サポートベクターマシン入門

栗田 多喜夫
Takio Kurita
産業技術総合研究所 脳神経情報研究部門
Neurosceince Research Institute,
National Institute of Advanced Indastrial Science and Technology
takio-kurita@aist.go.jp

概 要

最近、サポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン認識手法が注目されており、ちょっとしたブームになっている。カーネルトリックにより非線形の識別関数を構成できるように拡張したサポートベクターマシンは、現在知られている多くの手法の中でも最も認識性能の優れた学習モデルの一つである。サポートベクターマシンが優れた認識性能を発揮できるのは、未学習データに対して高い識別性能を得るための工夫があるためである。本稿では、サポートベクターマシンを中心に、識別器の学習において汎化性能を向上させるための工夫について紹介する。

1 はじめに

最近、サポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン認識手法が注目されている。カーネルトリックと呼ばれる方法を用いて、非線形の識別関数を構成できるように拡張したサポートベクターマシンは、現在知られている多くの手法の中でも最も認識性能の優れた学習モデルの一つであると考えられている。サポートベクターマシンが優れた認識性能を発揮できるのは、未学習データに対して高い識別性能 (汎化性能) を得るための工夫があるためである。サポートベクターマシンは、線形しきい素子を用いて、2 クラスのパターン識別器を構成する手法である。訓練サンプルから「マージン最大化」という基準で線形しきい素子のパラメータを学習する。本稿では、まず、サポートベクターマシンおよびカーネルトリックについて紹介し、その後、同様な構造で識別器を構成する統計手法として、重回帰分析とロジスティック回帰分析について紹介する。重回帰分析やロジスティック回帰分析でも、未学習サンプルに対する性能を向上させるための工夫が提案されているので、識別器の学習における汎化性能を向上させるための工夫について紹介する。また、サポートベクターマシンとそれらの手法を評価関数のレベルで比較する。

2 カーネル学習法

最近、サポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン認識手法が注目されており、ちょっとしたブームになっている。サポートベクターマシンは、1960 年代に Vapnik 等が考案した Optimal Separating Hyperplane を起源とし、1990 年代になってカーネル学習法と組み合わせた非線形の識別手法へと拡張された。カーネルトリックにより非線形の識別関数

が構成できるように拡張したサポートベクターマシンは、現在知られている手法の中でも最もパターン認識性能の優秀な学習モデルの一つである。ただし、サポートベクターマシンは、基本的には 2 つのクラスを識別する識別器を構成するための学習法であり、文字認識などの多クラスの識別器を構成するためには、複数のサポートベクターマシンを組み合わせるなどの工夫が必要となる。ここでは、まず、サポートベクターマシンを中心にカーネル学習法を用いて訓練サンプルから非線形の識別器を構成する方法について概説する。一般に、カーネル学習法を用いて学習された識別器が、訓練サンプルに含まれていない未学習データに対しても高い識別性能を発揮できるためには、汎化能力を向上させるための工夫が必要である。サポートベクターマシンでは、「マージン最大化」という基準を用いることでこれを実現している。

2.1 パターン認識における学習

パターン認識を実現するためには、まず、認識対象から何らかの特徴量を計測（抽出）する必要がある。一般には、特徴量は 1 種類だけではなく、複数の特徴量を計測し、それらを同時に用いることが多い。そのような特徴量は、通常、まとめて特徴ベクトル $\boldsymbol{x}^T = (x_1, \ldots, x_M)$ として表される。ここで、$\boldsymbol{x}^T$ は、ベクトル $\boldsymbol{x}$ の転置を表す。また、M は、特徴量の個数である。認識対象のクラスの総数を K とし、各クラスを $C_1, \ldots, C_K$ と表すことにする。パターン認識における最も基本的な課題は、未知の認識対象を計測して得られた特徴ベクトルからその対象がどのクラスに属するかを判定する識別器を開発することである。そのためには、クラスの帰属が既知の訓練用のサンプル集合から特徴ベクトルとクラスとの確率的な対応関係を知識として学習することが必要である。未知の認識対象の識別には、学習された確率的知識を利用してそれがどのクラスに属していたかを推定（決定）する方式を指定しなければならない。その際、間違って識別する確率をできるだけ小さくすることが望ましい。特徴ベクトルとクラスとの確率的な対応関係が完全にわかっている理想的な場合には、理論的に最適な識別方式（ベイズ識別方式）が存在する。しかし、実際のパターン認識問題では、特徴ベクトルとクラスとの確率的な対応関係が完全にわかっていることは稀で、そのような確率的な関係を訓練データから学習する必要がある。

2.2 サポートベクターマシン

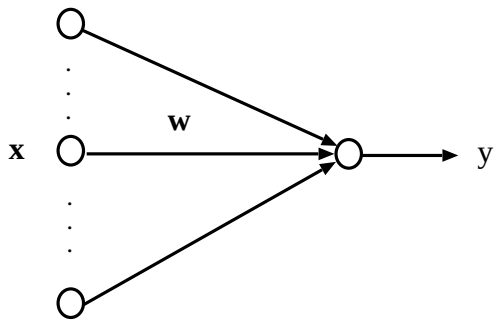


図 1: 線形しきい素子

サポートベクターマシンは、ニューロンのモデルとして最も単純な線形しきい素子を用いて、2 クラスのパターン識別器を構成する手法である。訓練サンプル集合から、「マージン最大化」と

いう基準で線形しきい素子のパラメータを学習する。線形しきい素子は、図 1 に示すようなニューロンを単純化したモデルで、入力特徴ベクトルに対し、識別関数（線形識別関数）

$$y = \mathrm{sign}(\boldsymbol{w}^T \boldsymbol{x} - h) \tag{1}$$

により 2 値の出力値を計算する。ここで、$\boldsymbol{w}$ はシナプス荷重に対応するパラメータであり、h はしきい値である。また、関数 $\mathrm{sign}(u)$ は、$u > 0$ のとき 1 をとり、$u \leq 0$ のとき -1 をとる符号関数である。このモデルは、入力ベクトルとシナプス荷重の内積がしきい値を超えれば 1 を出力し、超えなければ -1 を出力する。これは、幾何学的には、識別平面により、入力特徴空間を 2 つに分けることに相当する。今、2 つのクラスを C_1,C_2 とし、各クラスのラベルを 1 と -1 に数値化しておくとする。また、訓練サンプル集合として、N 個の特徴ベクトル $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ と、それぞれのサンプルに対する正解のクラスラベル $t_1, \ldots, t_N$ が与えられているとする。また、この訓練サンプル集合は、線形分離可能であるとする。すなわち、線形しきい素子のパラメータをうまく調整することで、訓練サンプル集合を誤りなく分けることができると仮定する。

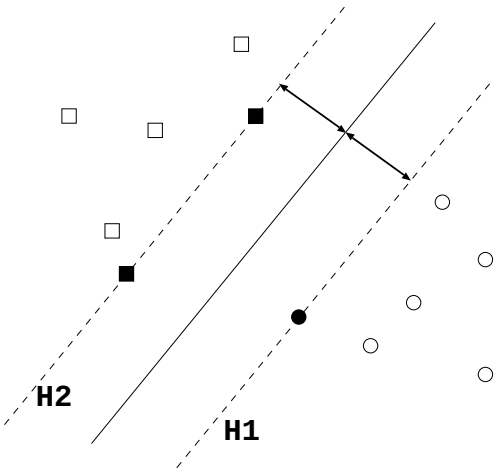


図 2: 線形しきい素子の分離超平面とマージン (◯がクラス 1 のサンプルで、□がクラス-1 のサンプルを示す。●と■はサポートベクターを示す。)

訓練サンプル集合が線形分離可能であるとしても、一般には、訓練サンプル集合を誤りなく分けるパラメータは一意には決まらない。サポートベクターマシンでは、訓練サンプルをすれすれに通るのではなく、なるべく余裕をもって分けるような識別平面が求められる。具体的には、最も近い訓練サンプルとの余裕をマージンと呼ばれる量で測り、マージンが最大となるような識別平面を求める。もし、訓練サンプル集合が線形分離可能なら、

$$t_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) \geq 1, \quad i = 1, \ldots, N \tag{2}$$

を満たすようなパラメータが存在する。これは、H1: $\boldsymbol{w}^T \boldsymbol{x} - h = 1$ と H2: $\boldsymbol{w}^T \boldsymbol{x} - h = -1$ の 2 枚の超平面で訓練サンプルが完全に分離されており、2 枚の超平面の間にはサンプルがひとつも存在しないことを示している。このとき、識別平面とこれらの超平面との距離（マージンの大きさ）は、$\frac{1}{||\boldsymbol{w}||}$ となる。したがって、マージンを最大とするパラメータ $\boldsymbol{w}$ と h を求める問題は、結局、制約条件

$$t_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) \geq 1, \quad (i = 1, \ldots, N) \tag{3}$$

の下で、目的関数

$$L(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w}||^2 \tag{4}$$

を最小とするパラメータを求める問題と等価になる。この最適化問題は、数理計画法の分野で 2 次計画問題として知られており、さまざまな数値計算法が提案されている。ここでは、双対問題に帰着して解く方法を紹介する。まず、Lagrange 乗数 $\alpha_i(\geq 0), i = 1, \ldots, N$ を導入し、目的関数を

$$L(\boldsymbol{w}, h, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 - \sum_{i=1}^{N} \alpha_i \{t_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) - 1\} \tag{5}$$

と書き換える。パラメータ $\boldsymbol{w}$ および h に関する偏微分から停留点では、

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \tag{6}$$

$$0 = \sum_{i=1}^{N} \alpha_i t_i \tag{7}$$

という関係が成り立つ。これらを上の目的関数の式に代入すると、制約条件、

$$\sum_{i=1}^{N} \alpha_i t_i = 0 \tag{8}$$

$$0 \leq \alpha_i, \quad i = 1, \ldots, N \tag{9}$$

の下で、目的関数

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{10}$$

を最大とする双対問題が得られる。これは、Lagrange 乗数 α_i $(\geq 0), i = 1, \ldots, N$ に関する最適化問題となる。その解で α_i^* が 0 でない、すなわち、$\alpha_i^* > 0$ となる訓練サンプル $\boldsymbol{x}_i$ は、先の 2 つの超平面 $\boldsymbol{w}^T \boldsymbol{x} - h = 1$ か $\boldsymbol{w}^T \boldsymbol{x} - h = -1$ のどちらかにのっている。このことから、α_i^* が 0 でない訓練サンプル $\boldsymbol{x}_i$ のことを「サポートベクター」と呼んでいる。これが、サポートベクターマシンの名前の由来である。直感的に理解できるように、一般には、サポートベクターは、もとの訓練サンプル数に比べてかなり少ない。つまり、沢山の訓練サンプルの中から小数のサポートベクターを選び出し、それらのみを用いて線形しきい素子のパラメータが決定されることになる。

実際、双対問題の最適解 $\alpha_i^*(i \geq 0)$、および停留点での条件式から、最適なパラメータ $\boldsymbol{w}^*$ は、

$$\boldsymbol{w}^* = \sum_{i \in S} \alpha_i^* t_i \boldsymbol{x}_i \tag{11}$$

となる。ここで、S はサポートベクターに対応する添え字の集合である。また、最適なしきい値 h^* は、2 つの超平面 $\boldsymbol{w}^T \boldsymbol{x} - h = 1$ か $\boldsymbol{w}^T \boldsymbol{x} - h = -1$ のどちらかにのっているという関係を利用して求めることができる。すなわち、任意のサポートベクター $\boldsymbol{x}_s, s \in S$ から

$$h^* = \boldsymbol{w}^{*T} \boldsymbol{x}_s - t_s \tag{12}$$

により求まる。

また、最適な識別関数を双対問題の最適解 $\alpha_i^*(i \geq 0)$ を用いて表現すると

$$\begin{aligned} y &= \mathrm{sign}(\boldsymbol{w}^{*T} \boldsymbol{x} - h^*) \\ &= \mathrm{sign}(\sum_{i \in S} \alpha_i^* t_i \boldsymbol{x}_i^T \boldsymbol{x} - h^*) \end{aligned} \tag{13}$$

となる。すなわち、$\alpha_i^* = 0$ となる多くの訓練サンプルを無視し、$\alpha_i^* > 0$ となる識別平面に近い少数の訓練サンプルのみを用いて識別関数が構成される。ここで、重要な点は、「マージン最大化」という基準から自動的に識別平面付近の少数の訓練サンプルのみが選択されたことであり、その結果として、未学習データに対してもある程度良い識別性能が維持できていると解釈できる。 すなわち、サポートベクターマシンは、マージン最大化という基準を用いて、訓練サンプルを撰択することで、モデルの自由度を抑制するようなモデル撰択が行われていると解釈できる。

2.3 ソフトマージン

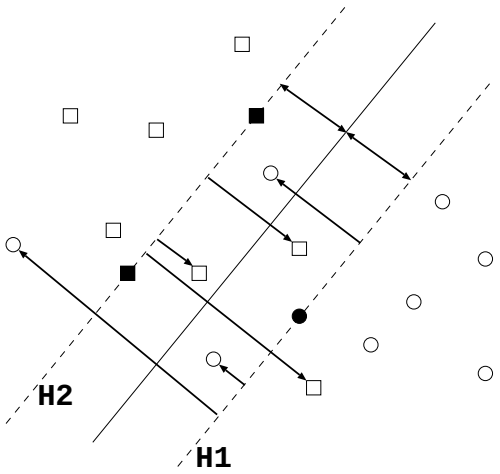


図 3: ソフトマージン (◯がクラス 1 のサンプルで、□がクラス-1 のサンプルを示す。●と■はサポートベクターを示す。)

上述のサポートベクターマシンは、訓練サンプルが線形分離可能な場合についての議論であるが、パターン認識の実問題で線形分離可能な場合は稀である。したがって、実際的な課題にサポートベクターマシンを使うには、さらなる工夫が必要である。まず考えられるのは、多少の識別誤りは許すように制約を緩める方法である。これは、「ソフトマージン」と呼ばれている。

ソフトマージン法では、マージン $\frac{1}{||\boldsymbol{w}||}$ を最大としながら、図 3 に示すように、幾つかのサンプルが超平面 H1 あるいは H2 を越えて反対側に入ってしまうことを許す。反対側にどれくらい入り込んだかの距離を、パラメータ $\xi_i(\geq 0)$ を用いて、$\frac{\xi_i}{||\boldsymbol{w}||}$ と表すとすると、その和

$$\sum_{i=1}^{N} \frac{\xi_i}{||\boldsymbol{w}||} \tag{14}$$

はなるべく小さいことが望ましい。これらの条件から最適な識別面を求める問題は、制約条件

$$\xi_i \geq 0, \quad t_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) \geq 1 - \xi_i, \quad (i = 1, \ldots, N) \tag{15}$$

の下で、目的関数

$$L(\boldsymbol{w}, \boldsymbol{\xi}) = \frac{1}{2}||\boldsymbol{w}||^2 + \gamma \sum_{i=1}^{N} \xi_i \tag{16}$$

を最小とするパラメータを求める問題に帰着される。ここで、あらたに導入したパラメータ γ は、第 1 項のマージンの大きさと第 2 項のはみ出しの程度とのバランスを決める定数である。

この最適化問題の解法は、基本的には線形分離可能な場合と同様にふたつの制約条件に対して、Lagrange 乗数 α_i、および、ν_i を導入し、目的関数を

$$\begin{aligned} L(\boldsymbol{w}, h, \boldsymbol{\alpha}, \boldsymbol{\nu}) &= \frac{1}{2}||\boldsymbol{w}||^2 + \gamma \sum_{i=1}^{N} \xi_i \\ &\quad - \sum_{i=1}^{N} \alpha_i \{t_i(\boldsymbol{w}^T \boldsymbol{x}_i - h) - (1 - \xi_i)\} \\ &\quad - \sum_{i=1}^{N} \nu_i \xi_i \end{aligned} \tag{17}$$

と書き換える。パラメータ $\boldsymbol{w}$、h、ξ_i に関する偏微分を 0 とする停留点では、

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \tag{18}$$

$$0 = \sum_{i=1}^{N} \alpha_i t_i \tag{19}$$

$$\alpha_i = \gamma - \nu_i \tag{20}$$

という関係が成り立つ。これらを目的関数の式に代入すると、制約条件

$$\sum_{i=1}^{N} \alpha_i t_i = 0 \tag{21}$$

$$0 \leq \alpha_i \leq \gamma, \quad i = 1, \ldots, N \tag{22}$$

の下で、目的関数

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{23}$$

を最大とする双対問題が得られる。線形分離可能な場合には、最適解 α_i^* の値により、平面 H1 および H2 上の訓練サンプル (サポートベクター) とそれ以外のサンプルに分類されたが、ソフトマージンの場合には、さらに、H1 および H2 をはさんで反対側にはみ出すサンプルが存在する。それらは、同様に、最適解 α_i^* の値により区別することができる。具体的には、$\alpha_i^* = 0$ なら、平面 H1 あるいは H2 の外側に存在し、学習された識別器によって正しく識別される。また、$0 < \alpha_i^* < \gamma$ の場合には、対応するサンプルは、ちょうど平面 H1 あるいは H2 の上に存在するサポートベクターとなり、これも正しく識別される。$\alpha_i^* = \gamma$ の場合には、対応するサンプルはサポートベクターとなるが、$\xi_i \neq 0$ となり、平面 H1 あるいは H2 の内側に存在することになる。

2.4 カーネルトリック

ソフトマージン法を用いることで、線形分離可能でない場合に対しても線形しきい素子のパラメータを求めることができるようになる。しかし、ソフトマージン法を用いたとしても、本質的に非線形で複雑な識別課題に対しては、必ずしも良い性能の識別器を構成できるとは限らない。本質的に非線形な問題に対応するための方法として、特徴ベクトルを非線形変換して、その空間で線形

の識別を行う「カーネルトリック」と呼ばれている方法が知られている。この方法を用いることでサポートベクターマシンの性能が飛躍的に向上した。それがサポートベクターマシンを有名にした大きな要因であると考えられる。

一般に、線形分離可能性はサンプル数が大きくなればなるほど難しくなり、逆に、特徴空間ベクトルの次元が大きくなるほど易しくなる。例えば、特徴ベクトルの次元が訓練サンプルの数よりも大きいなら、どんなラベル付けに対しても線形分離可能である。しかし、高次元への写像を行うと、次元の増加に伴い汎化能力が落ちてしまう。また、難しい問題を線形分離可能にするためには、訓練サンプルと同程度の大きな次元に写像しなければならないので、結果的に膨大な計算量が必要となってしまう。

今、元の特徴ベクトル $\boldsymbol{x}$ を非線形の写像 $\phi(\boldsymbol{x})$ によって変換し、その空間で線形識別を行うことを考えてみよう。例えば、写像 ϕ として、入力特徴を 2 次の多項式に変換する写像を用いるとすると、写像した先で線形識別を行うことは、もとの空間で 2 次の識別関数を構成することに対応する。一般には、こうした非線形の写像によって変換した特徴空間の次元は非常に大きくなりがちである。しかし、サポートベクターマシンの場合には、幸いにも、目的関数 L_D や識別関数が入力パターンの内積のみに依存した形になっており、内積が計算できれば最適な識別関数を構成することが可能である。つまり、もし非線形に写像した空間での二つの要素 $\phi(\boldsymbol{x}_1)$ と $\phi(\boldsymbol{x}_2)$ の内積が

$$\phi(\boldsymbol{x}_1)^T\phi(\boldsymbol{x}_2) = K(\boldsymbol{x}_1, \boldsymbol{x}_2) \tag{24}$$

のように、入力特徴 $\boldsymbol{x}_1$ と $\boldsymbol{x}_2$ のみから計算できるなら、非線形写像によって変換された特徴空間での特徴 $\phi(\boldsymbol{x}_1)$ や $\phi(\boldsymbol{x}_2)$ を陽に計算する代わりに、$K(\boldsymbol{x}_1, \boldsymbol{x}_2)$ から最適な非線形写像を構成できる。ここで、このような K のことをカーネルと呼んでいる。このように高次元に写像しながら、実際には写像された空間での特徴の計算を避けて、カーネルの計算のみで最適な識別関数を構成するテクニックのことを「カーネルトリック」と呼んでいる。

実用的には、K は計算が容易なものが望ましい。例えば、多項式カーネル

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (1 + \boldsymbol{x}_1^T\boldsymbol{x}_2)^p \tag{25}$$

Gauss カーネル

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(\frac{-||\boldsymbol{x}_1 - \boldsymbol{x}_2||^2}{2\sigma^2}\right) \tag{26}$$

シグモイドカーネル

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tanh\left(a\boldsymbol{x}_1^T\boldsymbol{x}_2 - b\right) \tag{27}$$

などが使われている。

式 (10) や式 (23) の目的関数 L_D は、

$$\begin{aligned} L_D(\boldsymbol{\alpha}) &= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j t_i t_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) \\ &= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j t_i t_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \end{aligned} \tag{28}$$

のように内積をカーネルで置き換えた形に書ける。また、式 (13) から最適な識別関数は、

$$\begin{aligned} y &= \mathrm{sign}(\boldsymbol{w}^{*T}\phi(\boldsymbol{x}) - h^*) \\ &= \mathrm{sign}(\sum_{i\in S}\alpha_i^* t_i \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}) - h^*) \end{aligned}$$

$$= \quad \mathrm{sign}(\sum_{i \in S} \alpha_i^* t_i K(\boldsymbol{x}_i, \boldsymbol{x}) - h^*) \tag{29}$$

のようにサポートベクターマシンの内積をカーネルで置き換えた形に書ける。ここで、この式にシグモイドカーネルを代入すると、いわゆる 3 層の多層パーセプトロンと同じ構造となる。また、Gauss カーネルを代入すると、Radial Basis Function (RBF) ネットワークと同じ構造になり、構造的には従来のニューラルネットワークと同じになる。しかし、カーネルトリックを用いて非線形に拡張したサポートベクターマシンでは、中間層から出力層への結合荷重のみが学習により決定され、前段の入力層から中間層への結合荷重は固定で、訓練データから機械的に求められる。また、中間層のユニット数が非常に大きく、訓練サンプル数と同じになる。つまり、カーネルトリックを用いて非線形に拡張したサポートベクターマシンでは、入力層から出力層への結合荷重を適応的に学習により求めない代わりにあらかじめ中間層に非常に多くのユニットを用意することで複雑な非線形写像を構成しようとする。

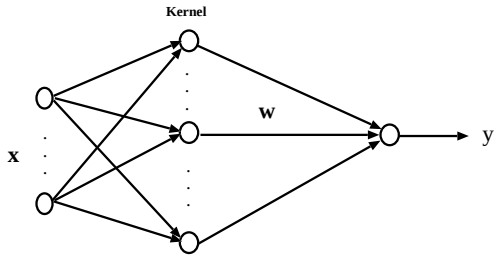


図 4: サポートベクターマシンの構造

2.5 カーネル学習法と汎化能力

カーネルトリックを用いて非線形に拡張したサポートベクターマシンでは、「マージン最大化」という基準から自動的に識別平面付近の少数の訓練サンプルに対応するカーネル（カーネル特徴）のみが選択され、最適な識別関数が構成される。これは、汎化能力の高い識別器を構成するために、カーネル特徴を選択することでモデルの自由度を低く抑えて、より安定なモデルを構成したとみなすことができる。そう考えると、カーネル特徴を選択するだけでなく、入力特徴を選択することも汎化能力の向上につながると期待できる。さらには、中間層のニューロン数を削減するために、いくつかのサンプルを統合した代表ベクトルを用いてカーネル特徴を構成することなども考えられる。

また、サポートベクターマシンでは、「マージン最大化」という基準でカーネル特徴が選択されたが、その基準は汎化能力を評価する手法の一つとして知られている CV 次元と関連している。パターン識別器の学習における汎化性能は、学習に用いない未知のデータに対する識別性能であるので、文字認識などのように、訓練サンプル以外に汎化性能を評価するためのデータを比較的容易に集めることができる場合には、訓練サンプル以外のサンプルに対する識別率を計算し、その結果から、直接的に汎化性能を評価することも可能である。つまり、汎化性能を評価するためのサンプルを用意し、そのサンプルに対する識別率に基づいてカーネル特徴や入力特徴を選択することが可能である。

カーネル学習では、入力特徴の選択の他に、Gauss カーネルの場合のカーネル幅 σ のようなカーネルのパラメータをうまく設定しなければ高い汎化性能は得られない。現状ではそれらのパラメー

タは試行錯誤的に決められていることが多いが、汎化性能を評価するためのサンプルに対する識別率を評価することで、適切なパラメータを決定することも可能である。

サポートベクターマシンでは、２クラスの識別のために線形しきい値素子を用いたが、それ以外にも、目的に応じて、主成分分析、判別分析、線形回帰などの多変量解析手法とカーネルトリックを組み合わせることも可能である。そうすることでカーネルベースの非線形の多変量解析が実現できる。特に、パターン識別器を構成するには、カーネル判別分析が有効であろう。また、後段に多クラスの識別のための最も簡単なニューラルネットモデルの１つである multinomial logit model を用いると、各クラスの事後確率を直接推定する非線形予測モデルを構成することも可能である。

3　識別のための線形手法と汎化性

前章では、最近注目されているサポートベクターマシンとカーネル学習法について概説した。ここでは、それが従来の統計的パターン認識手法とどのような関係にあるかについて考察してみたい。

3.1　線形しきい素子を用いた識別器

サポートベクターマシンは、線形しきい素子を用いた識別器であるが、Rosenblatt が提案したパーセプトロンも、同様に、線形しきい素子を用いて、訓練サンプルから学習する識別機械である。以下では多層パーセプトロンと区別するためにこれを単純パーセプトロンと呼ぶことにする。サポートベクターマシンと同様に、単純パーセプトロンでは、入力 $\boldsymbol{x} = (x_1, \ldots, x_M)^T$ に対する出力 y を

$$\begin{aligned} y &= f(\eta) \\ \eta &= \boldsymbol{w}^T \boldsymbol{x} - h = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} \end{aligned} \tag{30}$$

のように計算する。ここで、w_i は、i 番目の入力から出力への結合荷重であり、h はバイアスである。以下では、簡単のために、これらをまとめて、$\tilde{\boldsymbol{w}} = (h, w_1, \ldots, w_M)^T$ のように表すものとする。また、入力特徴ベクトルに定数項を加えたベクトルを $\tilde{\boldsymbol{x}} = (-1, x_1, \ldots, x_M)^T$ と表す。出力ユニットの入出力関数 f は、Rosenblatt のオリジナルなモデルではしきい関数

$$f(\eta) = \mathrm{sign}(\eta) = \begin{cases} 1 & \text{if } \eta \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{31}$$

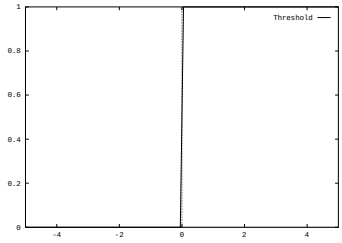
が用いられた。この他の入出力関数としては線形関数

$$f(\eta) = \eta \tag{32}$$

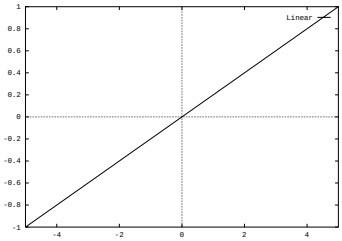
やロジスティック関数

$$f(\eta) = \frac{\exp(\eta)}{1 + \exp(\eta)} \tag{33}$$

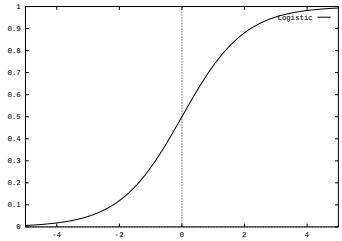
がよく使われる。多変量データ解析の用語を用いれば、出力ユニットの入出力関数が線形関数の単純パーセプトロンは、線形重回帰モデルであり、ロジスティック関数の単純パーセプトロンは、ロジスティック回帰モデルである。



(a) しきい値関数



(b) 線形関数



(c) ロジスティック関数

図 5: 出力ユニットの入出力関数

3.2 単純パーセプトロンの学習

単純パーセプトロンの結合荷重 (パラメータ) を推定するための学習アルゴリズムとしていくつかの方法が提案されているが、Rosenblatt らの方法は、ネットワークにあるパターンを分類させてみて間違っていたら結合荷重を修正する誤り訂正型の方法であった。しかし、この学習規則は、線形分離可能でない場合、すなわち、誤識別 0 にする線形識別関数が存在しない場合には、誤り訂正の手続きを無限に繰り返しても解に到達できない可能性がある。また、学習を途中で打ち切った場合に得られるパラメータが最適であるという保証もない。

3.3 線形重回帰分析

これに対して、出力ユニットの入出力関数として線形関数を用い、ネットワークの出力と教師信号と平均 2 乗誤差を最小にするような結合荷重を推定する場合には、平均 2 乗誤差の意味で最適なパラメータを求めることができる。

今、N 個の学習用のデータを $\{(\boldsymbol{x}_i, t_i)|i = 1, \ldots, N\}$ とする。ここで、$\boldsymbol{x}_i$ が入力ベクトルで、その入力ベクトルに対する望みの出力 (教師信号) が t_i である。この時、この学習用のデータに対す

る 2 乗誤差は、

$$\varepsilon_{emp}^2 = \sum_{i=1}^{N}(t_i - y_i)^2 = \sum_{i=1}^{N}\varepsilon_{emp}^2(i) \tag{34}$$

となる。最適なパラメータを求めるために、パラメータ (結合荷重) $\tilde{\boldsymbol{w}}$ を逐次更新することにより次第に最適なパラメータに近似させる最急降下法を用いることにすると、2 乗誤差 ε_{emp}^2 のパラメータに関する偏微分を計算する必要がある。2 乗誤差 ε_{emp}^2 のパラメータ w_j に関する偏微分は、

$$\frac{\partial\varepsilon_{emp}^2}{\partial w_j} = \sum_{i=1}^{N} -2(t_i - y_i)x_{ij} = \sum_{i=1}^{N} -2\delta_i x_{ij} \tag{35}$$

となる。また、バイアス h に関する偏微分は、

$$\frac{\partial\varepsilon_{emp}^2}{\partial h} = \sum_{i=1}^{N} -2(t_i - y_i)(-1) = \sum_{i=1}^{N} -2\delta_i(-1) \tag{36}$$

ただし、$\delta_i = (t_i - y_i)$ である。従って、最急降下法によるパラメータの更新式は、

$$w_j \quad \Leftarrow \quad w_j + \alpha(\sum_{i=1}^{N}\delta_i x_{ij}) \tag{37}$$

$$h \quad \Leftarrow \quad h + \alpha(\sum_{i=1}^{N}\delta_i(-1)) \tag{38}$$

のようになる。ここで、α は、学習係数 (learning rate) である。この更新法は、Widrow-Hoff の学習規則 (Widrow-Hoff learning rule) と呼ばれている。また、教師信号 t_i とネットワークの出力 y_i の誤差 δ_i に応じてパラメータを修正するため、デルタルール (delta rule) と呼ばれることもある。

Widrow-Hoff の学習規則では、最急降下法を用いて逐次近似によりパラメータを推定するが、重回帰分析の場合には、逐次学習ではなく、最適な解を行列計算により陽に求めることが可能である。

今、訓練サンプルデータの入力ベクトルを並べた $N \times (M+1)$ 次元の行列を $X = (\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N)^T$ とし、教師信号を並べた N 次元のベクトルを $\boldsymbol{t} = (t_1, \ldots, t_N)^T$ とする。これらを用いると 2 乗誤差は、

$$\varepsilon_{emp}^2 = \sum_{i=1}^{N}(t_i - y_i)^2 = ||\boldsymbol{t} - X\tilde{\boldsymbol{w}}||^2 \tag{39}$$

のように書ける。これをパラメータ $\tilde{\boldsymbol{w}}$ で偏微分して 0 とおくと、

$$\frac{\partial\varepsilon_{emp}^2}{\partial\tilde{\boldsymbol{w}}} = X^T(\boldsymbol{t} - X\tilde{\boldsymbol{w}}) = 0 \tag{40}$$

となる。従って、(X^TX) が正則ならば、最適なパラメータ $\tilde{\boldsymbol{w}}^*$ は、

$$\tilde{\boldsymbol{w}}^* = (X^TX)^{-1}X^T\boldsymbol{t} \tag{41}$$

となる。これは、重回帰分析 (multiple regression analysis) と呼ばれる最も基本的な多変量データ解析と等価である。重回帰分析では、$\boldsymbol{x}$ は説明変数 (explanatory variable)、t は目的変数 (criterion variable) と呼ばれている。

3.4 重回帰分析のための汎化性向上の工夫

重回帰分析 (multiple regression analysis) は、訓練サンプル集合から予測モデルを構築するための最も基本的な多変量データ解析手法のひとつとして様々な分野で応用されている。一般に、予測モデルを構築する目的は、訓練サンプル集合に含まれていない未学習の説明変数から、目的変数の値を推定したいためである。したがって、構築された予測モデルが訓練サンプル集合に含まれていない未学習サンプルに対して十分な予測性能を発揮しなければ、予測モデルを構築する意味が無い。未学習データに対する予測性能は、汎化性と呼ばれており、予測モデルを構築する際の重要な要素であり、従来から中心的な感心が払われて来た。その代表的なものが、変数撰択法や収縮法 (shrinkage method) あるいは正則化法 (regularization method) 等がある。以下にその代表的な手法として、変数撰択法とリッジ回帰 (ridge regression) について紹介する。

(1) 変数選択法

入力特徴ベクトル $\boldsymbol{x}$ の中には予測モデルに取って有用な特徴のみでなく不用な特徴が含まれていることがある。例えば、極端な場合として、予測に全無関係な特徴が含まれているとすると、その特徴は未学習サンプルの予測には有効に働かないで、逆に予測の邪魔をすることに成りかねない。また、訓練サンプルの数に比べて入力特徴の数が多い場合には、予測モデルのパラメータを一意に決めることすらできなくなってしまう。このような場合には、特徴の中から予測に有効な特徴の部分集合を選び出して予測モデルを構築することが必要となる。このような与えられた特徴の中から予測に有効な特徴の部分集合を選び出して予測モデルを構築する手法は、変数選択法と呼ばれている。

変数選択のためには、すべての特徴の部分集合に対して、予測性能を評価する必要がある。しかし、部分集合の数は、特徴の数が増えると指数関数的に増大する。したがって、特徴の数が多い場合には、すべての部分集合に対して評価することるは現実的では無い。そのため、比較的良い特徴の部分集合を探策する手法が提案さるれている。単純な方法としては、Forward stepwise selection あるいは、Backward stepwise selection と呼ばれる手法がある。Forward stepwise selection は、最初、特徴 1 個のみのモデルからはじめて、特徴を 1 個づつ追加して行くことで、最も良い特徴の組を選び出す。逆に、Backward stepwise selection は、全ての特徴を含むモデルから特徴を 1 個づつ取り除いて行くことで、最も良い特徴の組を選び出す。これらの他にも遺伝的アルゴリズムを用いて特徴の組を撰択することなども可能である。

変数選択を行うためには、特徴の部分集合に対して学習が終った予測モデルの予測性能を評価できなければならない。先の訓練サンプルに対する 2 乗誤差基準は、特徴の数を増やせば増やすほど小さくなるので、この基準で特徴の部分集合を選択することはできない。

予測モデルの汎化性能は、学習に用いない未知のデータに対する予測性能であるので、訓練サンプル以外に汎化性能を評価するためのデータを比較的容易に集めることができる場合には、訓練サンプル以外のサンプルに対する予測性能を評価することも可能である。つまり、汎化性能を評価するためのサンプルを用意し、そのサンプルに対する予測性能が最大となるような特徴の部分集合を選択することが可能である。この方法は、最も簡単で、最も直接的な方法であり、訓練サンプル以外にデータを集めることが可能な場合には、まず試みてみるべき方法である。

訓練サンプルを集めることが難しく、訓練サンプルが少ない場合には、訓練サンプル以外の評価用データを用意することが難しい。このような場合には、訓練サンプルのみから予測性能を評価しなければならない。かなり多くの計算量が必要であるが、計算パワーさえあれば、比較的簡単に予測性能を評価できる方法に、resampling 手法がある。leave-one-out 法は、その中でも最も単純な手法である。leave-one-out 法では、N 個のサンプルが与えられた場合、それを $N-1$ 個の訓練サ

ンプルと 1 個の評価用サンプルとに分割し、$N-1$ 個の訓練サンプルを用いた学習結果で 1 個の評価用サンプルを評価する。このような分割の仕方は N 通りあるので、その全てに対する評価結果の平均を計算し、それを予測性能の評価値として利用する。その他、もう少し洗練された手法として、jackknife 法 [9, 10] や bootstrap 法 [11, 12, 13] 等の resampling 手法もある。resampling 手法は、コンピュータの計算パワーを最大限に利用することで、予測性能を評価する手法であり、現在のようにコンピュータの性能が急激に向上し、コンピュータの計算パワーが至る所で有り余っているような状況では、もっともっと利用しても良い手法であると考える。

訓練サンプルに対する 2 乗誤差基準の代わりに、予測性能を評価するための訓練サンプルのみから計算できる評価基準も提案されている。重回帰分析では、F 統計量を用いる方法もあるが、その他にも、赤池の AIC(An Information Theoretical Criterion)[14, 15] や Rissanen の MDL(Minimum Description Length)[16, 17] などの情報量基準も有名である。このような方法は、学習は一回のみでよく、比較的簡便な評価が可能となる。重回帰分析を用いたパーセプトロンの結合係数の学習は最尤推定とみなすことができるので、学習されたパラメータを使って計算した対数尤度 (最大対数尤度) から AIC や MDL などの情報量基準を計算することにより、予測モデルの予測性能を比較することが可能となる。

AIC は赤池により最大対数尤度と期待平均対数尤度の間の偏りの解析的評価から導出されたもので、最尤推定するモデルの自由度を J とすると、

$$AIC = -2(\text{最大対数尤度}) + 2J \tag{42}$$

のように定義される。一方、MDL は Rissanen により符号化における記述長最小化 (Minimal Discription Length) 原理として導出されたもので、

$$MDL = -(\text{最大対数尤度}) + \frac{N}{2}\log J \tag{43}$$

のように定義される。これらの評価を用いると、訓練サンプルに対する当てはまりに大きな差があると第 1 項に大きな差があらわれ当てはまりの良いモデルが選ばれ、第 1 項に大きな差が無い場合には第 2 項が作用して自由度の小さいモデルが選択される。従って、予測性能の高いモデルを撰択するためには、様々な変数の部分集合を用いたモデルでのパラメータを学習し、そのパラメータかるら対数尤度を計算し、AIC あるいは MDL の小さい変数の組を選択すればよい。

(2) リッジ回帰

変数選択法では、説明変数の組を選択することで、未学習のデータ対する予測性能の良いモデルを構築しようとするが、この変数選択のプロセスは、変数を選ぶ、選ばないというように離散的である。それに対して、もう少し連続的にモデルを制限する方法として、Shrinkage 法と呼ばれる方法がある。その代表的なものが、リッジ (rigde regression) 回帰である。

リッジ回帰では、 2 乗誤差基準

$$\varepsilon^2_{emp} = \sum_{i=1}^{N}(t_i - y_i)^2 = \sum_{i=1}^{N}(t_i - (\sum_{j=1}^{M} w_j x_{ij} - h))^2 \tag{44}$$

に、パラメータ w_j 対して、その大きさが大きく成りすぎないように、

$$\sum_{j=1}^{M} w_j^2 \tag{45}$$

のようなペナルティを課す。具体的には、これらをまとめて

$$Q(\boldsymbol{w}, h) = \sum_{i=1}^{N}(t_i - (\sum_{j=1}^{M} w_j x_{ij} - h))^2 + \lambda \sum_{j=1}^{M} w_j^2 \tag{46}$$

のような評価基準を考え、これが最小となるようなパラメータを求める。ここで、λ は、2 乗誤差とペナルティとのバランスを決定する定数である。したがって、$\lambda = 0$ の場合には、リッジ回帰は通常の回帰分析と同じになる。

今、$Q(\boldsymbol{w}, h)$ を h で偏微分し 0 とおくと、

$$\frac{\partial Q(\boldsymbol{w}, h)}{\partial h} = 2N(\bar{t} - \sum_{j=1}^{M} w_j \bar{x}_j + h) = 0 \tag{47}$$

となる。これから、h に関する条件

$$h = -\bar{t} + \sum_{j=1}^{M} w_j \bar{x}_j \tag{48}$$

が得られる。ここで、$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$、および $\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$ である。これを $Q(\boldsymbol{w}, h)$ の式に代入すると、

$$\begin{aligned} Q(\boldsymbol{w}) &= \sum_{i=1}^{N}\{(t_i - \bar{t}) - \sum_{j=1}^{M} w_j(x_{ij} - \bar{x}_j)\}^2 \\ &+ \lambda \sum_{j=1}^{M} w_j^2 \\ &= (\tilde{\boldsymbol{t}} - \tilde{X}\boldsymbol{w})^T(\tilde{\boldsymbol{t}} - \tilde{X}\boldsymbol{w}) + \lambda \boldsymbol{w}^T\boldsymbol{w} \end{aligned} \tag{49}$$

となる。ここで、$\tilde{\boldsymbol{t}}$、および、$\tilde{X}$ は、それぞれ、$(t_i - \bar{t})$ を要素とするベクトル、および、$(x_{ij} - \bar{x}_j)$ を要素とする行列である。これを $\boldsymbol{w}$ で偏微分すると

$$\frac{\partial Q(\boldsymbol{w})}{\partial \boldsymbol{w}} = -2\tilde{X}\tilde{\boldsymbol{t}} + 2(\tilde{X}^T\tilde{X}\boldsymbol{w} + \lambda I)\boldsymbol{w} = 0 \tag{50}$$

となり、最適なパラメータ $\boldsymbol{w}^*$ は、

$$\boldsymbol{w}^* = (\tilde{X}^T\tilde{X} + \lambda I)^{-1}\tilde{X}^T\tilde{\boldsymbol{t}} \tag{51}$$

となる。これは、行列 $\tilde{X}^T\tilde{X}$ の対角要素に λ を加えてから逆行列を計算することに対応する。これにより、行列 $\tilde{X}^T\tilde{X}$ が正則でない場合でも、逆行列が不定になることが防げる。また、逆行列の数値計算を安定化させる効果もある。

3.5 ロジスティック回帰

入出力関数としてロジスティック関数を用い、最尤法によりパラメータを推定する場合には、ロジスティック回帰と呼ばれる手法と等価となる。この場合には、ロジスティック回帰のためのパラメータ推定アルゴリズムとして知られているフィッシャーのスコアリングアルゴリズムを学習に利用することも可能である。

今、訓練サンプル集合を $\{(\boldsymbol{x}_i, u_i)|i = 1, \ldots, N\}$ とする。ここでは、教師信号 u_i は、0 か 1 の 2 値で与えられるものとする。

入力 $\boldsymbol{x}$ を与えたときの出力 y を、入力 $\boldsymbol{x}$ のもとで教師信号 u が 1 である確率の推定値と考えると、訓練サンプル集合に対するネットワークの尤度は、

$$L = \prod_{i=1}^{N} y_i^{u_i}(1 - y_i)^{(1-u_i)} \tag{52}$$

で与えられる。従って、その対数 (対数尤度) は、

$$\begin{aligned} l &= \sum_{i=1}^{N}\{u_i \log y_i + (1-u_i)\log(1-y_i)\} \\ &= \sum_{i=1}^{N}\{u_i \log\{\frac{\exp(\eta_i)}{1+\exp(\eta_i)}\} \\ &\quad +(1-u_i)\log\{\frac{1}{1+\exp(\eta_i)}\}\} \\ &= \sum_{i=1}^{N}\{u_i\eta_i - \log\{1+\exp(\eta_i)\}\} \end{aligned} \tag{53}$$

となる。これを最大とするパラメータがネットワークの最尤推定値である。

線形重回帰分析の場合と同様に、まずは、最急降下法によりパラメータ $\tilde{\boldsymbol{w}}$ を逐次更新することで最適なパラメータを求める方法について考えてみよう。対数尤度のパラメータ w_j に関する偏微分は、

$$\frac{\partial l}{\partial w_j} = \sum_{i=1}^{N}(u_i - y_i)x_{ij} = \sum_{i=1}^{N}\delta_i x_{ij} \tag{54}$$

のようになる。ここで $\delta_i = (u_i - y_i)$ である。一方、対数尤度のパラメータ h に関する偏微分は、

$$\frac{\partial l}{\partial h} = \sum_{i=1}^{N}(u_i - y_i)(-1) = \sum_{i=1}^{N}\delta_i(-1) \tag{55}$$

となる。したがって、パラメータの更新式は、

$$w_j \Leftarrow w_j + \alpha(\sum_{i=1}^{N}\delta_i x_{ij}) \tag{56}$$

$$h \Leftarrow h + \alpha(\sum_{i=1}^{N}\delta_i(-1)) \tag{57}$$

のようになる。面白いことに、この更新式は、重回帰分析のパラメータを最急降下法で求める Widrow-Hoff の学習規則 (Widrow-Hoff learning rule) と全く同じ形をしている。ただし、出力値 y_i の計算方法が異なるので、結果的には、最適なパラメータは異なる値に収束する。

最尤推定においては、Fisher 情報行列が重要な役割を演じる。一般に、データ y がパラメータ $\theta_1, \ldots, \theta_M$ をもつ密度関数 $f(y, \theta_1, \ldots, \theta_M)$ をもつ分布に従うとき、

$$F_{ij} = -E\left(\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log f(y, \theta_1, \ldots, \theta_M)\right) \tag{58}$$

を Fisher 情報量と呼び、行列 $F = [F_{ij}]$ を Fisher 情報行列という。Fisher 情報量は不変推定量の分散と密接に関係している。

ここでは、ロジスティック回帰の Fisher 情報量を具体的に計算する。そのためには、式 (53) の対数尤度の 2 次微分を計算する必要がある。対数尤度の 2 次微分は、

$$\frac{\partial^2 l}{\partial\tilde{w}_k\partial\tilde{w}_j} = -\sum_{i=1}^{N}\omega_i\tilde{x}_{ik}\tilde{x}_{ij} \tag{59}$$

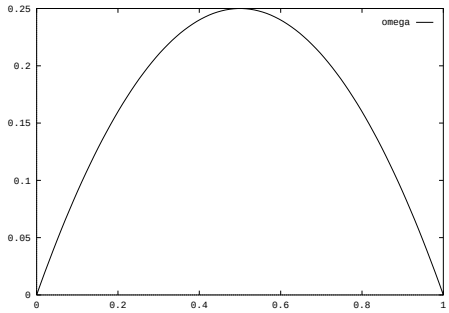


図 6: 重み ω_p

となる。ただし、 $\omega_i = y_i(1-y_i)$ である。1 次微分と 2 次微分をまるとめて行列表現すると

$$\begin{aligned} \nabla l &= \sum_{i=1}^{N} \delta_i \tilde{\boldsymbol{x}}_i = X^T \boldsymbol{\delta}, \\ \nabla^2 l &= -\sum_{i=1}^{N} \omega_i \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^T = -X^T W X \end{aligned} \tag{60}$$

となる。ただし、$X^T = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_N]$, $W = diag(\omega_1, \ldots, \omega_N)$ および $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$ である。

これらを用いて、パラメータ $\tilde{\boldsymbol{w}}$ に対する Fisher 情報行列、すなわち、Hessian 行列の期待値のマイナスは、

$$F = -E(\nabla^2 l) = X^T W X \tag{61}$$

となる。これは、入力ベクトル $\{\tilde{\boldsymbol{x}}_i\}$ の ω_i で重み付けた相関行列である。そのときの重み ω_i は、図 6 に示すような 2 次関数で、ニューロンの出力が確定している（ 0 あるいは 1 に近い）場合には小さくなり、出力が不確定な（ 0.5 に近い）場合には大きくなる。従って、Fisher 情報行列は、主に、出力が不確定な入力ベクトルの相関行列であると考えることができる。

対数尤度 (53) を最大とするようなパラメータを求めるためには、非線形最適化法を用いる必要がある。ロジスティック回帰では、このために Fisher のスコアリングアルゴリズムが使われる [18]。これは、一種のニュートン法で、Hessian 行列のかわりに Fisher 情報行列を用いる。ニューロン 1 個のみからなるネットワークの場合、Fisher 情報行列と Hessian 行列は単に符合が異なるだけなので、Fisher のスコアリングアルゴリズムはニュートン法そのものとなる。

今、現時点でのパラメータの推定値を $\boldsymbol{w}$ とし、それを修正ベクトル $\delta\tilde{\boldsymbol{w}}$ により、

$$\tilde{\boldsymbol{w}}^* = \tilde{\boldsymbol{w}} + \delta\tilde{\boldsymbol{w}} \tag{62}$$

のように更新するものとする。修正ベクトル $\delta\tilde{\boldsymbol{w}}$ は、線形方程式

$$F\delta\tilde{\boldsymbol{w}} = \nabla l \tag{63}$$

を解くことにより求められる。パラメータの更新式 (62) に左から F を掛けると、

$$F\tilde{\boldsymbol{w}}^* = F\tilde{\boldsymbol{w}} + F\delta\tilde{\boldsymbol{w}} = F\tilde{\boldsymbol{w}} + \nabla l \tag{64}$$

となる。今、 $F\tilde{\boldsymbol{w}}$ は、

$$F\tilde{\boldsymbol{w}} = X^T W \boldsymbol{\eta} \tag{65}$$

となる。ただし、$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)^T$ である。従って、新しい推定値 $\tilde{\boldsymbol{w}}^*$ は、

$$\begin{aligned} \tilde{\boldsymbol{w}}^* &= F^{-1}(F\tilde{\boldsymbol{w}} + \nabla l) \\ &= (X^T W X)^{-1}(X^T W \boldsymbol{\eta} + X^T \boldsymbol{\delta}) \\ &= (X^T W X)^{-1} X^T W(\boldsymbol{\eta} + W^{-1}\boldsymbol{\delta}) \end{aligned} \tag{66}$$

により求めることができる。ただし、$\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)^T$ である。この式は、入力データ X から目的変数 $\boldsymbol{\eta} + W^{-1}\boldsymbol{\delta}$ への重み付き最小２乗法の正規方程式とみなすことができる。従って、最尤推定値を求めるには、ある初期値からはじめて、この重み付き最小２乗法を繰り返せばよいことになる。

上記のアルゴリズムは、繰り返しアルゴリズムであるためパラメータの初期値が必要である。これは、例えば、以下のような簡単な方法で推定することが可能である。今、結合重みがすべて 0 、つまり、$\boldsymbol{w} = \boldsymbol{0}$ とする。このとき、$W = \frac{1}{4}I$, $\boldsymbol{\eta} = \boldsymbol{0}$ および $\boldsymbol{\delta} = \boldsymbol{u} - \frac{1}{2}\boldsymbol{1}$ である。従って、これらを (66) の計算式に代入すると、初期パラメータの推定値 $\tilde{\boldsymbol{w}}_0$ は、

$$\tilde{\boldsymbol{w}}_0 = 4(X^T X)^{-1} X^T (\boldsymbol{u} - \frac{1}{2}\boldsymbol{1}) \tag{67}$$

となる。これは、初期パラメータを入力から教師信号 $\boldsymbol{t} - \frac{1}{2}\boldsymbol{1}$ への線形回帰により求めることに対応している。

3.6 ロジスティック回帰のための汎化性向上の工夫

ロジスティック回帰分析の場合にも、特徴の中から予測に有効な特徴の部分集合を選び出す変数選択法は、汎化性能の高い予測モデルを構築するための有効な手段であり、先に紹介した変数撰択手法をそのままロジスティク回帰にも応用できる。変数撰択のための評価基準としても、同様に、訓練サンプル以外の汎化性能を評価するためのサンプルを用意し、予測性能を直接評価する方法、resampling 法により訓練サンプルから汎化性能を予測して評価する方法、情報量基準を用いて予測性能を評価する方法などが考えられる。

(1) Weight Decay

リッジ回帰では、2 乗誤差基準にパラメータが大きくなりすぎないようなペナルティを課した。ロジスティック回帰の場合にも、同様に、対数尤度最大化基準にパラメータが大きくなりすぎないようなペナルティを課してみよう。この場合の目的関数は、

$$\begin{aligned} Q(\tilde{\boldsymbol{w}}) &= -l + \lambda \sum_{j=1}^{M} w_j^2 \\ &= \sum_{i=1}^{N} \{\log\{1 + \exp(\eta_i)\} - u_i \eta_i\} \\ &\quad + \lambda \sum_{j=1}^{M} w_j^2 \end{aligned} \tag{68}$$

のように書ける。これを最小化するパラメータを求めるために、$Q(\tilde{\boldsymbol{w}})$ のパラメータ w_j に関する偏微分を計算してみると、

$$\frac{\partial Q}{\partial w_j} = -\frac{\partial l}{\partial w_j} + 2\lambda w_j$$

$$= -\sum_{i=1}^{N}(u_i - y_i)x_{ij} + 2\lambda w_j \tag{69}$$

となる。また、$Q(\tilde{\boldsymbol{w}})$ のパラメータ h に関する偏微分は、

$$\begin{aligned}\frac{\partial Q}{\partial h} &= -\frac{\partial l}{\partial h} \\ &= -\sum_{i=1}^{N}(u_i - y_i)(-1)\end{aligned} \tag{70}$$

となる。したがって、Weight Decay でのパラメータの更新式は、

$$w_j \Leftarrow w_j + \alpha(\sum_{i=1}^{N}(u_i - y_i)x_{ij}) - 2\alpha\lambda w_j \tag{71}$$

$$h \Leftarrow h + \alpha(\sum_{i=1}^{N}(u_i - y_i)(-1)) \tag{72}$$

となる。ここで、w_j の更新式の第 2 項は、w_j の絶対値を小さくする方向に作用する。つまり、予測に不必要な無駄なパラメータを 0 にするような効果がある。

3.7 正則化法としてのサポートベクターマシン

ソフトマージン法で識別誤りを許すようにしたサポートベクターマシンの目的関数の式 (16) を多少変形すると

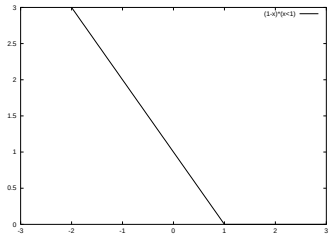
$$\begin{aligned}L(\boldsymbol{w}, \boldsymbol{\xi}) &= \sum_{i=1}^{N}\xi_i + \lambda\sum_{j=1}^{M}w_j^2 \\ &= \sum_{i=1}^{N}[1 - t_i\eta_i]_+ + \lambda\sum_{j=1}^{M}w_j^2\end{aligned} \tag{73}$$

のようになる。ここで、$[x]_+$ は、x の正の部分のみを取る関数である。図 7 (a) に、$[1-x]_+$ のグラフを示す。このグラフからもかるように、第 1 項は、$t_i\eta_i$ が 1 より大きい場合 (つまり、平面 H1 あるいは H2 に達するまで) は、ずっと 0 をとり、1 より小さくなるとしだいに大きな値をとるようになる。この評価関数で、第 1 項は、モデルとデータとの食い違いを評価する関数であり、第 2 項は、いわゆる正則化項で、パラメータに対するペナルティである。

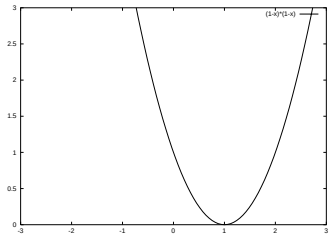
同様に、リッジ回帰の場合の評価関数も t_i の値が 0 か 1 かで場合分けして、変形すると

$$Q = \sum_{i=1}^{N}(1 - t_i\eta_i)^2 + \lambda\sum_{j=1}^{M}w_j^2 \tag{74}$$

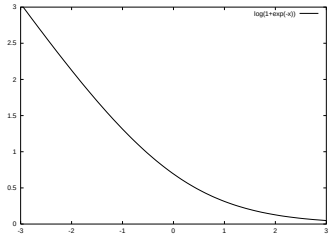
のようになる。この場合も、第 1 項は、モデルとデータとの食い違いを評価する関数であり、第 2 項は、パラメータに対するペナルティである。第 1 項の関数 $(1-x)^2$ を図 7 (b) に示す。この関数は、$t_i\eta_i$ が 1 より大きいか小さいかにかかわらず、$t_i\eta_i$ が 1 から離れるとともに大きな値を出力するようになる。この関数では、$t_i\eta_i$ が 1 以上になるような正しく識別されているようなサンプルに対しても 1 から離れるにつれて大きなペナルティを与えてしまう。これは、識別課題の場合には、最小 2 乗基準は必ずしも良くないことを意味している。



(a) サポートベクターマシン



(b) 重回帰分析 (リッジ回帰)



(c) ロジスティック回帰 (Weight Decay)

図 7: 評価基準の比較

さらに、Weight Decay を行うロジスティック回帰についても、同様に、教師信号を $u_i \in (0,1)$ から $t_i \in (-1,1)$ に変換して、評価関数を変形すると、

$$Q = \sum_{i=1}^{N} \log\{1+\exp(t_i\eta_i)\} + \lambda \sum_{j=1}^{M} w_j^2 \tag{75}$$

のように書ける。ここでも、第 1 項は、モデルとデータとの食い違いを評価する関数であり、第 2 項は、パラメータに対するペナルティである。第 1 項の関数 $\log\{1+\exp(t_i\eta_i)\}$ をグラフにすると、図 7 (c) のようになる。この関数は、サポートベクターマシンの第 1 項の関数とその形状は似ているが、$t_i\eta_i = 1$ で不連続ではない。重回帰の評価関数 (2 乗誤差基準) とは違って、$t_i\eta_i$ が 1 以上となるような正しく識別されているようなサンプルへのペナルティは小さくなる。

これらの 3 つの評価関数を比較すると、非常に良く似ていることが分かる。特に、第 2 項のパラメータに対するペナルティの入れ方は同じである。これは、汎化能力の向上の工夫としては同じものを使っていることを意味している。第 1 項の訓練データとモデルとのズレの評価方法は異なるが、サポートベクターマシンとロジスティック回帰の評価関数は非常に良く似ていることが分かる。サポートベクターマシンは、2 クラスの識別課題を前堤に導出されたが、ロジスティック回帰は、必ずしも 2 クラスの識別課題を前堤にしているわけではなく、多クラスの問題を扱うように定式化することは難しくない。

4 おわりに

本稿では、サポートベクターマシンを中心に、単純パーセプトロンタイプの識別器の学習における汎化性能を向上させるための工夫について紹介した。また、サポートベクターマシンと重回帰分析、あるいは、ロジスティック回帰分析の評価関数を比較した。こうした比較検討により、サポートベクターマシンが何をやっているのかについてのより深い理解が得られれば幸である。

参考文献

[1] V.N.Vapnik, *Statistical Learning Theory*, John Wiley & Sons (1998).

[2] 赤穂, 津田, “サポートベクターマシン—基本的仕組みと最近の発展—,” 数理科学, No.444, pp.52-58 (2000).

[3] 前田, “痛快！サポートベクトルマシン-古くて新しいパターン認識手法–,” 情報処理, Vol.42, No.7, pp.676-683 (2001).

[4] B.Scholkopf, C.J.C.Burges, A.J.Smola, *Advances in Kernel Methods - Support Vector Learning*, The MIT Press, 1999.

[5] N.Cristianini, J.S-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[6] T.Hastie, R.Tibshirani, J.Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, Springer-Verlag, 2001.

[7] R.O.Duda, P.E.Hart, D.G.Stork, *Pattern Classification* (Second Edition), John Wiley & Sons, 2001.

[8] K.R.Muller, S.Mika, G.Ratsch, K.Tsuda, B.Scholkopf, “An introduction to kernel-based learning algorithms,” IEEE Trans. On Neural Networks, Vol.12, No.2, pp.181-201, 2001.

[9] Miller,R.G.,(1974): “The jacknife -a review,” Biometrika, Vol.61, No.1, pp.1-15.

[10] Stone,M.,(1974): “Cross-validatory choice and assessment of statistic al predictions,” Journal of Royal Statistical Society, Vol.B36, pp.111-147.

[11] Efron,B.,(1979): “Bootstrap methods: anothoe look at the jackknife,” The Annals of Statistics, Vol.7, No.1, pp.1-26.

[12] Efron,B.,(1983): “Estimating the error rate of a prediction rule: imp rovements in cross-validation,” Journal of Amerian Statistical Association, Vol .78, pp.316-331.

[13] Efron,B.,(1985): “The bootstrap method for assessing statistical accu racy,” Behaviormetrika, Vol.17, pp.1-35.

[14] Akaike,H.,(1974): “A new look at the statistical model identification,” IEEE Trans. on Automatic Control, vol.AC-19, No.6, pp.716-723.

[15] 坂本, 石黒, 北川,(1983): “情報量統計学,” 共立出版.

[16] Rissanen,J.,(1983): “A universal prior for integers and estimation by minimum description length,” The Annals of Statistics, Vol.11, NO.2, pp.416-431.

[17] Rissanen,J.,(1986): “Stochastic complexity and modeling,” The Annals of Statistics, Vol.14, No.3, pp.1080-1100.

[18] P.McCullagh, and J.A.Nelder FRS, “*Generalized Linear Models*,” Chapman and Hall, 1989.