 \text{ のとき} \quad (1)$$

$$\mathbf{x}_i \cdot \mathbf{w} + b < 0 \quad y_i = -1 \text{ のとき} \quad (2)$$

これらの不等式は (\mathbf{w}, b) に関する同次式であるから、 (\mathbf{w}, b) を適当に正数倍することによって次のように書くことも出来る。

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad y_i = +1 \text{ のとき} \quad (3)$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad y_i = -1 \text{ のとき} \quad (4)$$

さらにこれらは次の不等式にまとめることができる。

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad i = 1, \dots, l \quad (5)$$

次に式 (3),(4) の等号が成り立つ点を考える。それらは次の H_0 に平行で H_0 との距離が $1/\|\mathbf{w}\|$ である超平面 H_1, H_2 上にある。

$$H_1: \quad \mathbf{x}_i \cdot \mathbf{w} + b = +1$$

$$H_2: \quad \mathbf{x}_i \cdot \mathbf{w} + b = -1$$

そのため、 $d_+ = d_- = 1/\|\mathbf{w}\|$ となり、マージンは $2/\|\mathbf{w}\|$ となる (図 1)。従って、制約 (5) の下で $\|\mathbf{w}\|^2$ を最小化することによって、最大マージンを与える超平面の組を求めることができる。以上のことをまとめると、SVM は次の制約つき最適化問題に定式化できる [1]。

$$\begin{aligned} & \text{minimize} && \|\mathbf{w}\|^2 \\ & \text{subject to} && y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad i = 1, \dots, l \end{aligned} \quad (6)$$

なお式 (5) の等号が成り立ち (すなわち超平面 H_1, H_2 のどちらかの上) にあり、それを取り除くと求められる解が変化するような点を支持ベクトル (support vector) と呼ぶ。そのため、このような手法をサポートベクターマシンと呼ぶ。

一般に制約付き最小化問題は、Lagrange 関数を用いると、より扱いやすい双対問題に帰着できることが多い [5]。問題 (6) の Lagrange 関数を

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) \equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i \{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1\} \quad (7)$$

とする．ここで α は Lagrange 乗数 $\alpha = (\alpha_1, \dots, \alpha_l)$ ($\alpha_i \geq 0$) である． $L(\mathbf{w}, b, \alpha)$ を用いて問題 (6) を書き換えると次のようになる．

$$\text{minimize} \quad \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha) \quad (8)$$

さらにこの問題の双対問題は次のようになる．

$$\begin{aligned} & \text{maximize} \quad \min_x L(\mathbf{w}, b, \alpha) \\ & \text{subject to} \quad \alpha_i \geq 0 \quad i = 1, \dots, l \end{aligned} \quad (9)$$

この最適化問題は，はじめに \mathbf{w}, b に関して L を最小化し，次に α に関して最大化する問題である．最小化問題 $\min_x L(\mathbf{w}, b, \alpha)$ の最適解においては L の勾配が 0 になるので，次のような式が成立する．

$$\nabla_{\mathbf{w}} L(\mathbf{w}, b, \alpha) = \mathbf{0}, \quad \nabla_b L(\mathbf{w}, b, \alpha) = 0$$

これから次の式が導かれる．

$$\sum_{i=1}^l \alpha_i y_i = 0 \quad (10)$$

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i \quad (11)$$

よって問題 (9) は次のような最適化問題に書き換えることができる．

$$\begin{aligned} & \text{maximize} \quad \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ & \text{subject to} \quad \alpha_i \geq 0 \quad i = 1, \dots, l, \\ & \quad \quad \quad \sum_{i=1}^l \alpha_i y_i = 0 \end{aligned}$$

この問題は元の問題 (6) に比べて，制約が簡単な形をしている．そのため容易に解くことができる．

2.2.2 分離不可能な場合

分離可能なデータに対する問題 (6) を分離不可能なデータに適用すると実行可能解が得られない．そこで正の Slack 変数 $\xi_i, i = 1, \dots, l$ を制約条件に導入することによって，制約条件 (3), (4) を緩めることを考える．つまり次のような制約条件を考える．

$$\begin{aligned} \mathbf{x}_i \cdot \mathbf{w} + b &\geq +1 - \xi_i \quad y_i = +1 \text{ のとき } (i \in \mathcal{I} \text{ と表す}) \\ \mathbf{x}_i \cdot \mathbf{w} + b &\leq -1 + \xi_i \quad y_i = -1 \text{ のとき } (i \in \mathcal{J} \text{ と表す}) \\ \xi_i &\geq 0 \quad i = 1, \dots, l \end{aligned}$$

これらは

$$y_i (\mathbf{x}_i \cdot \mathbf{w} + b) \geq +1 - \xi_i \quad (12)$$

$$\xi_i \geq 0 \quad i = 1, \dots, l \quad (13)$$

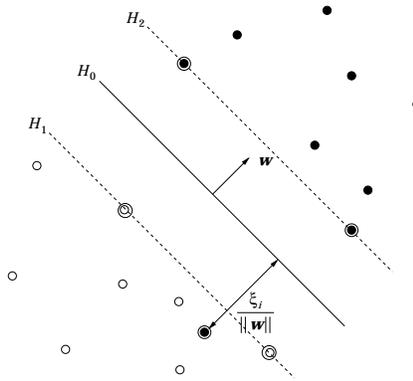


図 2: 線形分離ができない場合の分離超平面

と書き換えることができる．また ξ_i は出来るだけ小さい方が好ましいので $C_+ \sum_{i \in \mathcal{I}} \xi_i + C_- \sum_{i \in \mathcal{J}} \xi_i$ という項を問題 (6) の目的関数に加える．但し, C_+ と C_- は適当な正の定数である．このとき, 謬った分類をできるだけ少なくし, かつマージンができるだけ大きくなるような超平面を求める問題は次のような最適化問題に定式化できる．

$$\begin{aligned}
 & \text{minimize} && \frac{1}{2} \|\mathbf{w}\|^2 + C_+ \sum_{i \in \mathcal{I}} \xi_i + C_- \sum_{i \in \mathcal{J}} \xi_i \\
 & \text{subject to} && y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq +1 - \xi_i \\
 & && \xi_i \geq 0 \quad i = 1, \dots, l
 \end{aligned} \tag{14}$$

前節と同様にしてこの問題の双対問題を考えると以下のように定式化できる．

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\
 & \text{subject to} && 0 \leq \alpha_i \leq C_+ \quad y_i \in \mathcal{I} \\
 & && 0 \leq \alpha_i \leq C_- \quad y_i \in \mathcal{J} \\
 & && \sum_{i=1}^l \alpha_i y_i = 0
 \end{aligned} \tag{15}$$

スラック変数の導入により, 元の問題の方は複雑な形になるが, Lagrange 乗数を用いて定式化すると, 双対問題は比較的シンプルな形になることが分かる．

双対問題 (15) の最適解より主問題 (14) の最適解 \mathbf{w}, b を求める方法を与える．再び Lagrange 関数を

$$L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{w}\|^2 + C_+ \sum_{i \in \mathcal{I}} \xi_i + C_- \sum_{i \in \mathcal{J}} \xi_i - \sum_{i=1}^l \alpha_i \{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\} - \sum_{i \in \mathcal{I}} \beta_i \xi_i - \sum_{i \in \mathcal{J}} \beta_i \xi_i$$

とする．但し, β_i は制約条件 (13) に対する Lagrange 乗数である．問題 (14) の最適解において以下の Karush-Kuhn-Tucker 条件が成り立つ [5, 6]．

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial w_k} = w_k - \sum_{i=1}^l \alpha_i y_i x_{ik} = 0 \quad k = 1, \dots, l \tag{16}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = - \sum_{i=1}^l \alpha_i y_i = 0 \tag{17}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = C_+ - \alpha_i - \beta_i = 0 \quad i \in \mathcal{I} \quad (18)$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \xi_i} = C_- - \alpha_i - \beta_i = 0 \quad i \in \mathcal{J} \quad (19)$$

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0 \quad i = 1, \dots, l \quad (20)$$

$$\xi_i \geq 0 \quad i = 1, \dots, l \quad (21)$$

$$\alpha_i \geq 0 \quad i = 1, \dots, l \quad (22)$$

$$\beta_i \geq 0 \quad i = 1, \dots, l \quad (23)$$

$$\alpha_i \{y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\} = 0 \quad i = 1, \dots, l \quad (24)$$

$$\beta_i \xi_i = 0 \quad i = 1, \dots, l \quad (25)$$

ここで w_k, x_{ik} はそれぞれベクトル \mathbf{w}, \mathbf{x}_i の第 k 要素を表す． \mathbf{w} は式 (16) より

$$w_k = \sum_{i=1}^l \alpha_i y_i x_{ik}$$

で計算できる．これは式 (11) から明らかである．式 (18),(19) と式 (25) より $\alpha_i < C_+$ ($i \in \mathcal{I}$) または $\alpha_i < C_-$ ($i \in \mathcal{J}$) ならば $\xi_i = 0$ となる．したがって $0 < \alpha_i < C_+$ ($i \in \mathcal{I}$) あるいは $0 < \alpha_i < C_-$ ($i \in \mathcal{J}$) なる i に対して式 (24) より b は

$$b = - \left(\mathbf{x}_i \cdot \mathbf{w} + \frac{1}{y_i} \right)$$

で計算できる．[1]．

2.3 非線形サポートベクターマシン

2.3.1 カーネル関数

LSVM は 2 クラスが超平面で分割できる (線形分離可能な) 場合には良い認識率を達成できる．だが一般には線形分離可能であるとは限らない．そこで，写像

$$\Phi: \mathbf{R}^n \mapsto \mathcal{H}$$

を用いてサンプルデータをより高次元の空間 \mathcal{H} に写して， \mathcal{H} 上で線形分離を行うことが提案されている [2]．このように高次元の空間で線形分離を行えば，実質的にもとの空間で非線形分離を行っていることになる (図 3)．このような写像 Φ をつかえば，識別関数は式 (11) より

$$\begin{aligned} f(\Phi(\mathbf{x})) &= \mathbf{w} \cdot \Phi(\mathbf{x}) + b \\ &= \sum_{i=1}^l \alpha_i y_i \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}_i) + b \end{aligned}$$

のように書ける．考える空間 \mathcal{H} の次元が大きければ $\Phi(\mathbf{x})$ の計算に時間がかかる．しかし，内積 $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ が $\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)$ を陽に知ることなく $\mathbf{x}_i, \mathbf{x}_j$ のみから計算できる場合には計算量が大幅に減る．すなわち

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$

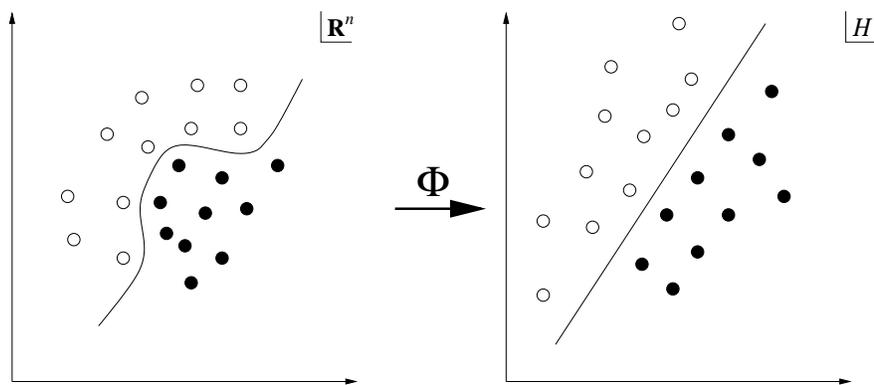


図 3: 高次元空間 \mathcal{H} 上での線形分離

となるようなカーネル関数 K が存在するならば K を使い, Φ を陽に知る必要はない [11]. このようにして Φ の計算を避ける方法はカーネルトリックと呼ばれる [7].

しかしながら, 適当に選んだカーネル関数が \mathcal{H} 上の内積になっているとは限らない. 次の定理を満たすカーネル関数は \mathcal{H} 上の内積で表されるということが知られている [1, 3, 7, 10, 11].

定理 1 (Mercer) 関数 K がある写像 Φ を用いて

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^l \Phi_i(\mathbf{x})\Phi_i(\mathbf{y})$$

と展開できるための必要十分条件は $\int g(\mathbf{x})^2 d\mathbf{x}$ が有限となるような任意の $g(\mathbf{x})$ に対して

$$\int \int K(\mathbf{x}, \mathbf{y})g(\mathbf{x})g(\mathbf{y})d\mathbf{x}d\mathbf{y} \geq 0$$

が成り立つことである.

Mercer の条件を満たすカーネル関数 K で, 計算が簡単なものとしては

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (26)$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2} \quad (27)$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x} \cdot \mathbf{y} - \delta) \quad (28)$$

などが知られている [1]. 但し, $p, \sigma, \kappa, \delta$ は適当な定数である.

3 倒産予測への応用

この節では, NSVM を用いて倒産・非倒産を判別するために, 企業の情報を数値化する方法を提案する. さらに NSVM の学習結果に対してシグモイド関数を用いることにより企業の倒産確率を推定する方法を提案する.

3.1 非線形サポートベクターマシンによる倒産・非倒産の判別

サポートベクターマシンで倒産・非倒産を判別するためには倒産・非倒産企業の情報が必要である. しかしながら生の財務データでは倒産・非倒産を判別する座標データには適さないので, 表 1

表 1: 倒産予測に用いる財務指標

番号	財務指標の名前	計算式
1	流動比率	流動資産/流動負債
2	自己資本比率	自己資本/総資本
3	借入金依存度	(割引手形+短期借入金+長期借入金)/総資本
4	キャッシュフロー対流動負債比率	キャッシュフロー/流動負債
5	売上高総利益率	売上総利益/売上高
6	売上高経常利益率	経常利益/売上高
7	売上高支払利息比率	支払利息割引料/売上高
8	総資本経常利益率	経常利益/総資本
9	総資本営業利益率	営業利益/総資本
10	経営資本営業利益率	営業利益/経営資本
11	自己資本当期利益率	当期利益/自己資本
12	当座比率	(現金・預金+受取手形+売掛金)/流動負債
13	総資本回転率	売上高/総資本
14	固定資産回転率	売上高/固定資産

に挙げたような財務指標を計算することが提案されている [13] . 倒産・非倒産にあまり関係のない指標を用いると判別の精度が悪くなる . そこで表 1 の財務指標の中から倒産に関係のあると思われる指標を変数増減法 [8] で選び , これらを要素とするベクトル列 $\{x_i\}$ を生成する . 変数増減法は多変量解析で用いられる変数選択法であるがここでは詳しく触れない . さらにベクトル x_i に

$$y_i = \begin{cases} 1, & i \text{ が倒産企業するとき} \\ -1, & i \text{ が非倒産企業するとき} \end{cases}$$

なるクラスを割り当てる .

3.2 倒産確率の推定

ここでは SVM の学習結果から , 倒産確率を推定する方法を提案する . 識別関数値が z となる企業の倒産確率はシグモイド関数

$$g(z; \varepsilon) = \frac{1}{1 + e^{-\frac{z}{\varepsilon}}}$$

で表される関数 g で計算できると仮定する (図 4) . ここで ε は $z = 0$ 付近の勾配を決めるパラメータである .

まずサンプルデータの倒産確率の離散的な分布を求める . これは識別関数値 z を微少区間に分割し , それぞれの区間に含まれる全企業数に対する倒産企業数の割合を計算することにより求められる .

$\Delta > 0$: 分割幅

n_+^k : $z \in [(k-1)\Delta, k\Delta)$ なる倒産企業数 ($k = 0, \pm 1, \pm 2, \dots$)

n_-^k : $z \in [(k-1)\Delta, k\Delta)$ なる非倒産企業数 ($k = 0, \pm 1, \pm 2, \dots$)

とすれば

$$r_k = \frac{n_+^k}{n_+^k + n_-^k} : \text{区間 } [(k-1)\Delta, k\Delta) \text{ における倒産企業比率}$$

$$z_k = \frac{2k-1}{2} \Delta : \text{区間 } [(k-1)\Delta, k\Delta) \text{ の中点}$$

なる離散的な分布 $\{(z_k, r_k)\}$ が求められる。

次に $\{(z_k, r_k)\}$ に対して非線形最小二乗法を適用することにより関数 g を求める。すなわち、各区間の倒産企業比率 r_k とその区間の midpoint の関数の値 $g(z_k)$ との差の二乗の総和

$$\sum_{k \in S} \{r_k - g(z_k; \varepsilon)\}^2$$

が最小となるような ε を求める。但し S は $n_+^k + n_-^k \neq 0$ であるような k の添字集合である。ここで求めた関数 g により、企業の倒産確率を推定することができる。

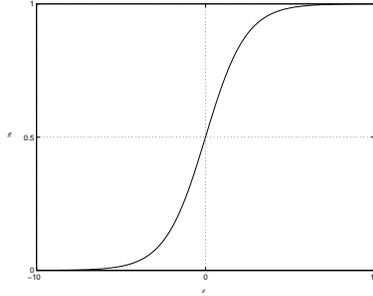


図 4: シグモイド関数

4 数値実験

前節で提案した方法により実際に倒産・非倒産の判別実験を行った。また NSVM の有用性を示すために LSVM、判別分析と結果を比較した。さらに、三つの手法によって得られた識別関数値によりそれぞれ倒産確率を推定する関数を推定した。なお判別分析 [8] は多変量解析の分野における線形識別関数による分類法の一つであるがここでは詳しく触れない。

4.1 数値実験で用いたデータ

東京商工リサーチから提供された 1999 年、2000 年の建設企業の各企業 1 期分の財務データを用いる。使用する財務データは条件を等しくするため資本金 1 億円以下あるいは従業員 300 人以下の中小企業に規模を限定し、表 2 のようにサンプルを無作為に抽出し、実験を行った。なお、学習用データ、判別実験用データ及び倒産確率を表す関数を推定するためのデータには重複するサンプルはない。表 1 に挙げた財務指標の中から変数増減法により番号 2,3,5,7,13 の指標を予測に用いる指標として選択した。

4.2 実験環境

本実験は Sun Microsystems 社の SUN ULTRA 60 上で、UNIX SunOS 5.7 の環境のもとで行った。SVM のプログラムは MathWorks 社の MATLAB5.3.1 の MATLAB 言語によりコーディングした。SVM で二次計画問題を解く際に、MATLAB の Optimization Toolbox が提供する quadprog

表 2: 実験に用いたデータ

データ	学習		予測		倒産確率の推定	
	倒産	非倒産	倒産	非倒産	倒産	非倒産
1	50	50	100	100	50	50
2	100	100	100	100	50	50
3	150	150	100	100	50	50
4	200	200	100	100	50	50
5	50	300	100	100	50	50
6	100	400	100	100	50	50

関数を用いた．なお判別分析は [12] で提供されるオンラインの統計解析プログラムを利用した．倒産確率の推定の際の非線形最小二乗法の計算には gnuplot の fit 関数を用いた．

4.3 実験結果

4.3.1 倒産・非倒産の判別結果

倒産・非倒産の判別結果を表 3 にまとめる．なお，NSVM で用いたカーネル関数は $K(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$ であり $\sigma = 20$ とした．データ 1 からデータ 4 については $C_+ = C_- = 1 \times 10^{100}$ とし，データ 5 については $C_+ = 6 \times 10^{100}, C_- = 1 \times 10^{100}$ ，データ 6 については $C_+ = 4 \times 10^{100}, C_- = 1 \times 10^{100}$ とした．

NSVM に対しては $\sigma = 1 \times 10^{-5}, \sigma = 1 \times 10^{10}$ でも実験を行った．それらの判別結果を $\sigma = 20$ としたときの結果とともに表 4 に示す．

4.4 倒産確率の推定結果

離散分布の計算に用いた分割幅 Δ とシグモイド関数 g のパラメータ ε の計算結果を表 5 に示す．倒産企業の離散分布と推定確率を表す関数 $g(z; \varepsilon)$ のグラフを付録 A.1 の図 5 から図 16 に示す．

4.5 結果の考察

前節で得られた結果 (表 3) を見ると，データ 5, 6 のように倒産企業と非倒産企業のサンプル数が極端に違っていると判別結果が悪くなることがわかる．識別率に関して NSVM は学習に関して判別分析より優れていることがわかる．

次に NSVM の学習の性質を考察する． $\sigma = 1 \times 10^{-5}$ の場合，学習は完璧に行われている．実際識別関数値は，倒産企業はすべて 1，非倒産企業はすべて -1 となっている．一方識別データの方はすべての企業の識別関数値が 0 となっている (識別関数値が 0 の場合は誤識別とした)．これは学習段階で求めた識別関数が学習データのみにはしか適応しない関数になっていることが原因と考えられる (過学習の問題)． $\sigma = 1 \times 10^{10}$ の場合，学習データも識別データもすべてのサンプルの識別関数値は 3.085×10^{-6} となっている．もはやこれは学習をしているとは言えない．パラメータ σ の値は学習の度合いを決定するものであると言える．ここであげた結果から分かるように，学習

表 3: 倒産・非倒産の判別結果

学習

データ	手法	識別率	
		倒産	非倒産
1	NSVM	42/50(84.0%)	42/50(84.0%)
	LSVM	21/50(42.0%)	41/50(82.0%)
	判別分析	32/50(64.0%)	38/50(76.0%)
2	NSVM	76/100(76.0%)	73/100(70.0%)
	LSVM	76/100(76.0%)	60/100(60.0%)
	判別分析	72/100(72.0%)	72/100(72.0%)
3	NSVM	120/150(80.8%)	120/150(80.0%)
	LSVM	95/150(63.3%)	107/150(71.3%)
	判別分析	118/150(78.7%)	95/150(63.3%)
4	NSVM	153/200(76.5%)	155/200(77.5%)
	LSVM	131/200(65.5%)	163/200(81.5%)
	判別分析	128/200(64.0%)	159/200(79.5%)
5	NSVM	14/50(28.0%)	297/300(99.0%)
	LSVM	10/50(20.0%)	294/300(98.0%)
	判別分析	27/50(54.0%)	268/300(89.3%)
6	NSVM	49/100(49.0%)	386/400(96.5%)
	LSVM	55/100(55.0%)	342/400(85.5%)
	判別分析	60/100(60.0%)	352/400(88.0%)

予測

データ	手法	識別率	
		倒産	非倒産
1	NSVM	63/100(63.0%)	68/100(68.0%)
	LSVM	45/100(45.0%)	81/100(81.0%)
	判別分析	70/100(70.0%)	77/100(77.0%)
2	NSVM	74/100(74.0%)	80/100(80.0%)
	LSVM	72/100(72.0%)	68/100(68.0%)
	判別分析	66/100(66.0%)	79/100(79.0%)
3	NSVM	79/100(79.0%)	73/100(73.0%)
	LSVM	73/100(73.0%)	68/100(68.0%)
	判別分析	77/100(77.0%)	81/100(81.0%)
4	NSVM	75/100(75.0%)	75/100(75.0%)
	LSVM	65/100(65.0%)	84/100(84.0%)
	判別分析	74/100(74.0%)	80/100(80.0%)
5	NSVM	14/100(14.0%)	98/100(98.0%)
	LSVM	36/100(36.0%)	94/100(94.0%)
	判別分析	56/100(56.0%)	87/100(87.0%)
6	NSVM	28/100(28.0%)	91/100(91.0%)
	LSVM	81/100(81.0%)	69/100(69.0%)
	判別分析	51/100(51.0%)	85/100(85.0%)

表 4: NSVM のパラメータの違いによる判別結果 (データ 1)

学習

データ	σ	識別率	
		倒産	非倒産
1	20	42/50(84.0%)	42/50(84.0%)
	1×10^{-5}	50/50(100%)	50/50(100%)
	1×10^{10}	50/50(100%)	0/50(0%)
2	20	76/100(76.0%)	73/100(70.0%)
	1×10^{-5}	100/100(100%)	100/100(100%)
	1×10^{10}	100/100(100%)	0/100(0%)
3	20	120/150(80.8%)	120/150(80.0%)
	1×10^{-5}	50/150(100%)	50/150(100%)
	1×10^{10}	150/150(100%)	0/150(0%)
4	20	153/200(76.5%)	155/200(77.5%)
	1×10^{-5}	200/200(100%)	200/200(100%)
	1×10^{10}	0/200(0%)	200/200(100%)
5	20	14/50(28.0%)	297/300(99.0%)
	1×10^{-5}	50/50(100%)	300/300(100%)
	1×10^{10}	0/50(0%)	300/300(100%)
6	20	49/100(49.0%)	386/400(96.5%)
	1×10^{-5}	100/100(100%)	400/400(100%)
	1×10^{10}	0/100(0%)	400/400(100%)

予測

データ	σ	識別率	
		倒産	非倒産
1	20	63/100(63.0%)	68/100(68.0%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	100/100(100%)	0/100(0%)
2	20	74/100(74.0%)	80/100(80.0%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	100/100(100%)	0/100(0%)
3	20	79/100(79.0%)	73/100(73.0%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	100/100(100%)	0/100(0%)
4	20	75/100(75.0%)	75/100(75.0%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	100/100(100%)	0/100(0%)
5	20	9/50(18.0%)	287/300(95.7%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	0/100(0%)	100/100(100%)
6	20	28/100(28.0%)	373/400(93.3%)
	1×10^{-5}	0/100(0%)	0/100(0%)
	1×10^{10}	0/100(0%)	100/100(100%)

表 5: 分割幅 Δ と確率密度関数の係数 ϵ

データ	NSVM		LSVM		判別分析	
	Δ	ϵ	Δ	ϵ	Δ	ϵ
1	0.2	0.54074	1×10^{14}	1.73563×10^{14}	0.2	0.244356
2	0.2	0.316899	5×10^5	3.05037×10^5	1	1.03245
3	0.2	0.419418	30	150.525	0.2	3.21217
4	0.4	0.256704	1	1.25928	0.5	0.698909

段階においてただ闇雲に高識別率を目指せば良いというものではない．過学習の問題を考慮して最適なパラメータの値を設定しなければならない．

次に倒産確率の推定結果について考察を加える．付録 A.2 の図 17 から図 28 のグラフを見ると NSVM は確率 0.5 を中心として倒産・非倒産企業がほぼ対称に分布しているのに対して，LSVM と判別分析にはその傾向が見られない．また，LSVM と判別分析では確率 0.5 近辺にサンプルが集中している傾向があり，倒産確率推定の意味をなさない．一方，NSVM はその傾向が少ない．このことから識別率だけでなく倒産確率の推定においても NSVM の学習結果が優れていることが分かる．また判別分析をデータ 3 に適用した結果（図 27）では倒産・非倒産企業ともほぼ確率 0.5 以上に分布しており，倒産確率推定の意味をなさない．

5 結論

本報告書ではパターン認識手法としての SVM を企業の倒産予測に応用しその有用性を検証した．実験によりパターン認識の一手法として SVM が他の手法より学習においては優れていることが分かった．また倒産予測において概ね 70～80%であり，実務で用いる上で良好な結果を得るにはいたらなかった．これは企業の財務データが一期分しかなかったこと，選択した指標が倒産・非倒産を判定する基準としてはやや適切さに欠けたことなどが原因に挙げられる．また，非倒産企業の中に倒産はしていないものの経営状態が必ずしも健全とは言えない企業が含まれていることも考えられる．

今後は複数期間の財務データを用いてそこから有用な指標を抽出することを考える．また非倒産企業としてデータをとった年度から数年間倒産していない企業を採用するのが好ましいと考えられる．これは前述したように非倒産企業のなかに経営状態が健全でない企業が含まれる可能性があるためである．倒産・非倒産のサンプル数が大きく違うときに良い認識率を達成できる手法を与えることも重要である．

謝辞

日頃から御教授頂き、本研究に対しても熱心な御指導を賜った福島雅夫教授，ならびに本報告書作成にあたり細部に至るまで貴重な御指摘と御指導を頂いた山下信雄助手に深く感謝の意を表します．また，財務データを快く提供して頂いた東京商工リサーチ，大変お世話になった滝根哲哉助教授をはじめとする福島研究室の皆様に厚く御礼申し上げます．

参考文献

- [1] Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, Vol. 2 (1998).
- [2] Cortes, C. and Vapnik, V.: Support-Vector Networks, *Machine Learning*, Vol. 20 (1995), pp.273–297.
- [3] Smola, A. and Schölkopf, B.: A tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
- [4] Vapnik, V. and Chervonenkis, A.: A note on one class of perceptrons, *Automation and Remote Control*, Vol. 25 (1964).
- [5] 福島雅夫：非線形最適化の理論, 産業図書, 1980.
- [6] 福島雅夫：数理計画入門, システム制御情報ライブラリー 15, 朝倉書店, 1996.
- [7] 津田宏治：サポートベクターマシンとは何か, 電子情報通信学会誌, Vol. 83 (2000), pp460–466.
- [8] 河口至商：多変量解析入門 I, 森北出版, 1973.
- [9] 茨木俊秀, 福島雅夫：最適化の手法, 情報数学講座 14, 共立出版, 1993.
- [10] 高須淳宏：Support Vector Machine による分類, 森下, 宮野 (編), bit 別冊 “発見科学とデータマイニング”, 共立出版, 2000, 第 12 章, pp.118–127.
- [11] 赤穂昭太郎, 津田宏治：サポートベクターマシン, 数理科学, (2000), pp.52–58.
- [12] 青木繁伸：Black-Box : <http://aoki2.si.gunma-u.ac.jp/BlackBox/BlackBox.html>.
- [13] 石川明男, 金崎芳輔：企業倒産の判別分析と倒産確率の推定, 証券アナリストジャーナル, Vol. 38 (2000), pp.87–101.

A グラフ

A.1 学習結果を用いて推定された倒産確率

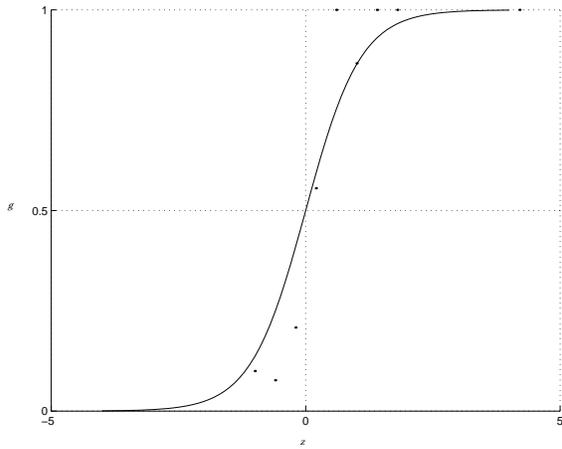


図 5: データ 1 の倒産確率 (NSVM)

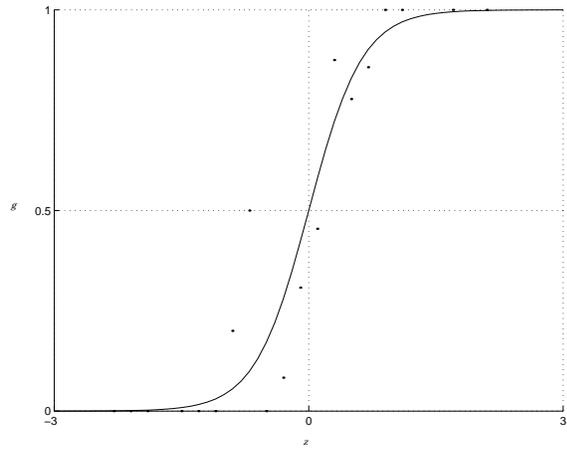


図 6: データ 2 の倒産確率 (NSVM)

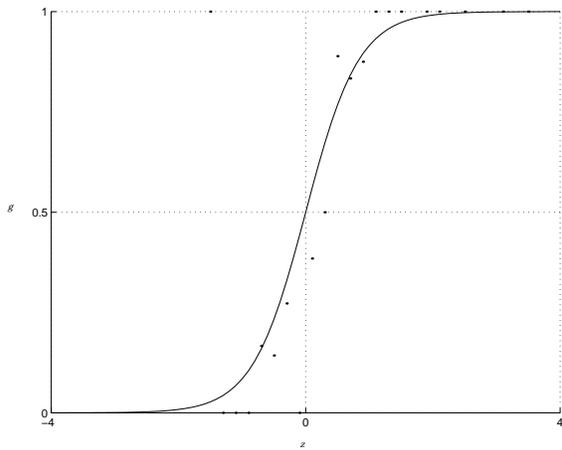


図 7: データ 3 の倒産確率 (NSVM)

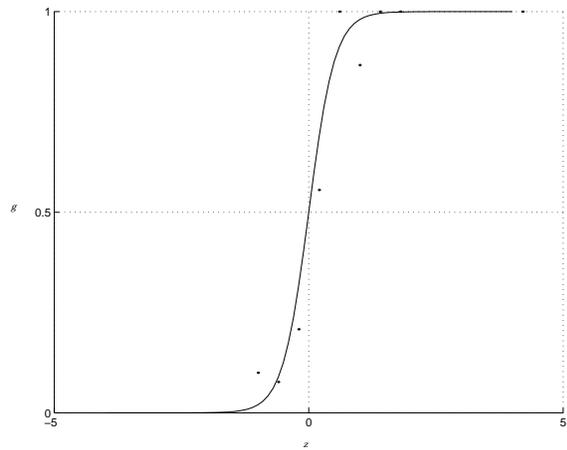


図 8: データ 4 の倒産確率 (NSVM)

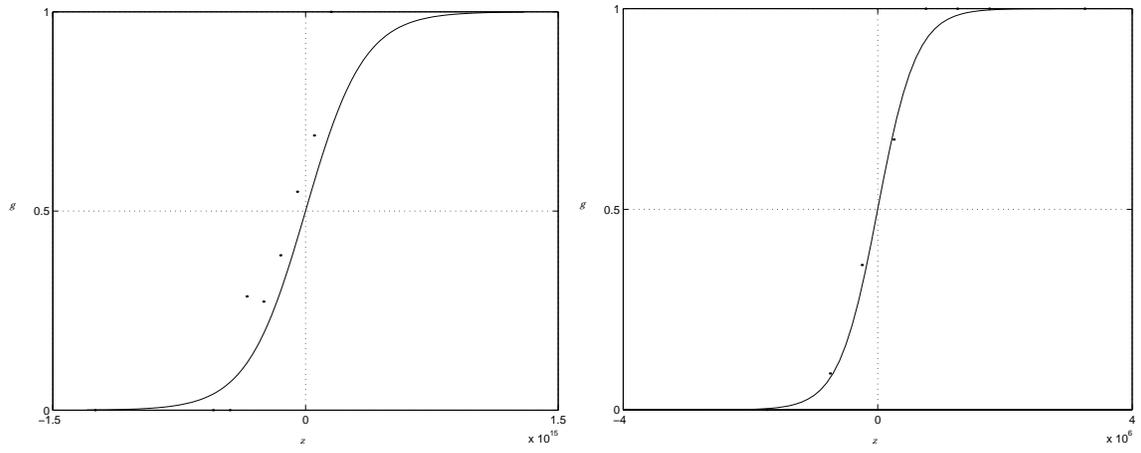


図 9: データ 1 の倒産確率 (LSVM)

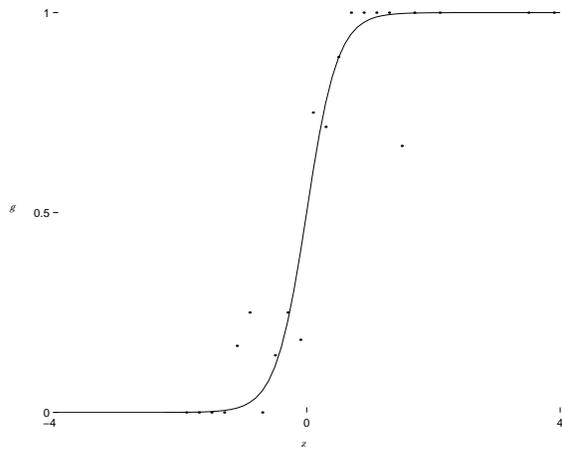


図 13: データ 1 の倒産確率 (判別分析)

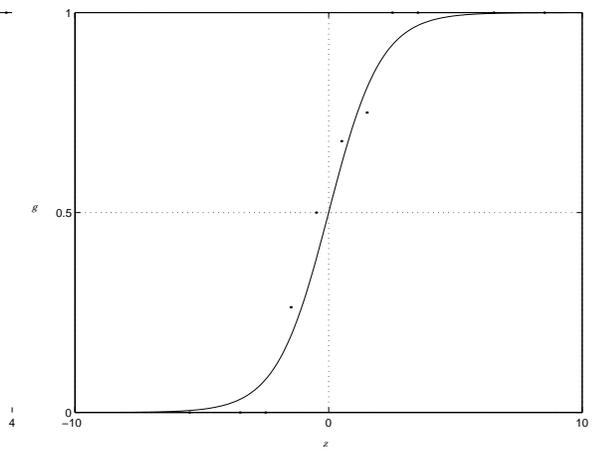
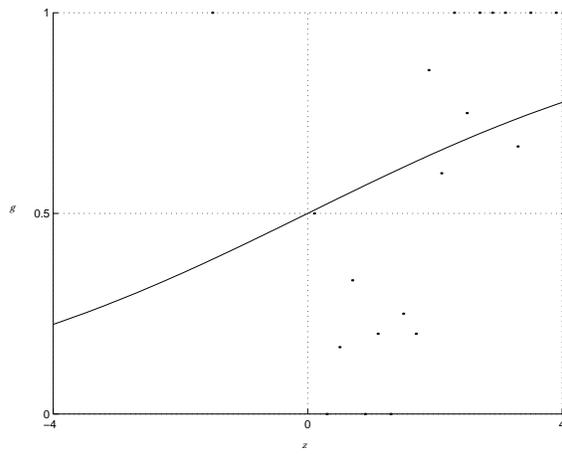


図 14: データ 2 の確率密度関数 (判別分析)



A.2 推定確率を表す関数により算出された倒産確率に対する企業数の分布

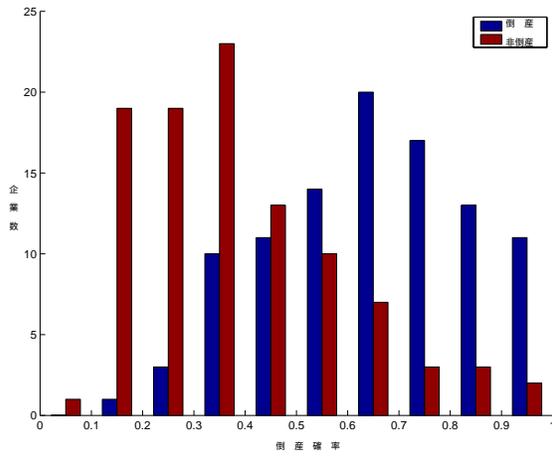


図 17: 識別データ 1 の分布 (NSVM)

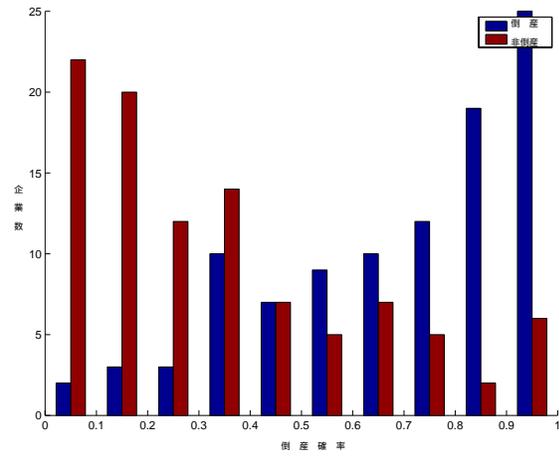


図 18: 識別データ 2 の分布 (NSVM)

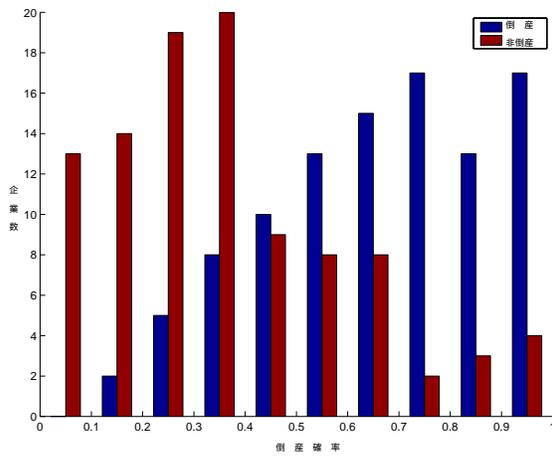


図 19: 識別データ 3 の分布 (NSVM)

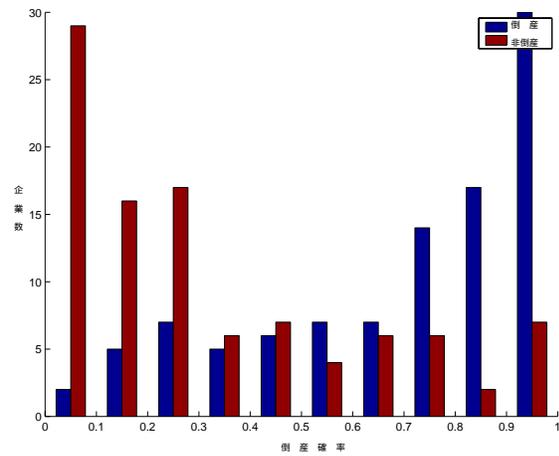


図 20: 識別データ 4 の分布 (NSVM)

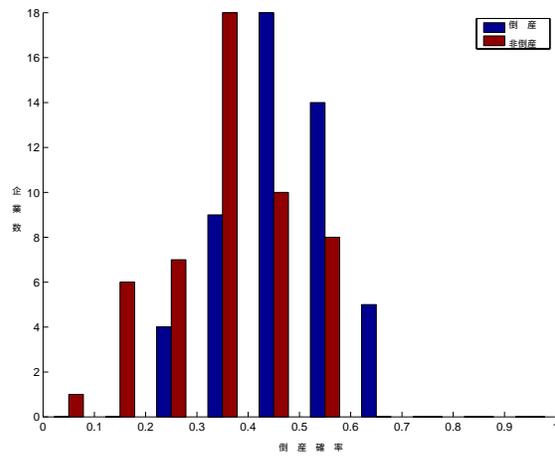


図 21: 識別データ 1 の分布 (LSVM)

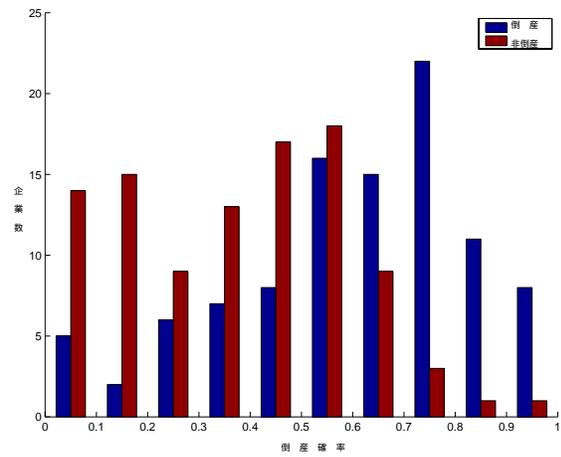


図 22: 識別データ 2 の分布 (LSVM)

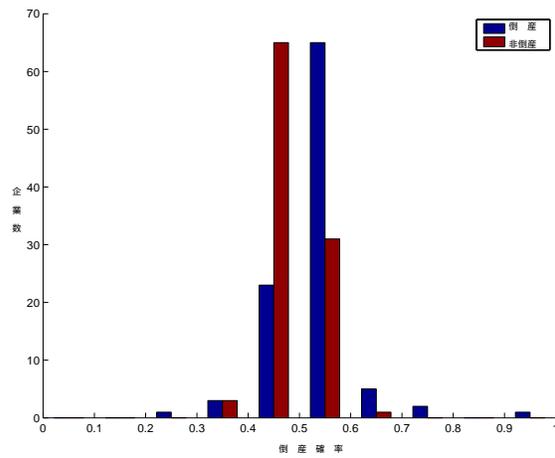


図 23: 識別データ 3 の分布 (LSVM)

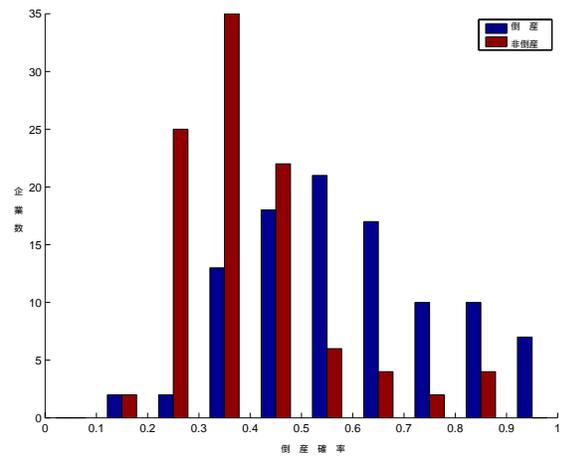


図 24: 識別データ 4 の分布 (LSVM)

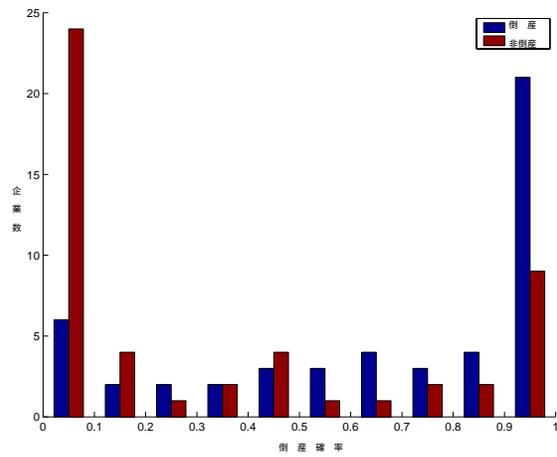


図 25: 識別データ 1 の分布 (判別分析)

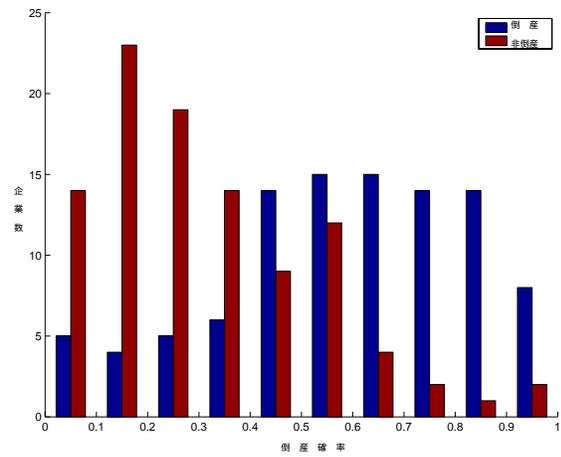


図 26: 識別データ 2 の分布 (判別分析)

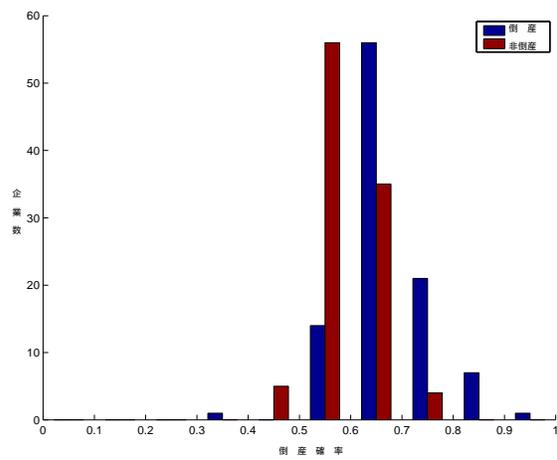


図 27: 識別データ 3 の分布 (判別分析)

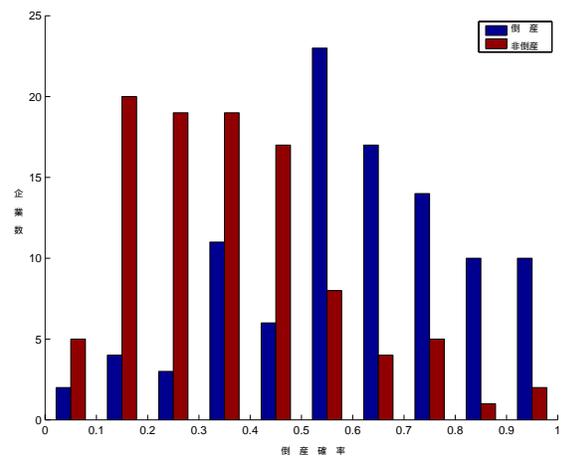


図 28: 識別データ 4 の分布 (判別分析)