

## 論 説

## 技術評価支援システムの試作

## — 識別関数の構成とオーバーフィッティング問題の解消 —

吉 田 要

## 目 次

1. はじめに
2. 評価因子
  - 2-1. 標準化競争
  - 2-2. 政治的影響
  - 2-3. 技術開発の動向
3. 記号と幾何計画問題の定義
4. 高次元化とオーバーフィッティング
  - 4-1. データクリーニングの例
  - 4-2. 高次元化
    - (1) 1次元の高次元化例
    - (2) オーバーフィッティング問題
5. 識別関数の構成と計算例
  - 5-1. GPPOSY
  - 5-2. データ変換ツール
  - 5-3. 計算例題
6. おわりに
7. 参考文献
8. 付録

## 1. はじめに

技術評価に関しては、種々な手法が考案されている。類似技術を市場から探して価値を査定したり、関係する技術の市場規模などでの評価、あるいは開発コスト、市場導入タイミングに基づく評価法などがある。競合他社などの分析手法である CTI（競争的技術開発分析：Competitive Technical Intelligence）におけるツールとして、CFTP（顧客視点の技術開発計画：Customer Focused Technology Planning）、研究開発プロジェクトを評価する DM（Decision Management）、など種々なものがある。<sup>7), 8), 9), 10)</sup>

技術開発には多大の投資を必要とするものが少なくないため、技術開発が成功した場合に得られる利益と失敗した場合の損失を適正に評価することが重要である。技術開発に伴うリスクを軽減するためには、技術的優位性、独創性、市場のニーズのみならず、政治的、国家的な圧力など、評価すべき要素は少なくない。定量的に評価する方法のひとつとして、評価すべき因子を数量化し、それらの指標を元に、成功事例と失敗事例から新たな事例の成否を判定する識別関数を構成することにより、技術を客観的に評価する手法の有力なものとして SVM (Support

Vector Machine)<sup>1), 12)</sup>がある。

SVM は、2 次計画法を利用して、識別関数を構成するものであるが、無限次元空間を含む高次元空間へデータを写像して、線形識別関数を求め逆写像したものが非線形識別関数になっている。このとき、計算上はカーネルトリックといわれるテクニックによって計算量が次元によって増加しないことが利点となり、強力な手法となっている。この手法では、識別に要するサポートベクター（分離超平面あるいは超曲面を構成するのに必要なベクトル）は、データの数に比べて空間の次元が高くなるに従って多く必要となる。往々データの大部分が識別関数を構成するために必要となることがある。その場合には、いわゆるオーバーフィッティングが生じる事が懸念される。新たなデータセットに対して、正しく推定できない可能性を高くし、識別関数の頑健性に問題が生じることになる。このことは、SVM に限定した話ではなく、識別関数が余りに高精度で実現される場合には、どの方法でも生じる可能性がある。このように過度に精度の高い識別関数は、新たに発生するデータに対して正しく推定する上で頑健性に乏しいといわざるを得ない。

識別関数は、利用する関数の形によって多様な関数を構成することが可能である。SVM では識別関数を構成する上で 2 次計画法を利用している。われわれは空間の次元を過度に高くしないという視点から、幾何計画法<sup>3), 13)</sup>を利用した方法を提案した。<sup>15)</sup>

本論文では、オーバーフィッティング問題について考察し、問題を解消するための一方法を提案する。与えられたデータを的確に識別し分類できることは識別関数として重要なことであるが、他のデータセットに対しても頑健性を維持するという点から、高次元化の程度をいかにして抑えるかについて指数関数の展開項を高次元化のベクトル要素として利用する方法を提案し、いくつかの計算例題で有効性を確かめた。

## 2. 評価因子

技術評価システムを構築する場合、意思決定レベルが同一に扱えない情報に対しては、意思決定の階層性を考慮したシステムの構築が必要である。一方、同一レベルでの評価において、多くのデータから評価する式を構成する場合でも、識別関数の頑健性やオーバーフィッティングなどの問題がある。データ（情報）の階層性が現れる場合を含めていくつか事例を紹介しておく。

### 2-1. 標準化競争

よく知られている事例として、Video Tape の標準化競争がある。1980 年から 1988 年にかけて、Sony を代表とするグループは Beta Max 方式を提案した。

高品質でハイスペックであり、業務用として多く採用されている。この方式で家庭用のビデ

オカメラと録画再生装置の開発もおこなわれた。一方、同じ時期に Panasonic グループは低価格で長時間録画を目指した VHS 方式のカメラと録画再生装置の開発を行い、熾烈な競争を行ってきた。技術的には、Beta 方式が優れているといわれていたが、VHS グループが勝利することになり、ベータ側のグループ各社と、商品を購入した顧客にとって望ましくない状況が発生した。

また、2002 年から 2007 年にかけて、Panasonic&Sony グループは Blue Ray 方式の DVD 技術の開発を行ってきた。対して、Toshiba グループは HD 方式の技術開発を進め、ビデオテープ戦争の再来かとも思われたが、発売が本格化する前に、Toshiba グループは撤退を表明した。大口ユーザーである、映画制作者のソフト作成が大きく影響をすることが明白になった。<sup>6)</sup> この 2 つケースは、国内メーカー同士の標準化競争であるが、技術の優劣以外の要素が強く反映されており、これらを評価因子に加える場合は、階層的な評価が必要と思われる。

また、2000 年に、三菱電機は新暗号方式（コード名 MISTY）を発表した<sup>5)</sup> MISTY は標準化のための国際会議（3rd Generation Partnership Project）に提案されたが、多くの国の特許に対する抵抗により標準化のためのプロセスが進みにくい状況であった。三菱電機は、Misty（コード名は KASUMI に変更された）を日本と EU での標準化とするために、特許を放棄した。特許による利益を敢えて放棄することで、先行開発企業としての技術力の優位性を得ることができた。これも、技術優位だけでは標準化しにくいという例であるが、技術開発そのものは成功している例である。

これらのケースは、技術的優劣によらないで、技術が評価された事例である。

## 2-2. 政治的影響

日の丸 OS と呼ばれた TRON（The Real-time Operating System Nucleus）<sup>11)</sup> は、マイクロチップからスーパーコンピュータまでをサポートする OS として開発され、その守備範囲は広くネットワークから、ビジネスや工業など多岐にわたる領域をサポートするものとして設計された。

この Tron プロジェクトには、日本の多くのメーカーが参画し、試作機も完成し、小中高の情報教育に導入する方針が発表された。その後、米国商務省から、ビジネス用 OS として開発された BTron が米国のスーパー 301 条により貿易障壁と認定されるという情報が入った。当時の日米の貿易不均衡も大きく作用したことは否めないとしても、Tron がオープンであることを考えると、その後の展開は大変残念であった。Windows 以外にも多くの OS があり、多様化している現状を見ると、もう少し粘り強い対応があったのではないかと考えられる。このケースでは、技術の評価状況が劇的に変化したように思われ、これらの因子を取り込むモデルが必要と思われる。

### 2-3. 技術開発の動向

昨今の、環境重視のもと各社はエコカーの開発を積極的に進めてきている。日本ではハイブリッドカーが普及しているが、EU ではディーゼルエンジン車が多く採用され、ハイブリッドカーが普及していない状況である。また、ここに来て、電気自動車各国で急ピッチで開発されてきており、従来の巨大メーカー以外の新興企業が開発に参入するなど競争が激しくなっている。この自動車に関する技術開発は競合他社に対して先行的に行われるべきであるが、その分リスクが大きくなる。技術開発の先行投資は市場の成熟時期や利用環境の整備などの見極めが重要な評価項目となる。

以上のケースからも分かるとおり、技術開発の成否は技術の優位性のみならず市場の規模あるいは技術が開発された結果の影響が大きい程、評価を行う上で困難な問題が生じてくる。

### 3. 記号と幾何計画問題の定義

便宜上、いくつかの記号を定義しておく。

$n$  次元列ベクトル  $\mathbf{a}$ ,  $\mathbf{x}$  を次のように書く。

$$\mathbf{a}, \mathbf{x} \in R^n \quad \mathbf{x} \equiv (x_1, \dots, x_n)^t, \mathbf{a} \equiv (a_1, \dots, a_n)^t$$

内積

$$\langle \mathbf{c}, \mathbf{x} \rangle$$

超平面  $H$

$$H = \{\mathbf{x} | \langle \mathbf{c}, \mathbf{x} \rangle = d_1, \mathbf{c}, \mathbf{x} \in R^n, d_1 \in R\}$$

複数の超平面に含まれるものを線形多様体という。

$n - 1$  次元の線形多様体を超平面という。

ベクトルが線形多様体の次元より 2 個以上多いとき

それらは一般の位置に無いということにする。

項を次のように表す。

$$\mathbf{x}^{\mathbf{a}} \equiv \prod_{i=1}^n x_i^{a_i}$$

また、 $n \times m$  行列  $A_M$  を

$$A_M \equiv (\mathbf{a}_1, \dots, \mathbf{a}_m) \in R^{n \times m}$$

で表し、多項式を

$$\mathbf{c} \in R^m, \mathbf{x}^{A_M} = (\mathbf{x}^{a_1}, \dots, \mathbf{x}^{a_m})^t$$

$$\langle \mathbf{c}, \mathbf{x}^{A_M} \rangle \equiv \sum_{i \in M} c_i \mathbf{x}^{a_i}$$

と表す。また、多項式関数  $G_k$  を

$$G_k(\mathbf{c}_k, \mathbf{x}, A_{P(k)}) \equiv G_k(\mathbf{x}) \equiv \langle \mathbf{c}_k, \mathbf{x}^{A_{P(k)}} \rangle$$

と書く。

ここで、 $\mathbf{c}_k \in R_+^{p(k)}$  のとき  $G_k$  を正多項式 (Posynomial) という。  
この多項式を使って、幾何計画法  $GP$  は次のように定義できる。

[Posynomial  $GP$ ]

$$\begin{aligned} \min \quad & G_0(\mathbf{x}) \\ \mathbf{x} \in \quad & R_+^n \\ \text{sub.to} \quad & G_k(\mathbf{x}) \leq 1 \quad k \in Q \equiv \{1, \dots, q\} \end{aligned}$$

この問題の双対問題は次の形式で書かれる。

[Dual Problem]

$$\begin{aligned} \max \quad & v(\boldsymbol{\delta}) = \prod_{k \in Q \cup \{0\}} \left\{ \prod_{i \in P(k)} \left( \frac{c_{ki}}{\delta_{ki}} \right)^{\delta_{ki}} \lambda_k^{\lambda_k} \right\} \\ \text{sub.to} \quad & \end{aligned}$$

$$A\boldsymbol{\delta} = \mathbf{0} \quad \text{直交条件}$$

$$\lambda_0 = \sum_{i \in P(0)} \delta_{oi} = 1 \quad \text{正規化条件}$$

$$\boldsymbol{\delta} \in R_{\geq 0}^{\#A}, \quad \boldsymbol{\delta} \neq \mathbf{0}$$

$$\text{ただし, } \lambda_k = \sum_{i \in P(k)} \delta_{ki}$$

$$A \equiv [A_{P(0)}, \dots, A_{P(q)}] \text{ 指数行列, } \#A = \sum_{i=0}^q p(i)$$

難度 (Degree of Difficulty)

$$\begin{aligned} \text{d.d.} &\equiv \sum_{i \in \{0\} \cup Q} p(i) - (n+1) \\ &= \{GP \text{ に含まれる項の総数} \} - \{ \text{ベクトル } \mathbf{x} \text{ の次元数} + 1 \} \end{aligned}$$

#### 4. 高次元化とオーバーフィッティング

プロジェクトや技術を評価する為に、成功事例や失敗事例を学習することで、それらを識別する為の識別関数の構成について考察をする。SVM (Support Vector Machine) は、パターンを識別する為に、得られているデータの属しているベクトル空間から高次元空間（無限次元空間を含む）へ写像することで、識別可能性が飛躍的に上がり、大変強力なツールとなっている。しかしながら、識別関数の頑健性の観点からは必ずしも望ましいとはいえない。与えられたデー

タが総て正しいものであるとしても、それらを完全に分類することで、非常に特殊な事例の分類がなされることがあり、その識別関数を利用することは、特殊な事例を含まないような一般的なケースにおいては寧ろ識別性の観点からみて逆効果になることも予想される。特殊と思えるデータが誤りであるということも考えられるが、事前に判定できないような場合には正しいものとして使わねばならない。識別関数の信頼性の上から、前処理として、ノイズなど不適切と思われるデータと共に特殊な事例も削除することが望ましいと考えられる。

#### 4-1. データクリーニングの例

データクリーニングについて、簡単な事例について説明をしたい。

与えられたデータを処理する上で、あらかじめデータからノイズや特異例を除くことは識別関数の頑健性を得るために重要である。

図 1 は付録アイリスデータ<sup>4)</sup>のセトナの散布図である。グラフからわかるようにあるデータは他のデータの傾向と異なっており、データの計測エラーもしくは特異例と考えることができる。他の表示方式でも特異例であれば除く対象となる可能性が高い。他の表示方式の一つとして、全種類を 3 次関数で近似し、係数の散布図を示す。データは次の 4 項目からなっている。

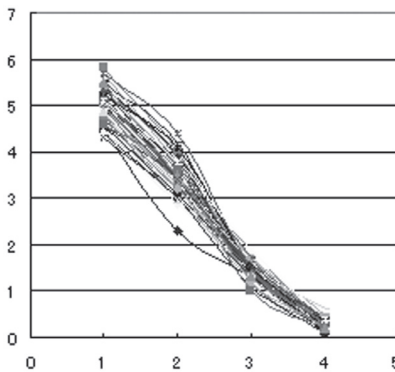


図 1 セトナの散布図

がく片      がく片幅      花びら      花びら幅

$y_1$                        $y_2$                        $y_3$                        $y_4$

$y_i$  は関数の  $x_i$   $i = 1, 2, 3, 4$  での値としてみる。すなわち、

$$y_i = f(x_i)$$

$$f(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

簡単のために、 $x_i = i$  とし、関数を特徴付ける係数は 4 個ある。

これを 次のベクトル  $f$  と行列  $D$  で表記すると、事例  $k$  の係数は次のようになる

$$a_{ki} \quad i = 3, 2, 1, 0$$

$$f_k(4) = a_{k3}4^3 + a_{k2}4^2 + a_{k1}4 + a_{k0}$$

$$f_k(3) = a_{k3}3^3 + a_{k2}3^2 + a_{k1}3 + a_{k0}$$

$$f_k(2) = a_{k3}2^3 + a_{k2}2^2 + a_{k1}2 + a_{k0}$$

$$f_k(1) = a_{k3}1^3 + a_{k2}1^2 + a_{k1}1 + a_{k0}$$

$$f_k = (f_k(1), f_k(2), f_k(3), f_k(4))^t$$

$$D = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \\ 64 & 16 & 4 & 1 \end{pmatrix}$$

$$\mathbf{a}_k = (a_{k3}, a_{k2}, a_{k1}, a_{k0})^t$$

$$\mathbf{f}_k = D \cdot \mathbf{a}_k$$

より

$$\mathbf{a}_k = D^{-1} \mathbf{f}_k$$

$$A = (\mathbf{a}_1, \dots, \mathbf{a}_k, \dots, \mathbf{a}_m)$$

$$F = (f_1, \dots, f_k, \dots, f_m)$$

$$A = D^{-1} F$$

となる。50個ずつ3種類のデータ(セトナ, パーシクル, バージニカ)についてグラフを見てみると、パーシクルとバージニカ間はセトナに対してよりも区別しにくいことが分かる。

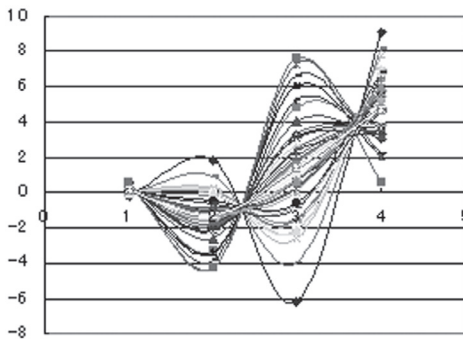


図2 セトナの3次関数の係数の散布図

図2はセトナの3次関数の係数の散布図である。やはり、特殊な例が見られる。このデータがノイズを含むものかどうかは現時点で確定できないので、正しいとすれば特殊な事例で、度々は起こらないものと考えることができる。一般的な事例を扱うと考えれば、これを除いてよいであろう。図3, 4はパーシクルとバージニカの散布図, 図5, 6は3次関数の係数の散布図である。同様な傾向が見られ、識別しにくいように見える。

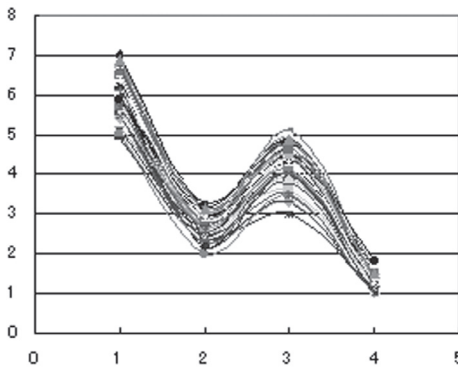


図3 パーシクルの散布図

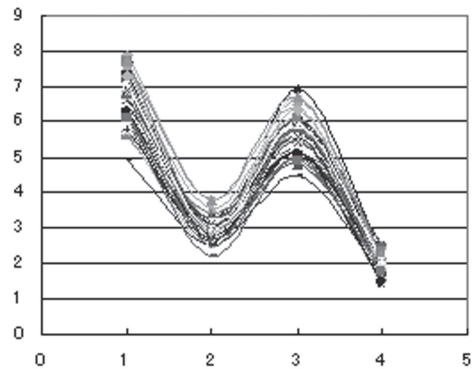


図4 バージニカの散布図

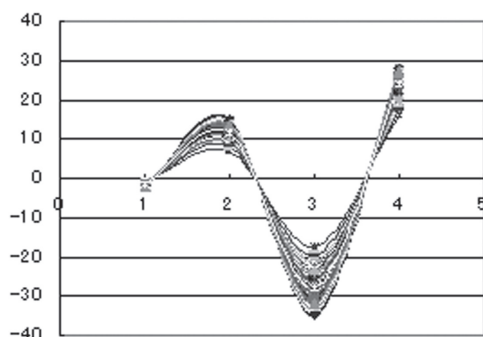


図 5 パーシクルの 3 次関数の係数の散布図

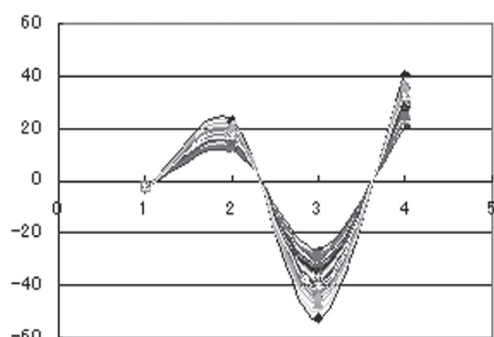


図 6 パージニカの 3 次関数の係数の散布図

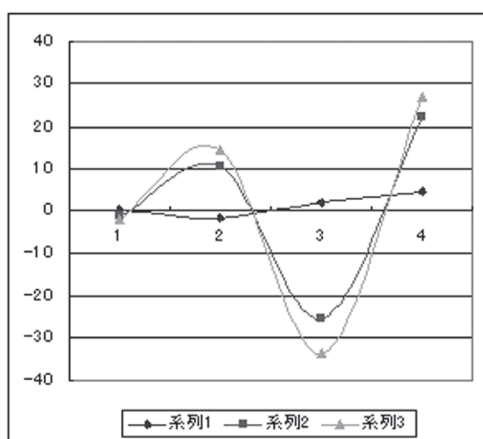


図 7 3 次関数のモデル係数

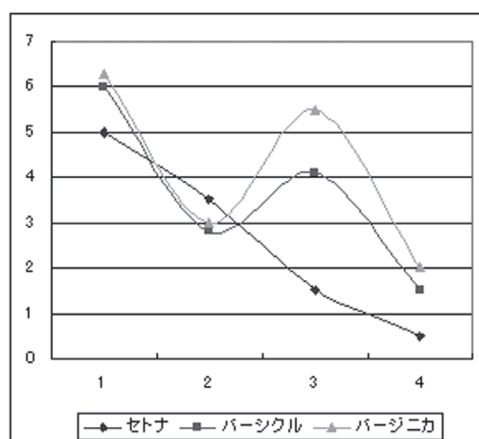


図 8 モデル散布図

これらの種類ごとのデータに対して、最小二乗法による 3 次関数のモデル式の係数を元にした散布図である。グラフは図 7, 8 のようになる。このグラフからのずれによって、判別すべきデータについて、幾分の情報を得ることができる。このモデル式では、パーシクルとパージニカの特徴の違いは比較的に見やすい。

また、このようにデータを関数で表して、それらを解析することで、特徴を調べることは興味深い今後の検討課題としたい。

#### 4-2. 高次元化

高次元化するには種々の方法が考えられるが、データの特性から高次元化のための写像関数を決定することが望ましい。ここでは、簡単な事例で考察しておく。

ベクトル  $a \in R^n$  に対して、

写像  $\phi \in R^s$  によって

高次元化したベクトル  $\phi(a)$  を次の形で与える。



$$\phi(a) = (\phi_1(a), \dots, \phi_s(a))^t$$

すると、与えられたデータセットからなる指数行列  $A$  は高次元化された結果

$$\text{ex}A \equiv \begin{pmatrix} a_{01} & \dots & a_{pq} \\ \phi(a_{01}) & \dots & \phi(a_{pq}) \end{pmatrix}$$

となる。この  $\text{ex}A$  は双対問題の指数行列を構成する。

以下、もとの空間では識別困難な事例でも、高次元化することで、識別可能になることを簡単な事例で説明しておく。

### (1) 1次元の高次元化例

1次元上に、3個の点（白丸と黒丸）があり、図9のように与えられたとすると、1個の超平面（0次元なので点になる）で、これらを2グループに分けることはできない。



図9 1次元上の3個のデータ

一般に、 $n$ 次元空間において、 $n+1$ 個のデータが一般の位置にあれば、任意に与えられても、超平面によって2分割することが出来る。図9の例では、データが3個あるので、2次元空間に写像すると1個の超平面で分離できることが期待できる。高次元化のための写像  $\phi$  の作り方は種々なものが考えられるが、ここでは簡単に、2次関数を利用する。

$$\phi(a) = (a, a^2)^t$$

この写像で2次元空間へ写像すると、図10のように写像され、1個の超平面（直線）で分離できる。

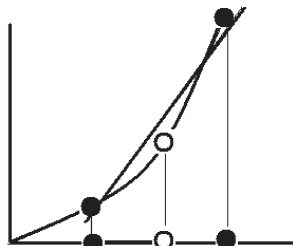


図10 2次元空間への写像

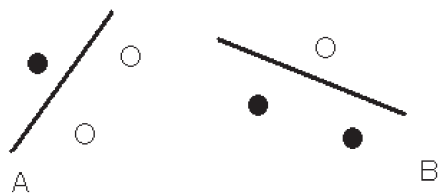


図11 2次元面内での3個のデータ

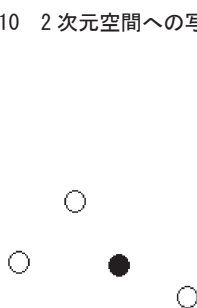


図12 2次元面内での4個のデータ

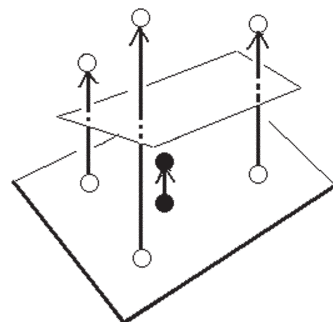


図13 3次元への写像

同様に、図 11 のように、2 次元空間でもデータ 3 個が A, B のケースのように、一般の位置に配置されれば超平面(直線)を適当に引くことによって分離することができる。しかし、図 12 に見るように、4 個の点は 1 個の超平面では分離できない。この場合でも、図 13 に見るとおり、3 次元空間に写像することで分離できる。このことは、与えられたデータを、高次元空間に写像することで、識別関数の非線形性を緩めることができる。全てのデータに対して同一の仕方で写像する方法が必要である。また、逆写像が構成できるようであれば、元の空間で非線形識別となっている。

## (2) オーバーフィッティング問題

このように、写像先の空間の次元を上げることによって、線形識別可能性は高くなる。しかし、非線形性をゆるめるためにデータを高次元化すると、元の空間ではオーバーフィッティング問題が生じてくるので、高次元化は十分注意が必要である。

例えば、図 14 でポイント A ○ を正しく識別することで、識別関数は曲線のようにになる。しかし、ポイント A はデータの誤りである可能性もある。その場合は削除すべきものである。仮にポイント A が正しいデータであったとしても、ほとんど起こり得ないような値であれば、○データが●の近くに現れたときには識別上の誤判定を起こす可能性が出てくる。これは、与えられたデータセットに対して必要以上にフィットしているいわゆるオーバーフィッティングが起こっているからと考えることができる。このようなケースには、ポイント A を●判定とするような識別関数が、その他のデータを正しく判定する上で有効と考えられる。少なくとも、4-1 のデータクリーニングによって、事前にデータの状態を調べておいて、特性から著しく離れたデータは、削除して識別関数を構成することは識別関数の頑健性 (robustness) を得る上で必要なことである。

このことをもう少し詳しく見てみる。

正事例と負事例を識別する関数が、図 15 の曲線のようにであったとしたら、識別はこのデータに関して完全に分類している。しかし、白丸のグループに割り込む形の黒丸は果たして黒グループとして識別することが妥当であろうか。もし、その黒丸がなければ、識別関数はよりシンプルな点線になる。

さらに、図 11 に見るように新しく識別しなければならないデータ△が追加されたとき、識別関数がその△を誤判定することがあっても、一

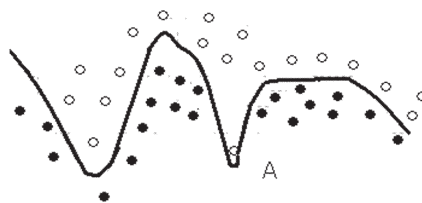


図 14 オーバーフィッティング

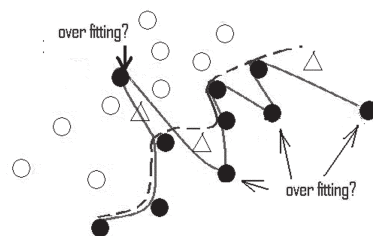


図 15 識別関数の妥当性

般的なデータではこの識別関数はすぐれたものである可能性が捨てきれない。

以上から、識別関数の持つべき特徴を出来る限り追求することが重要であるが、十分な情報がない時点では無理な要求であるといわねばならない。そこで、ひとつの方法は、写像先の空間を、低次元の空間で分離する写像を考えることで、このような問題が生ずるのを避けることを検討する。

$\phi$ として、データの第1要素を指数関数の級数展開の第2項以降を用いて、高次元化する。高次元化することで、頑健性がどのようになるかを識別関数のサポートベクターの個数を用いて評価する。他の関数も可能であるが、どのように選ぶのが適切かについての選択基準は今後の課題としたい。

## 5. 識別関数の構成と計算例

以下、計算例題にて、高次元化によって識別関数の構成が容易となるにつれサポートベクターの数（オーバーフィッティング程度）が増加することを示す。

### 5-1. GPPOSY<sup>2)</sup>

幾何計画問題を解くために、ピッツバーグ大学で開発されたプログラム GPPOSY を利用する。この GPPOSY で入力するデータは図 12 の形式でなければならない。一方、必要なデータが、エクセル形式で提供されることが多い。また、本システムは主として prolog を用いて記述している。これらの便宜上、データの変換ツールを作成した。まず、CSV データから prolog データに変換するツール csv.exe を作成した。さらに、高次元化プログラム updim.pl を作成し、高次元化した prolog データを GPPOSY 用データに変換するツール pl2gp.pl を作成した。これらを利用することで、csv データを効率よく GPPOSY の入力データに変換することができる。

### 5-2. データ変換ツール

GPPOSY のデータファイルは、次の形式をしている。

ex1.dat	5 4
	1 2 1 1 1 1
変数の数 制約式の数	1 1
	1 -1
各多項式に含まれる項数の列	2 1
	1 1
各項に含まれる変数の数 係数	2 1
	2 1
各変数の番号 次数	2 -1
	3 1
:	.....

図 12 GPPOSY の入力データ形式

プログラムの実行課程は例題 3 にて説明する。

以上の準備の下、いくつかの例題について識別関数を求めてみる。

### 5-3. 計算例題

#### 例題 1

文献 15 では、Fisher の iris データでセトナとバーシクルの識別を行った。図 2, 3, 4, 5 から分かるように、セトナとは異なり、バーシクルとバージニカは識別しにくいように思われる。

データはバーシクルとバージニカともに 50 個ずつである。この中から学習データを 25 個ずつ使い、識別関数を構成し他のデータをテストデータとして検証する。

この場合、データは 4 次元で識別対象のデータの個数は 50 個なので、49 次元まで高次元化すれば、線形識別できることが考えられる。しかし、高次元化すると、現在のデータセットに対して、サポートベクターは 49 個近くまで出現する可能性があり、オーバーフィッティングになると考えられる。識別関数の頑健性を得るためには、次元を低く抑えることが望ましい。そこで、データの次元を上げないで、識別能力がどのくらいあるかをまず検討する。4 次元データに対する幾何計画問題を設定し、最適化する。詳しくは付録参照

最適解から得られる識別関数は次のようになった。

$$\ln(\phi(\alpha, \beta, \gamma, \delta)) = -1.12998\alpha - 2.61935\beta + 6.907755\gamma \\ + 6.907755\delta - 30.6426$$

この識別関数による判定結果は、100 個のデータに対して、正しく識別したものは 98 個であり、識別能力は十分であり高次元化は必要ないと考えられる。また、サポートベクターは付録から 3 個と分かる。識別関数の頑健性は高いと考えられる。

次の例は 1 次元での識別である。1 次元のままでは識別不可能なので、3 点を 2 次元に写像して幾何計画問題を設定する。

#### 例題 2 図 9, 10

クラス A 1, 3

クラス B 2

この 1 次元問題を写像すること次の式で写像して、2 次元問題に変換する。

$$\Psi(\alpha) = (\alpha, \alpha^2)^t$$

この変換によって、問題は次のような最適化問題として設定することができる。

$$\begin{aligned} \min \quad & s^{-1} \\ \text{s.t.} \quad & \\ & k_1^{-1}s + k_1^{-1}k_2 \leq 1 \end{aligned}$$

$$\begin{aligned}
 k_1 x^{-1} y^{-1} &\leq 1 \\
 k_1 x^{-3} y^{-9} &\leq 1 \\
 k_2^{-1} x^2 y^4 &\leq 1
 \end{aligned}$$

この問題を GPPOSY によって解きだした結果は次のようになる。

PRIMAL VARIABLES ARE

$I$	$X(I)$	
$s$	1 0.10546875	$xy = 0.42198008497651 \geq k_1$
$k_1$	2 0.42198009	$x^3 y^9 = 0.42197999694615 \geq k_1$
$k_2$	3 0.31651135	$x^2 y^4 = 0.316511318016632 \geq k_2$
$x$	4 0.31651135	$k_1 - k_2 = 0.10546874 \geq s$
$y$	5 1.3332226	

これらの値をもとに、制約式の値を求めると、分離されていることが確かめられる。

このとき、1, 2, 3 で構成される全てのベクトルが、サポートベクターとなっていることは、最適解は制約式の境界値で得られていることから分かる。また、識別関数は

$$\phi(\alpha) = x^\alpha y^{\alpha^2} - \frac{(k_1 + k_2)}{2}$$

となる。この値が正ならば A 負ならば B である。この例では、頑健性は低い。

さらに、より点を多くした場合について扱ってみる。

### 例題 3

Class A 1, 2, 5, 6, 9, 10

Class B 3, 4, 7, 8

この例題も、9 次元まで高次元化すると、線形識別できる。ここでは、できるだけ低次元と

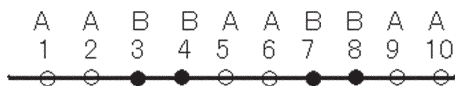


図 16 1 次元データ上の 10 個のデータ

いうことで、4 次元空間で扱う。

また、データ変換ツールについて説明する。

写像関数は、指数関数の 2 次項以降を用いると、

$$\Psi(\alpha) = \left( \alpha, \frac{\alpha^2}{2!}, \frac{\alpha^3}{3!}, \frac{\alpha^4}{4!} \right)^t = (\alpha, \beta, \gamma, \delta)^t$$

となる。この写像によって、データを 4 次元空間に写像する。GP 問題は次のように設定できる。

ここで、 $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$  および分類されるべきクラスは、以下のテーブルのとおりである。

$$\min s^{-1}$$

$$\text{s.t.}$$

$$k_1^{-1}s + k_1^{-1}k_2 \leq 1$$

$$k_1 x^{-\alpha_i} y^{-\beta_i} z^{-\gamma_i} u^{-\delta_i} \leq 1 \quad \alpha_i \in A$$

$$k_2^{-1} x^{\alpha_j} y^{\beta_j} z^{\gamma_j} u^{\delta_j} \leq 1 \quad \alpha_j \in B$$

表 1 例題 3 のパラメータと分類クラス

$\alpha$	$\beta$	$\gamma$	$\delta$	class
1	0.5	-1.66667	0.041666667	A
2	2	1.333333	0.666666667	A
3	4.5	4.5	3.375	B
4	8	10.66667	10.66666667	B
5	12.5	20.83333	26.04166667	A
6	18	36	54	A
7	24.5	57.16667	100.0416667	B
8	32	85.33333	170.6666667	B
9	40.5	121.5	273.375	A
10	50	166.6667	416.6666667	A

Prolog を実行するために mjlist による実行状況を表示して説明する。図 17 では例題 3 のデータを登録したもので、1 次元の csv データである。そのデータに対して prolog の宣言データ vec を作成するプログラムの csv.pl の画面 (図 18) である。この部分は実行型として csv.exe としても作成している。実行して得られるデータが pldata.pl が図 19 である。

さらに、図 20 の updim.pl の高次元化プログラムを実行する。その結果が図 22 に



図 17 csv データ

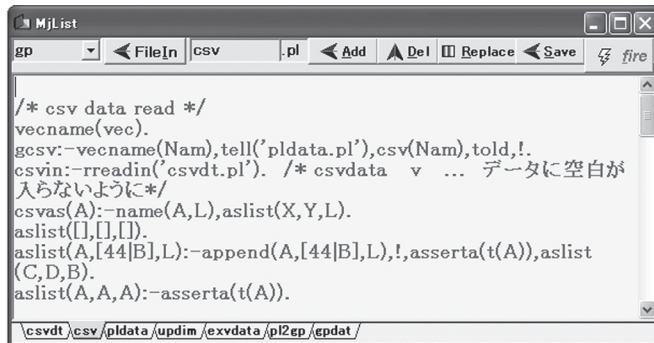


図 18 csv.pl (csv.exe)

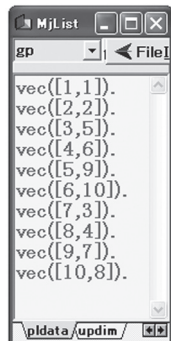


図 19 prolog データ

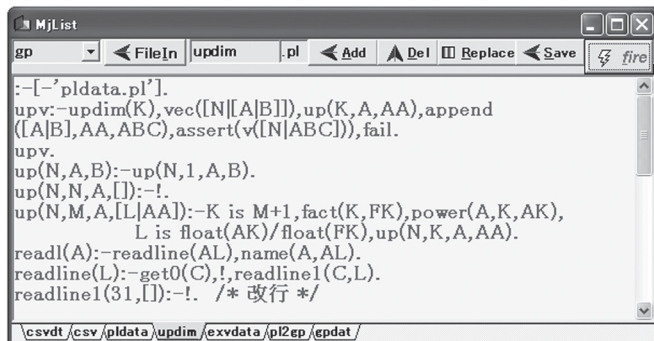


図 20 プログラム updim.pl

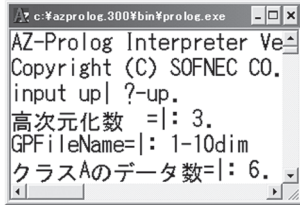


図 21 updim の実行

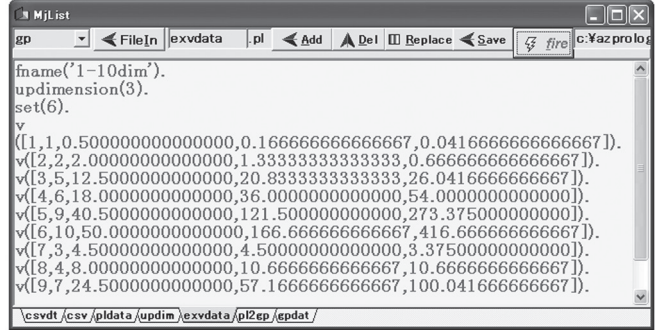


図 22 高次元化されたデータ

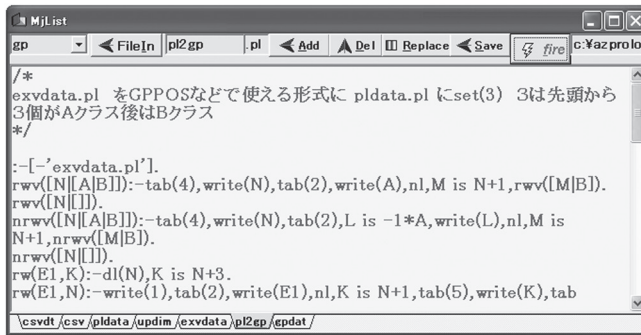


図 23 pl2gp.pl

exvdata.pl に表示している。

この図 22 の prolog データに対して、GPPOSY 用ヘッダーを変換する。図 23 はデータを変換するプログラムである。図 24 は実行結果である。

GPPOSY を実行して得られた最適解から、識別関数  $\phi$  を構成する。

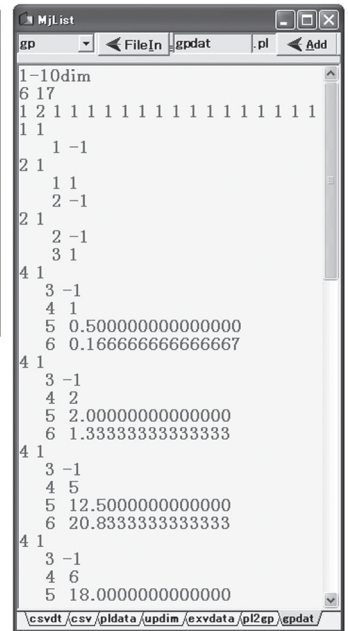


図 24 GPdata

実行結果

```
s    0-1.3725622E-01
k1   0.37515629      k2   0.36143067
x    0.38291043      z    0.78641739
y    1.8495826       u    1.0446663
```

$$\phi(\alpha) = x^\alpha y^{\frac{\alpha^2}{2!}} z^{\frac{\alpha^3}{3!}} u^{\frac{\alpha^4}{4!}} - \frac{(k_1+k_2)}{2} = x^\alpha y^\beta z^\gamma u^\delta - \frac{(k_1+k_2)}{2}$$



この識別関数をもとに、写像先で分離できていることを、確認しておく。

写像されたデータは次のようになる。

$\phi(x) : x \in \{1, 2, 5, 6, 9, 10\}$  は  $k_1$  以上。

$x \in \{3, 4, 7, 8\}$  は  $k_2$  以下になっており、識別できていることがわかる。

識別関数による計算をもとに判定をしたクラスを表に示す。値は定数項を除いた数値である。

表 2 例題 3 の判定結果

1	0.501226	A	3	0.351246	B	5	0.375165	A	7	0.36172	B	9	0.375155	A
2	0.374856	A	4	0.36172	B	6	0.375156	A	8	0.351326	B	10	0.502154	A

例題 3 は高次元化することで、識別可能になることが分かる。このときのサポートベクターを構成するのは、境界値を取る点である A の 5, 6, 9 と、B の 4, 7 の 5 点である。

以上を整理すると、例題 1 では、4 次元ベクトルを高次元化しなくても 100 個のデータに対して 98% の識別率であり、サポートベクターは 4 個（付録参照）なのでこのデータセットに対する依存性は高くなく、識別関数の頑健性は高いと考えられる。例題 2 では、1 次元データ 3 個を 2 次元空間に写像して識別したが、サポートベクターは 3 個あり、他のデータセットを識別する頑健性は高くないと考えられる。例題 3 は 1 次元データが 10 個あり、9 次元にあげると識別可能になるが例題 2 と同様、サポートベクターは 9 個以上になり、オーバーフィッティングになる可能性が高く、頑健性が脆弱になると考えられる。しかし、より低次元の 4 次元空間でも識別可能であり、この場合はサポートベクターは 5 個でオーバーフィッティングはそれほど生じていないと考えられる。

以上の例から、分かるように高次元化は識別可能な関数の存在可能性を高めるが、行き過ぎた高次元化はオーバーフィッティングを誘導し、識別関数の頑健性を損なうことが分かる。

## 6. おわりに

本論文では、識別関数のオーバーフィッティング問題の解消について考察し、問題解消のための方法を提案した。また、その方法の有効性を計算例題によって確かめた。ついで、GPPOSY を利用するために、データ変換用のツール (csv.pl, updim.pl, pl2gp.pl) を作成した。残された問題として識別関数を単項式ではなく、識別能力が高いと思われる多項式で構成することや、評価のための要素が同一レベルでないときの扱いを含めた識別関数の構成などがある。

## 7. 参考文献

- [1] Burgs, C.: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2, pp.121-167(1998)
- [2] Cenk Tunasar & Jayant Rajgopal :



User's Manual for GPGLP - A Posynomial Geometric Programming Solver, *Technical Report No. TR95-11*

- [ 3 ] Duffin, R. J., Peterson, E.L. and Zener, C.:  
Geometric Programming Theory and Application; Jhon Wiley & Sons(1967)
- [ 4 ] Fisher, R. A.: アイリスデータ 例えば  
[http://http://www.sci.kagoshima-u.ac.jp/~drs/ja/data/fishers\\_iris/](http://http://www.sci.kagoshima-u.ac.jp/~drs/ja/data/fishers_iris/)
- [ 5 ] 現行携帯電話（GSM）の標準暗号に三菱電機の MISTY 技術採用  
開発 No.0223 <http://www.mitsubishielectric.co.jp/news/2002/0711.htm>
- [ 6 ] 岩本敏裕：DVD 産業における技術革新と競争戦略  
—21 世紀初頭における日本の家庭用録画・再生機器 1) 産業—  
『立命館経営学』2008 第 47 巻 第 4 号 pp.279~302
- [ 7 ] 計測と制御 VOL.43 NO.10 2004 計測自動制御学会誌
- [ 8 ] 真柄睦「CI に基づく技術評価」742,747 文献 7 所収
- [ 9 ] 大澤良隆, 油本暢勇「産業界における研究開発プロジェクト評価の一手法」  
748,754 文献 7 所収
- [10] 高森年「産業に役立つ技術評価の方法について」725,729 文献 7 所収
- [11] トロンについては、トロン協会の資料「トロン沿革」に詳しい  
<http://www.assoc.tron.org/jpn/intro/enkaku.html>
- [12] 山口貴大「サポートベクターマシンによる倒産予測」2001  
<http://www-optima.amp.i.kyoto-u.ac.jp/research/paper/grad/yamaguchi.pdf>  
2005 年 7 月アクセス
- [13] 吉田要：幾何計画法の識別関数の構成への応用,  
平成 16 年度三重地区計測制御研究会講演論文集,2004
- [14] 吉田要：可変係数付き幾何計画法問題の近似最適化法；計測自動制御学会,  
Vol.26, No.8, pp.~00--99 (1990)
- [15] 吉田要「技術評価支援システムの試作にむけて—識別関数の構成—  
立命館経営学 Vol.44 No.4

## 8. 付 録

### 1. Exe ファイルの作成

Azprolog のコマンド azpc で c ソースを生成し、MSC でコンパイルし、exe ファイルを作る。

Azpc -p csv.pl /e csv

なるコマンドで実行される。Set(6) の 6 は先頭から 6 個が A クラス後は B クラスの指定

### 2. csv.pl

vecname(v).

gcsv:-vecname(Nam),tell('pldata.pl'),csv(Nam),told,!.

csvin:-rreadin('csvdt.pl'). /\* csvdata v ... データに空白が入らないように \*/

csvas(A):-name(A,L),aslist(X,Y,L).

aslist([],[],[]).

aslist(A,[44|B],L):-append(A,[44|B],L),!,asserta(t(A)),aslist(C,D,B).

```

aslist(A,A,A):-asserta(t(A)).
append([],A,A).
append([A | B],C,[A | D]):-append(B,C,D).
asts:-term(A),retract(term(A)),retract(csvn(N)),M is N+1,assert(csvn(M)),csvgas(A),
gather(t,X),abolish(t,1),assert(ts([M | X])),fail.
asts.
/* gather(term,list) */
gather1(X,R,RR):-Y=..[X,B],retract(Y),asserta(xX(B)),gather1(X,[B | R],RR).
gather1(X,R,R):-retract(xX(B)),Y=..[X,B],assert(Y),fail,true.
gather(X,A):-gather1(X,[],A),!.
pts([N | A],[N | B]):-pts1(A,B).
pts1([A | B],[C | D]):-name(A,C),pts1(B,D).
pts1([],[]):-!.
tms(A):-ts(B),pts(A,B).
csvg(T):-csvgin,assert(csvn(0)),asts,tms(A),X=..[T,A],assert(X),fail.
csvg(T):-retract(csvn(Z)),listing(T),!.
/* file から term の読み込み */
ast("):-!.
ast(A):-name(A,[31]),!.
ast(A):-asserta(d(A)).
readin(A):-get0(W),readline0(W,A).
readline0(26,"):-!.
readline0(W,A):-readline1(W,L),name(A,L),!.
readline(L):-get0(C),!,readline1(C,L).
readline1(31,[]):-!. /* 改行 */
readline1(26,[]):-!. /* ファイルの最後 */
readline1(33090,[33090]):-!. /* 。 */
/*readline1(C,[C | L]):-get0(A),readline1(A,L).*/
readline1(C,[C | L]):-get0(G),readline1(G,L).
rreadin(F):-see(F),repeat,readin(A),assert(term(A)),ast(A),A=",!
    retract(term(")),seen.
top_level:-gcsv,halt.

```

## 3. updim.pl

```

:-['pldata.pl'].

upv:-updim(K),vec([N | [A | B]]),up(K,A,AA),append([A | B],AA,ABC),assert(v([N | ABC])),fa
il.

upv.

up(N,A,B):-up(N,1,A,B).

up(N,N,A,[]):-!.

up(N,M,A,[L | AA]):-K is M+1,fact(K,FK),power(A,K,AK),
    L is float(AK)/float(FK),up(N,K,A,AA).

readl(A):-readline(AL),name(A,AL).

readline(L):-get0(C),!,readline1(C,L).

readline1(31,[]):-!. /* 改行 */

readline1(C,[C | L]):-get0(A),readline1(A,L).

power(X,1,X):-!.

power(X,M,XM):-K is M-1,power(X,K,XK),XM is float(X)*XK.

fact(1,1):-!.

fact(N,FN):-K is N-1,fact(K,FK),FN is N * FK.

append([],A,A).

append([A | B],C,[A | D]):-append(B,C,D).

check(0).

check(N):-K is N+1,asserta(updim(K)),write('GPFileName='),readl(FName),asserta(fname(
FName)),
    write(' クラス A のデータ数 ='),read(AN),
    asserta(set(AN)),tell('exvdata.pl'),upv,listing(fname),updim(L),LL is
L-1,write(updimension(LL)),
    write('.'),nl,listing(set),listing(v),told.

up:-write(' 高次元化数  ='),read(N),check(N),halt.

:-write('input up').

```

## 4. pl2gp.pl gp ファイルを作成する。

```

/* pldata.pl を GPPOSY などので使える形式 gpdata.dat に変換。 */

:-['pldata.pl'].

rwv([N | [A | B]]):-tab(4),write(N),tab(2),write(A),nl,M is N+1,rwv([M | B]).

```

```

rwv([N | []]).
nrwv([N | [A | B]]):-tab(4),write(N),tab(2),L is -1*A,write(L),nl,M is N+1,nrwv([M | B]).
nrwv([N | []]).
rw(E1,K):-dl(N),K is N+3.
rw(E1,N):-write(1),tab(2),write(E1),nl,K is N+1,tab(5),write(K),tab(2),write(1),nl,rw(E1,K).
nrw(E2,K):-dl(N),K is N+3.
nrw(E2,N):-write(1),tab(2),write(E2),nl,K is N+1, tab(5),write(K),tab(2),write(-1),
nl,nrw(E2,K).
rwh(0).
rwh(N):-write(1),tab(1),M is N-1,rwh(M).
st:-fname(Nam),write(Nam),nl,cnt,dl,set(K),cn(CN),dl(No),MM is No+3,asserta(rw(MM)),a
serta(nrw(MM)),
    VN is No+3,write(VN),tab(1),CN1 is CN+1+2*No,write(CN1),nl,
    write(1),tab(1),write(2),tab(1),CN2 is CN+2*No,rwh(CN2),nl,initial,
    dataList(K,No),bound(0.001,0.001),write(0),nl,write(0),nl.
dataList(K,No):-v([N | DL]),ND is No+1,pp(K,N,ND,DL),fail.
initial:-write(1),tab(1),write(1),nl,tab(5),write(1),tab(1),write(-1),nl,
    write(2),tab(1),write(1),nl,tab(5),write(1),tab(1),write(1),nl,
        tab(5),write(2),tab(1),write(-1),nl,
    write(2),tab(1),write(1),nl,tab(5),write(2),tab(1),write(-1),nl,
        tab(5),write(3),tab(1),write(1),nl.
pp(K,N,ND,DL):-N=<K,!,write(ND),tab(1),write(1),nl,tab(4),write(3),tab(1),write(-1),nl,
    rwv([4 | DL]).
pp(K,N,ND,DL):-write(ND),tab(1),write(1),nl,tab(4),write(2),tab(1),write(1),nl,nrwv([4 |
DL]).
dataList(K,No).
bound(E1,E2):-rw(E1,3),nl,nrw(E2,3).
cnt:-abolish(cnt,1),assert(cnt(0)),v(Z),cn(N),retract(cn(N)),M is N+1,assert(cnt(M)),fail.
cnt.
dl:-v(A),length(A,M),N is M-1,assert(dl(N)).
:-tell('gpdat.pl'),st,told,halt.:-['pldata.pl'].

```

## 5. 例題 1 の計算結果

Iris25-25 に対して，幾何計画問題を設定し解いた結果である。この問題は 4 次元空間の問題での識別問題で，難度 55 の問題として設定できる。得た結果からの識別関数による識別は正答率 98%であり，十分であるといえる。

実行結果

vv25.out

RUN DATE (M/D/Y) : 3/04/2010

## SUMMARY OF PROBLEM DATA

PROBLEM NAME: v25-25

NUMBER OF VARIABLES 7

NUMBER OF CONSTRAINTS 59

TOTAL NUMBER OF TERMS 61

DEGREES OF DIFFICULTY 53

POSY	TERM	COEFFICIENT	EXPONENTS							
0	1	.10000D+01 -1.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
1	2	.10000D+01 1.0000 -1.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
1	3	.10000D+01 .0000 -1.0000 1.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0000	
2	4	.10000D+01 .0000 .0000 -1.0000 7.0000	3.2000	4.7000	1.4000					
3	5	.10000D+01 .0000 .0000 -1.0000 6.4000	3.2000	4.5000	1.5000					
4	6	.10000D+01 .0000 .0000 -1.0000 6.9000	3.1000	4.9000	1.5000					
5	7	.10000D+01 .0000 .0000 -1.0000 5.5000	2.3000	4.0000	1.3000					
6	8	.10000D+01 .0000 .0000 -1.0000 6.5000	2.8000	4.6000	1.5000					
7	9	.10000D+01 .0000 .0000 -1.0000 5.7000	2.8000	4.5000	1.3000					
8	10	.10000D+01 .0000 .0000 -1.0000 6.3000	3.3000	4.7000	1.6000					
9	11	.10000D+01 .0000 .0000 -1.0000 4.9000	2.4000	3.3000	1.0000					
10	12	.10000D+01 .0000 .0000 -1.0000 6.6000	2.9000	4.6000	1.3000					
11	13	.10000D+01 .0000 .0000 -1.0000 5.2000	2.7000	3.9000						

.....  
途中省略

.....  
55 57 .10000D-02 .0000 .0000 .0000 .0000 .0000 .0000 1.0000

52

立命館経営学 (第 49 卷 第 1 号)

56	58	.10000D-02	.0000	.0000	.0000	-1.0000	.0000	.0000	.0000
57	59	.10000D-02	.0000	.0000	.0000	.0000	-1.0000	.0000	.0000
58	60	.10000D-02	.0000	.0000	.0000	.0000	.0000	-1.0000	.0000
59	61	.10000D-02	.0000	.0000	.0000	.0000	.0000	.0000	-1.0000

\*\*\*\*\*

INITIAL LP SOLUTION IS -31.648796 =LOG( .179930E-13)

\*\*\*\*TERMINATED: SPECIFIED TOLERANCES FOR FEASIBILITY  
AND/OR CONVERGENCE CRITERION CANNOT BE MET;  
BEST AVAILABLE SOLUTION FOLLOWS:

ACTIVE GLP COLUMNS (POSY<0 IMPLIES A BOUNDING CONSTRAINT):

ID	POSY	LAMBDA
1	0	.10000000E+01
2	1	.49788792E+01
3	1	.54152812E+00
24	22	.35714278E+01
26	24	.94897963E+00
35	33	.55204073E+01
56	54	.30489784E+01
57	55	.15326533E+01

\*\*\*\*\*

\* REPORT OF OPTIMAL \*

\* PRIMAL VARIABLES \*

\*\*\*\*\*

PRIMAL VARIABLES ARE

I	X(I)
1	.40792672E+13

2 .22461370E+14  
 3 .18382103E+14  
 4 .32304001  
 5 .72850406E-01  
 6 1000.0001  
 7 1000.0001

POSYNOMIAL NUMBER 0 VALUE= .24514207E-12 VALUES OF TERMS  
 ARE .24514207E-12

POSYNOMIAL NUMBER 1 VALUE= 1.0000000 VALUES OF TERMS  
 ARE .18161257 .81838742

POSYNOMIAL NUMBER 2 VALUE= .91239227E-02 VALUES OF TERMS  
 ARE .91239227E-02

POSYNOMIAL NUMBER 3 VALUE= .90079881E-02 VALUES OF TERMS  
 ARE .90079881E-02

POSYNOMIAL NUMBER 4 VALUE= .10544179 VALUES OF TERMS  
 ARE .10544179

.....

途中省略

.....

POSYNOMIAL NUMBER 55 VALUE= 1.0000001 VALUES OF TERMS  
 ARE 1.0000001

POSYNOMIAL NUMBER 56 VALUE= .30955918E-02 VALUES OF TERMS  
 ARE .30955918E-02

POSYNOMIAL NUMBER 57 VALUE= .13726760E-01 VALUES OF TERMS  
 ARE .13726760E-01

POSYNOMIAL NUMBER 58 VALUE= .99999990E-06 VALUES OF TERMS  
 ARE .99999990E-06

POSYNOMIAL NUMBER 59 VALUE= .99999990E-06 VALUES OF TERMS  
 ARE .99999990E-06

PRIMAL SOLUTION .24514207E-12

LP SOLUTION .24514207E-12

CONVERGENCE CRITERION -.31922306E-13

TOTAL NUMBER OF ITERATIONS = 12

TOTAL NUMBER OF COLUMNS GENERATED= 73

CPU TIME = .0800 SECONDS

最適解は下のようになる。

表 3 例題 1 の最適解とその対数値

	解	ln( 解 )		解	ln( 解 )		解	ln( 解 )
s	4.08E+12	29.03694	x	0.32304	-1.12998	z	1000.0001	6.907755
k1	2.25E+13	30.74282	y	7.29E-02	-2.61935	u	1000.0001	6.907755
k2	1.84E+13	30.5424						

この最適解のもと、識別関数  $\phi(\alpha, \beta, \gamma, \delta)$  の対数値は  $d = (k1+k2) / 2=30.6426$ 

$$\ln(\phi(\alpha, \beta, \gamma, \delta)) = -1.12998\alpha - 2.61935\beta + 6.907755\gamma$$

$$+ 6.907755\delta - 30.6426$$

サポートベクターは表中の判定の \* の 3 点ある。

表 4 例題 1 の識別結果・判定結果・サポートベクター

					真の種類			判定種類	
学習データ	7	3.2	4.7	1.4	versicolor	1	-4.69686	versicolor	
学習データ	6.4	3.2	4.5	1.5	versicolor	1	-4.70964	versicolor	
学習データ	6.9	3.1	4.9	1.5	versicolor	1	-2.2496	versicolor	
学習データ	5.5	2.3	4	1.3	versicolor	1	-6.17068	versicolor	
学習データ	6.5	2.8	4.6	1.5	versicolor	1	-3.08413	versicolor	
学習データ	5.7	2.8	4.5	1.3	versicolor	1	-4.25247	versicolor	
学習データ	6.3	3.3	4.7	1.6	versicolor	1	-2.78625	versicolor	
学習データ	4.9	2.4	3.3	1	versicolor	1	-12.6624	versicolor	
学習データ	6.6	2.9	4.6	1.3	versicolor	1	-4.84061	versicolor	
学習データ	5.2	2.7	3.9	1.4	versicolor	1	-6.87942	versicolor	
学習データ	5	2	3.5	1	versicolor	1	-10.3461	versicolor	
学習データ	5.9	3	4.2	1.5	versicolor	1	-5.69311	versicolor	
学習データ	6	2.2	4	1	versicolor	1	-8.54606	versicolor	
学習データ	6.1	2.9	4.7	1.4	versicolor	1	-2.89407	versicolor	
学習データ	5.6	2.9	3.6	1.3	versicolor	1	-10.6184	versicolor	
学習データ	6.7	3.1	4.4	1.4	versicolor	1	-6.16825	versicolor	
学習データ	5.6	3	4.5	1.5	versicolor	1	-3.28179	versicolor	
学習データ	5.8	2.7	4.1	1	versicolor	1	-8.93896	versicolor	
学習データ	6.2	2.2	4.5	1.5	versicolor	1	-1.8643	versicolor	
学習データ	5.6	2.5	3.9	1.1	versicolor	1	-8.87987	versicolor	
学習データ	5.9	3.2	4.8	1.8	versicolor	1	-7E-07	versicolor	*
学習データ	6.1	2.8	4	1.3	versicolor	1	-8.15834	versicolor	
学習データ	6.3	2.5	4.9	1.5	versicolor	1	-4E-07	versicolor	*



					真の種類			判定種類	
学習データ	6.1	2.8	4.7	1.2	versicolor	1	-4.01369	versicolor	
学習データ	6.4	2.9	4.3	1.3	versicolor	1	-6.68694	versicolor	
検査データ	6.6	3	4.4	1.4	versicolor	1	-5.79332	versicolor	
検査データ	6.8	2.8	4.8	1.4	versicolor	1	-2.73235	versicolor	
検査データ	6.7	3	5	1.7	versicolor	0	0.31066	virginica	誤判定
検査データ	6	2.9	4.5	1.5	versicolor	1	-3.47185	versicolor	
検査データ	5.7	2.6	3.5	1	versicolor	1	-12.7087	versicolor	
検査データ	5.5	2.4	3.8	1.1	versicolor	1	-9.19572	versicolor	
検査データ	5.5	2.4	3.7	1	versicolor	1	-10.5773	versicolor	
検査データ	5.8	2.7	3.9	1.2	versicolor	1	-8.93896	versicolor	
検査データ	6	2.7	5.1	1.6	versicolor	0	1.88745	versicolor	誤判定
検査データ	5.4	3	4.5	1.5	versicolor	1	-3.05579	versicolor	
検査データ	6	3.4	4.5	1.6	versicolor	1	-4.09075	versicolor	
検査データ	6.7	3.1	4.7	1.5	versicolor	1	-3.40515	versicolor	
検査データ	6.3	2.3	4.4	1.3	versicolor	1	-4.31156	versicolor	
検査データ	5.6	3	4.1	1.3	versicolor	1	-7.42644	versicolor	
検査データ	5.5	2.5	4	1.3	versicolor	1	-6.69455	versicolor	
検査データ	5.5	2.6	4.4	1.2	versicolor	1	-4.88416	versicolor	
検査データ	6.1	3	4.6	1.4	versicolor	1	-3.84678	versicolor	
検査データ	5.8	2.6	4	1.2	versicolor	1	-7.98625	versicolor	
検査データ	5	2.3	3.3	1	versicolor	1	-12.5134	versicolor	
検査データ	5.6	2.7	4.2	1.3	versicolor	1	-5.94986	versicolor	
検査データ	5.7	3	4.2	1.2	versicolor	1	-7.53944	versicolor	
検査データ	5.7	2.9	4.2	1.3	versicolor	1	-6.58673	versicolor	
検査データ	6.2	2.9	4.3	1.3	versicolor	1	-6.46095	versicolor	
検査データ	5.1	2.5	3	1.1	versicolor	1	-14.5319	versicolor	
検査データ	5.7	2.8	4.1	1.3	versicolor	1	-7.01557	versicolor	
学習データ	6.3	3.3	6	2.5	virginica	1	12.21039	virginica	
学習データ	5.8	2.7	5.1	1.9	virginica	1	3.985353	virginica	
学習データ	7.1	3	5.9	2.1	virginica	1	8.638332	virginica	
学習データ	6.3	2.9	5.6	1.8	virginica	1	5.659597	virginica	
学習データ	6.5	3	5.8	2.2	virginica	1	9.316319	virginica	
学習データ	7.6	3	6.6	2.1	virginica	1	12.90877	virginica	
学習データ	4.9	2.5	4.5	1.7	virginica	1	-2E-07	virginica	*
学習データ	7.3	2.9	6.3	1.8	virginica	1	9.365046	virginica	
学習データ	6.7	2.5	5.8	1.8	virginica	1	7.636895	virginica	
学習データ	7.2	3.6	6.1	2.5	virginica	1	11.09838	virginica	
学習データ	6.5	3.2	5.1	2	virginica	1	2.57547	virginica	
学習データ	6.4	2.7	5.3	1.9	virginica	1	4.688917	virginica	
学習データ	6.8	3	5.5	2.1	virginica	1	6.214223	virginica	
学習データ	5.7	2.5	5	2	virginica	1	4.622221	virginica	
学習データ	5.8	2.8	5.1	2.4	virginica	1	7.177296	virginica	
学習データ	6.4	3.2	5.3	2.3	virginica	1	6.142346	virginica	
学習データ	6.5	3	5.5	1.8	virginica	1	4.480891	virginica	

					真の種類			判定種類	
学習データ	7.7	3.8	6.7	2.2	virginica	1	12.08185	virginica	
学習データ	7.7	2.6	6.9	2.3	virginica	1	17.29739	virginica	
学習データ	6	2.2	5	1.5	virginica	1	1.615154	virginica	
学習データ	6.9	3.2	5.7	2.3	virginica	1	8.340458	virginica	
学習データ	5.6	2.8	4.9	2	virginica	1	3.258639	virginica	
学習データ	7.7	2.8	6.7	2	virginica	1	13.31964	virginica	
学習データ	6.3	2.7	4.9	1.8	virginica	1	1.348037	virginica	
学習データ	6.7	3.3	5.7	2.1	virginica	1	6.922968	virginica	
検査データ	7.2	3.2	6	1.8	virginica	1	6.619913	virginica	
検査データ	6.2	2.8	4.8	1.8	virginica	1	0.508325	virginica	
検査データ	6.1	3	4.9	1.8	virginica	1	0.788229	virginica	
検査データ	6.4	2.8	5.6	2.1	virginica	1	7.88086	virginica	
検査データ	7.2	3	5.8	1.6	virginica	1	4.380681	virginica	
検査データ	7.4	2.8	6.1	1.9	virginica	1	8.823208	virginica	
検査データ	7.9	3.8	6.4	2	virginica	1	8.401973	virginica	
検査データ	6.4	2.8	5.6	2.2	virginica	1	8.571636	virginica	
検査データ	6.3	2.8	5.1	1.5	virginica	1	0.395327	virginica	
検査データ	6.1	2.6	5.6	1.4	virginica	1	3.908294	virginica	
検査データ	7.7	3	6.1	2.3	virginica	1	10.72345	virginica	
検査データ	6.3	3.4	5.6	2.4	virginica	1	8.494576	virginica	
検査データ	6.4	3.1	5.5	1.8	virginica	1	4.331954	virginica	
検査データ	6	3	4.8	1.8	virginica	1	0.210451	virginica	
検査データ	6.9	3.1	5.4	2.1	virginica	1	5.148515	virginica	
検査データ	6.7	3.1	5.6	2.4	virginica	1	8.828389	virginica	
検査データ	6.9	3.1	5.1	2.3	virginica	1	4.45774	virginica	
検査データ	5.8	2.7	5.1	1.9	virginica	1	3.985353	virginica	
検査データ	6.8	3.2	5.9	2.3	virginica	1	9.835007	virginica	
検査データ	6.7	3.3	5.7	2.5	virginica	1	9.68607	virginica	
検査データ	6.7	3	5.2	2.3	virginica	1	5.636446	virginica	
検査データ	6.3	2.5	5	1.9	virginica	1	3.253458	virginica	
検査データ	6.5	3	5.2	2	virginica	1	3.790115	virginica	
検査データ	6.2	3.4	5.4	2.3	virginica	1	6.535248	virginica	
検査データ	5.9	3	5.1	1.8	virginica	1	2.395776	virginica	
						98			