解　説

Large Margin Classifiers
— Introduction to Large Margin Classifiers —

小野田　崇　（財）電力中央研究所情報研究所
Takashi Onoda　Communication and Information Research Laboratory, Centeral Research Institute of Electric Power Industry.
onoda@criepi.denken.or.jp

Keywords: support vector machine, AdaBoost, margin, classification.

1. はじめに

最近，機械学習の研究領域で，パターン認識の一手法である Large Margin Classifiers（以下，LMC）[Smola 00] が注目を集めている．この LMC は margin の定義の違いにより，Support Vector Machines 系と AdaBoost 系の二つに大別される．本稿では，LMC の概要として，Support Vector Machines（以下，SVM）と AdaBoost とを margin の視点から紹介する．

LMC の代表格である SVM の研究は，1980 ～ 90 年代にかけて注目されたニューラルネットワークに比べて優れたパターン認識結果が報告（[Schölkopf 97] など）されて以来，数多くの理論的な研究が急速に行われてきた．本稿では，Boser, Guyon, Vapnik の [Boser 92], Guyon, Boser, Vapnik の [Boser 93], Cortes, Vapnik の [Cortes 95] および Vapnik の [Vapnik 95, Vapnik 98] に基づき，margin の視点から SVM の概要を述べる．

また，もう一つの LMC として，AdaBoost についての概要を述べる．この AdaBoost は近年，脚光を浴びている ensemble methods の一つである．ensemble methods では，Radial Basis Function ネットワークなどを基本学習アルゴリズム*1 として与えて学習を行う．ensemble methods は，学習後，誤分類した学習サンプルに注目して学習サンプル全体を重み付けし，この重み付けされた学習サンプルを基本学習アルゴリズムを用いて学習することで，学習仮説である hypothesis を生成するといった手順を繰り返す．ensemble methods においては，この繰返しによって，学習仮説である hypotheses の集合が構成される．これらの hypotheses は，重み付きの線形結合によって，最終的には一つの分類器に統合される．以下では，学習の繰返しによって生成される hypothesis を学習仮説と呼び，最終的に一つに統合される分類器を最終学習仮説と呼ぶ．

特に，Freund らが [Freund 96] で提案している AdaBoost は，幅広い適用領域，さまざまな学習法を基本学習アルゴリズムとした場合に対して，未知サンプルに対する分類誤差の改善という視点から大きな効果をあげ，多くの研究者の注目を集めている [Drucker 93, LeCun 95, Onoda 00, Schwenk 98].

以下，2 章では SVM の構成する判別関数の構造，線形 SVM，そして非線形 SVM について述べ，3 章で，AdaBoost の学習特性を margin の視点から紹介する．4 章では，LMC として紹介する SVM と AdaBoost を同じ問題に適用した際の判別面の形成のようすについて紹介する．5 章では，まとめと，LMC 研究に関連する情報の入手方法について述べる．

2. Support Vector Machines

2·1　SVM の扱う margin

SVM の中核をなす超平面判別関数の構造について考える．ここで，内積空間 F およびパターンベクトル集合 $\mathbf{z}_1, \ldots, \mathbf{z}_r$ が与えられたとすると，任意の超平面判別関数は次のように表現される．

$$\{\mathbf{z} \in F : (\mathbf{w} \cdot \mathbf{z}) + b = 0\} \qquad (1)$$

この式（1）は，自由度として係数 $\mathbf{w}$ と非負値である b をパラメータとして有している．式（1）を図示すると図 1 のようになる．図 1 は 2 次元の観測空間にデータが観測されたようすを表している．ここで，白い円と黒い

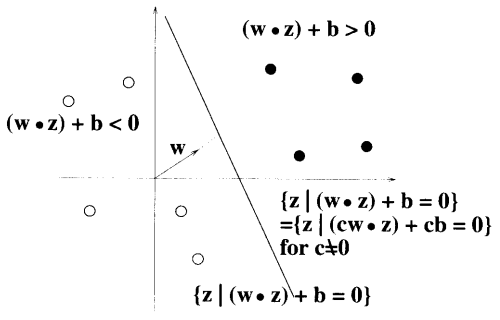

図 1　線形判別関数

*1 いくつかの論文では base hypothesis と呼ばれている．

円とを分類したいとする．しかし，式（1）からだけでは，0でない定数cを$\mathbf{w}$およびbに掛けたものすべてが，線形判別関数として導かれてしまう（図1）．

そこで，以下の式で表現される制約を加えることによって，判別関数となる超平面を定数cの掛からない$(\mathbf{w}, b)\in F\times\mathbf{R}$を有する関数になるようにする．

$$\min_{i=1,\ldots,r}|(\mathbf{w}\cdot\mathbf{z})+b|=1. \tag{2}$$

つまり，この制約によって$\mathbf{w}$とbは距離$1/\|\mathbf{w}\|$をもつ超平面に最も接近するデータ点を表現することとなる．そのようすを示したのが図2である．したがって，2値分類問題の場合，超平面間へ垂直に測った2値間の距離をmarginと呼び*2，そのmarginは，少なくとも$2/\|\mathbf{w}\|$となる．この超平面集合上の一つの構造の存在は，Vapnikの導出した以下の結果により確認することができる [Vapnik 95].

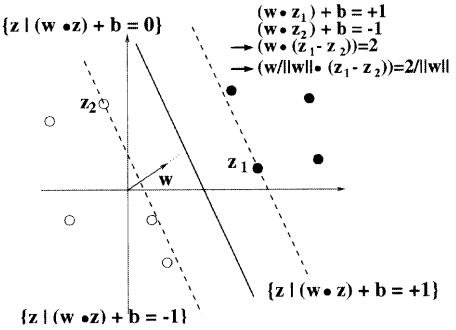

図2 制約付き線形判別関数

〈**命題1**〉 Rを点$\{\mathbf{z}_1, \ldots, \mathbf{z}_r\}$を含む最も小さい球$B_R(\boldsymbol{a})=\{\mathbf{z}\in F: \|\mathbf{z}-\mathbf{a}\|<R\}$（$\mathbf{a}\in F$）の半径とし，次式がこれらの点を定義する識別関数であるとする．

$$f_{\mathbf{w},b}=\mathrm{sgn}((\mathbf{w}\cdot\mathbf{z})+b) \tag{3}$$

そのとき，関数集合$\{f_{\mathbf{w},b}: \|\mathbf{w}\|\le A\}$は次式を満たすVC–次元$h$を有する．

$$h<R^2A^2+1. \tag{4}$$

上記定理において，条件$\|\mathbf{w}\|\le A$を省くとVC–次元がN_F+1となる関数の集合を導くことができる．ただし，N_Fは空間Fの次元を表す．つまり，条件$\|\mathbf{w}\|\le A$によりN_Fより小さなVC–次元を得ることが可能であり，結果的に高次元空間で分類問題を取り扱うことができる．

命題1は次のように解釈できる．

（1）marginと$\|\mathbf{w}\|$が反比例の関係にあるので，小さいmarginを大きくできれば，小さいVC–次元となることが式（4）からわかる．

（2）判別を小さなmarginで行う場合，より大きいクラスの分類問題を扱うことができる．

学習サンプルから高い汎化能力*3を有する学習機械を実現するには，学習サンプルに対する誤分類とVC–次元の両方を小さくする必要がある [Vapnik 95]．つまり，超平面判別関数はmarginを最大化し，同時に可能な限り学習サンプルを判別できる関数である必要がある．このmargin最大化と学習サンプルの学習については次節で述べる．

2・2 線形 Support Vector Machines

学習サンプル$(\mathbf{z}_1, y_1), \ldots, (\mathbf{z}_l, y_l)$, $z_i\in F$, $y_i\in\{\pm 1\}$が与えられ，次式を満たす判別関数$f_{\mathbf{w},b}=\mathrm{sgn}((\mathbf{w}\cdot\mathbf{z})+b)$を推定する問題を考える．

$$f_{\mathbf{w},b}(\mathbf{z}_i)=y, \quad i=1,\ldots,l. \tag{5}$$

この関数が存在すれば，式（2）の制約は次のように表現できる．

$$y_i\cdot((\mathbf{z}_i\cdot\mathbf{w})+b)\ge 1, \quad i=1,\ldots,l. \tag{6}$$

$(\mathbf{w}, b)$, $(-\mathbf{w}, -b)$のように$\mathbf{w}$とbの方向の違いにより，同じ超平面判別関数の式が二つ存在することとなる．しかし，式（5）と式（6）によって判別関数は一意に定めることができる．

命題1によって，汎化能力の高い判別関数は式（6）で表現される制約条件のもと，次式を最小化することで推定できる．

$$\tau(\mathbf{w})=\frac{1}{2}\|\mathbf{w}\|^2. \tag{7}$$

この凸最適化問題を解くため，式（7）のLagrangianを計算すると

$$L\left(\mathbf{w},b,\vec{\alpha}\right)$$

$$=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}\alpha_i\left(y_i((\mathbf{z}_i\cdot\mathbf{w})+b)-1\right) \tag{8}$$

ここで，$\alpha_i\ge 0$はLagrange乗数である．このLagrangianをα_iについて最大化し，$\mathbf{w}$とbについて最小化する．パラメータ$\mathbf{w}$とbについてのLの導関数は鞍点において次式のように0にならなければならないので，

$$\frac{\partial}{\partial b}L\left(\mathbf{w},b,\vec{\alpha}\right)=0, \quad \frac{\partial}{\partial\mathbf{w}}L\left(\mathbf{w},b,\vec{\alpha}\right)=0. \tag{9}$$

式（9）から次式が成立する．

$$\sum_{i=1}^{l}\alpha_i y_i=0, \tag{10}$$

$$\mathbf{w}=\sum_{i=1}^{l}\alpha_i y_i\mathbf{z}_i \tag{11}$$

結局，$\mathbf{w}$は学習サンプルの展開式となる．$\mathbf{w}$の解はた

*2 以下，「margin上」という場合は，marginを測定する際の超平面上を意味する．

*3 未学習サンプルの判別に対しても高い判別能力をもつということ．

だ一つに決まるが，係数 α_i はその必要がない．

Karush-Kuhn-Tucker 条件により，鞍点において Lagrange 乗数 α_i は，式 (6) を正確に表現し直した次式の制約条件に対して非ゼロでなくてはならない．

$$\alpha_i \cdot [y_i((\mathbf{z}_i \cdot \mathbf{w}) + b) - 1] = 0$$
$$i = 1, ..., l. \qquad (12)$$

$\alpha_i > 0$ を有するパターン $\mathbf{z}_i$ を *Support Vectors* と呼ぶ．式 (12) より，*Support Vectors* は margin 上に存在することとなる．*Support Vectors* 以外の学習サンプルは凸最適化問題の解法には関係のないものとなる．つまり，*Support Vectors* 以外の学習サンプルは式 (6) の制約条件を自動的に満たし，式 (11) の展開項の部分には現れないのである．

この凸最適化問題を解いて得られる超平面判別関数の汎化能力については，以下の命題が成立する [Vapnik 98]．

〈**命題 2**〉 サンプル数 l の学習サンプルから得られる *Support Vectors* 数の期待値を $l-1$ で割った値は，未学習サンプルに対する誤分類率の上限である．

式 (8) の Lagrangian に式 (10)，式 (11) の条件を代入すると，双対問題となる次の凸最適化問題を得ることができる．

$$\max_{\vec{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{z}_i \cdot \mathbf{z}_j)$$
$$\text{subject to } \alpha_i \geq 0,\ i = 1, ..., l,$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0. \qquad (13)$$

式 (11) の展開式を判別関数の式 (5) に代入することによって，式 (5) の判別関数を，分類されるパターンと *Support Vectors* との内積で評価される次式に書き換えることができる．

$$f(\mathbf{z}) = \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i y_i (\mathbf{z} \cdot \mathbf{z}_i) + b \right) \qquad (14)$$

以上より，式 (13) で表現される凸二次計画問題を解くことで，判別関数 $f_{\mathbf{w},b}(\mathbf{z}) = \mathrm{sgn}((\mathbf{w} \cdot \mathbf{z}) + b)$ を得ることができる．これが基本となる線形 SVM である．

現実問題としては，学習サンプルを完全に分離できる超平面は存在しない場合が多い．そのような場合，次式で表現される緩和変数を導入して，式 (6) を満たさない学習サンプルが存在しても良いようにする [Cortes 95]．

$$\xi_i \geq 0, \quad i = 1, ..., l. \qquad (15)$$

この緩和変数を使って式 (6) の制約条件を次式のように緩和できる．

$$y_i((\mathbf{z}_i \cdot \mathbf{w}) + b) \geq 1 - \xi_i, \quad i = 1, ..., l. \qquad (16)$$

この緩和変数の導入によって，式 (7) と式 (6) で表現される凸最適化問題が次式のようになる．

$$\min_{\mathbf{w}, \vec{\xi}} \tau(\mathbf{w}, \vec{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + \gamma \sum_{i=1}^{l} \xi_i$$
$$\text{subject to } y_i((\mathbf{z}_i \cdot \mathbf{w}) + b) \geq 1 - \xi_i$$
$$i = 1, ..., l. \qquad (17)$$

目的関数の右辺第一項は，判別関数クラスの VC-次元の最小化に関連することが式 (4) よりわかる．一方，$\Sigma_{i=1}^{l} \xi_i$ は学習サンプル中で誤分類されるパターンの上限値である．適切な正定数 γ を選択できるとすれば，式 (17) で表現される凸最適化問題は，任意の関数集合における，Vapnik の提唱する Structual Risk Minimization の概念を実践することとなる [Vapnik 95]．

学習サンプルが完全に分離できる場合の式 (11) と同様に，式 (17) の最適解において，$\mathbf{w}$ は次式のように，学習サンプルの展開式となる．

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{z}_i. \qquad (18)$$

ここで，係数 α_i が非ゼロとなるのは，学習サンプル $(\mathbf{z}_i, y_i)$ が制約条件式 (16) を満たす場合である．式 (17) で表現される最適化問題の双対問題となる以下の凸二次計画問題を解くことで，係数 α_i を求めることができる．

$$\max_{\vec{\alpha}} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{z}_i \cdot \mathbf{z}_j)$$
$$\text{subject to } 0 \leq \alpha_i \leq \gamma,\ i = 1, ..., l,$$
$$\sum_{i=1}^{l} \alpha_i y_i = 0. \qquad (19)$$

Karush-Kuhn-Tucker 条件から，式 (19) で表現される凸二次計画問題の最適解は次の条件を満たす．

$$\alpha_i = 0 \Rightarrow y_i f(\mathbf{z}_i) \geq 1$$
$$0 \leq \alpha_i \leq \gamma \Rightarrow y_i f(\mathbf{z}_i) = 0$$
$$\alpha_i = \gamma \Rightarrow y_i f(\mathbf{z}_i) \leq 1 \qquad (20)$$

この条件より，分類結果 $\mathrm{sgn}(f(\mathbf{z}_i))$ が y_i と一致していて，margin 値 $y_i f(\mathbf{z}_i)$ が 1 より大きいサンプルに対応する α_i は 0 になることがわかる．

2·3 非線形 Support Vector Machines

2·2 節では線形 SVM について述べた．しかし，線形 SVM は線形分離可能な場合には高い汎化能力を達成できるが，実際の問題では線形分離可能な場合は多くない．そこで，より一般的な判別関数を推定するため，前処理として入力ベクトル $\mathbf{x}_1, ..., \mathbf{x}_l$ を次式のように高次元特徴空間に写像し，その後，その特徴空間で線形 SVM を行うという方法が考えられる．

$$\Phi: \mathbf{x}_i \mapsto \mathbf{z}_i. \qquad (21)$$

本章で用いる $\mathbf{z}_i$ と 2·2 節で用いた $\mathbf{z}_i$ とは異なり，$\mathbf{z}_i$ は観測された入力パターン $\mathbf{x}_i$ を高次元特徴空間に写像し

た結果であることに注意されたい．

式（19）で表現される凸二次計画問題の目的関数を最大化し，式（14）で表現される判別関数を推定するには，高次元空間での以下の内積を計算する必要がある．

$$(\Phi(\mathbf{x})\cdot\Phi(\mathbf{x}_i)). \tag{22}$$

式（22）で表現される内積の計算には膨大な計算が必要となる．Mercerの条件の下，もとの観測空間で定義される次式を満たすkernel関数を用いて，膨大な計算を削減できる．

$$(\Phi(\mathbf{x})\cdot\Phi(\mathbf{x}_i))=k(\mathbf{x},\mathbf{x}_i). \tag{23}$$

このkernel関数を用いると，高次元特徴空間での式（14）に相当する判別関数は次のようになる．

$$f(\mathbf{x})=\mathrm{sgn}\left(\sum_{i=1}^{l} y_i\alpha_i\cdot k(\mathbf{x},\mathbf{x}_i)+b\right). \tag{24}$$

結局，ユークリッド空間の内積に代わって，適切なkernel関数kを選択できれば，このkernel関数kに基づく非線形SVMには，前節で述べた線形SVMの特性がすべて適用できる．図3に非線形SVMの原理を示す．図3では，入力空間（ここでは$\mathbf{R}^2$）上のデータ（上図左）を非線形の写像を使ってより高次の特徴空間（ここでは$\mathbf{R}^3$）にマッピングし，特徴空間上で分離可能な超平面を作成することで（下図左），入力空間では非線形の判別関数になる（下図右）．図3上図右は，SVM構成できるようすを示している．

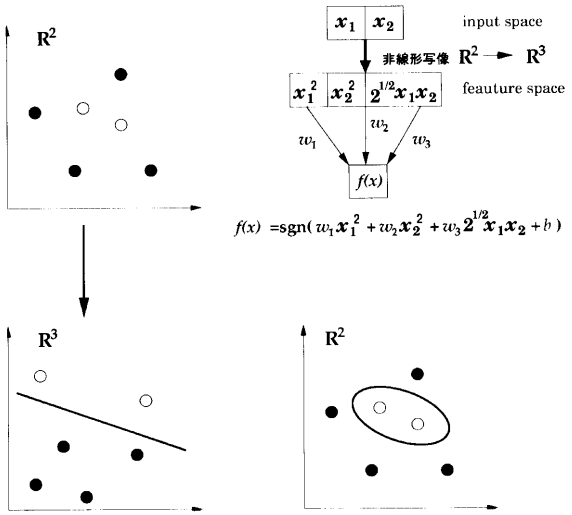

図3 非線形SVMの原理

また，非線形SVMはさまざまなkernel関数を利用して，多様な学習機械を構成できる．図4に非線形SVMの構造を示した．図4中のkernel関数kには，以下で述べるようなkernel関数を利用することが可能である．また，SVMを構成するためのパラメータは凸二次計画問題を解くことで推定することが可能である．図4中の第一層の$\mathbf{x}_i$は学習サンプル集合の部分集合（*Support Vectors*）であり，第二層のγはLagrange乗数α_iから$\gamma=y_i\alpha_i$で計算できる．以下に一般的に利用されているkernel関数を紹介しておく．

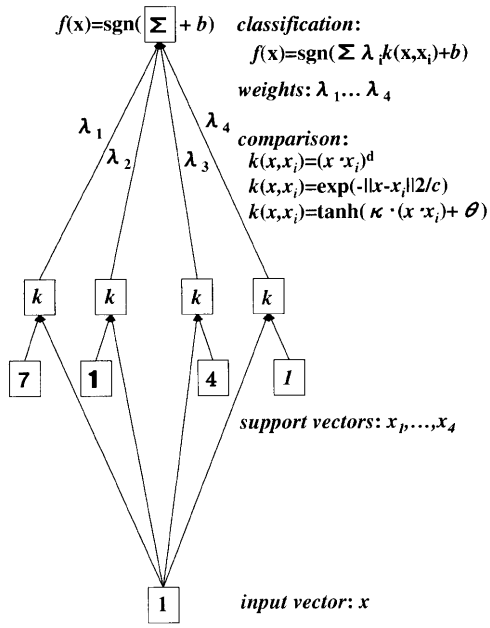

図4 非線形SVMの構造

Polynomial kernel関数

$$k(\mathbf{x},\mathbf{x}_i)=(\mathbf{x},\mathbf{x}_i)^d. \tag{25}$$

Radial basis kernel関数

$$k(\mathbf{x},\mathbf{x}_i)=\frac{\exp(-\|\mathbf{x}-\mathbf{x}_i\|)^2}{c}. \tag{26}$$

Sigmoid kernel関数

$$k(\mathbf{x},\mathbf{x}_i)=\tanh(\kappa\cdot(\mathbf{x},\mathbf{x}_i)+\theta). \tag{27}$$

式（24）で表現される判別関数を求めるには，以下の最適化問題を解けばよい．

$$\max_{\vec{\alpha}}\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i\cdot\mathbf{x}_j)$$

$$\text{subject to } 0\le\alpha_i\ge\gamma,\ i=1,\ldots,l,$$

$$\sum_{i,j=1}^{l}\alpha_i y_i=0. \tag{28}$$

kernel関数kはMercerの条件を満たす必要がある．つまり，式（23）のように高次元特徴空間での内積にkernel関数kは一致する必要がある．このことから，$K_{ij}=(y_iy_jk(\mathbf{x}_i,\mathbf{x}_j))_{ij}$は正値行列となる．つまり，

$$\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i\cdot\mathbf{x}_j)$$

$$=\left(\sum_{i=1}^{l}\alpha_i y_i\Phi(\mathbf{x}_i)\cdot\sum_{j=1}^{l}\alpha_j y_j\Phi(\mathbf{x}_j)\right)\ge 0. \tag{29}$$

式（16）から$\xi_j=0$である*Support Vectors* $\mathbf{x}_j$に対し，次式が成立する．

$$\sum_{i=1}^{l} y_i \alpha_i \cdot k(\mathbf{x}_i \cdot \mathbf{x}_j) + b = 0. \tag{30}$$

以上より，変数 b は次式のように，$0 \leq \alpha_j \leq \gamma$ となる α_j を有する *Support Vectors* の平均によって得ることができる．

$$b = y_j - \sum_{i=1}^{l} y_i \alpha_i \cdot k(\mathbf{x}_i \cdot \mathbf{x}_j). \tag{31}$$

3. AdaBoost

一般的な *ensemble methods* は次のようになる．$\{h_t(\mathbf{x}): t=1, \cdots, T\}$ を入力ベクトル $\mathbf{x}$ と重みベクトル $\mathbf{c}=[c_1 \cdots c_T]$ で定義される T 個の学習仮説の集合であるとする．ここで，ベクトル $\mathbf{c}$ は $c_t > 0$ かつ $\sum_{t=1}^{T} c_t = 1$ を満たしている．また，T はあらかじめ与えられた学習仮説生成回数である．2 値分類問題においては，その出力が二つの分類ラベルのうちの一つになる．つまり，$h_t(\mathbf{x}) = \pm 1$ となる．この学習仮説の集合は以下のように重み付けされた多数決による分類ラベルを生成する．

$$\mathrm{sign}\left(\sum_{t=1}^{T} c_t h_t(\mathbf{x}) \right) \tag{32}$$

T 個の学習仮説 $\{h_t(\mathbf{x})\}$ の集合と $\mathbf{c}$ を求めるために，いくつかのアルゴリズムが提案されている．例えば，Breiman が [Breiman 96] および [Breiman 97] で提案している Bagging，Arcing や Schapire らが [Schapire 98] で提案している AdaBoost がある．

特に，Schapire らが [Schapire 98] で提案している AdaBoost は，margin の概念を導入し，幅広い適用領域で大きな効果をあげ，多くの研究者の注目を集めている [Drucker 93, LeCun 95, Onoda 00, Schwenk 98]．本章では SVM とは違うもう一つの LMC として，2 値分類問題における AdaBoost の概要を紹介する．

3·1 AdaBoost アルゴリズム

2 値分類問題の場合，一つの入出力の組 $\mathbf{z}_i = (y_i, \mathbf{x}_i)$，$(i=1, ..., N)$ に対して AdaBoost では margin $\rho(\mathbf{z}_i, \mathbf{c})$ を以下のように定義する．

$$\rho(\mathbf{z}_i, \mathbf{c}) = y_i \sum_{t=1}^{T} c_t h_t(\mathbf{x}_i),$$

$$\rho(\mathbf{z}_i, \mathbf{c}) \in [-1, 1]. \tag{33}$$

ここで，$y_i = \pm 1$ は分類ラベルであり，$\mathbf{x}_i$ は入力ベクトルを表す．式 (33) において，入出力の組 $\mathbf{z}_i$ に対してその margin の値が正であれば，式 (32) より正しい分類ラベルが予測されたことになる．つまり，margin の値が大きくなれば，分類の正しさが増加することになる．また，誤分類率 $d(\mathbf{z}_i, \mathbf{c})$ を次式で定義する．

$$d(\mathbf{z}_i, \mathbf{c}) = \sum_{t=1}^{T} c_t I(y_i \neq h_t(\mathbf{x}_i))$$

$$= \left(\frac{1}{2} - \frac{\rho(\mathbf{z}_i, \mathbf{c})}{2} \right) \tag{34}$$

ここで，$I(y_i \neq h_t(\mathbf{x}_i))$ は $I(true)$ ならば 1，$I(false)$ ならば 0 となる損失関数である．

AdaBoost の学習は，次に示す誤分類率 $d(\mathbf{z}_i, \mathbf{c})$ の関数 $g(\mathbf{b})$ を b_t に関して最小化する過程である [Breiman 97]．

$$g(\mathbf{b}) = \sum_{i} \exp\{|\mathbf{b}|(d(\mathbf{z}_i, \mathbf{c}) - \phi)\}. \tag{35}$$

ここで，$|\mathbf{b}| = \sum_{t=1}^{T} b_t > 0$ である．この b_t は学習仮説 h_t の非正規化重み付けである．最終学習仮説の重み $\mathbf{c}$ は $\mathbf{b}$ を次のように正規化して求められる．

$$\mathbf{c} = \frac{\mathbf{b}}{|\mathbf{b}|}. \tag{36}$$

AdaBoost(ϕ) は $\phi \in [0, 1]$ となるパラメータ ϕ をもち，$\phi = 1/2$ のとき，Schapire らが [Schapire 98] で提案した AdaBoost と一致する．

以下，ある時点での学習仮説生成回数を t とする．$t+1$ 回目の学習仮説 h_{t+1} を生成するために，学習サンプル $\mathbf{z}_i$ が t 回の学習仮説生成の後，$w_{t+1}(\mathbf{z}_i)$ で重み付けされる．この重み $w_{t+1}(\mathbf{z}_i)$ は，次式に従って更新される．

$$w_{t+1}(\mathbf{z}_i) = \frac{w_t(\mathbf{z}_i) \exp\{-b_t I(h_t(\mathbf{x}_i) = y_i)\}}{\sum_{j=1}^{N} w_t(\mathbf{z}_j) \exp\{-b_t I(h_t(\mathbf{x}_j) = y_j)\}} \tag{37}$$

また，h_t の学習誤差 ε_t は

$$\varepsilon_t = \sum_{i=1}^{N} w_t(\mathbf{z}_i) I(y_i \neq h_t(\mathbf{x}_i)) \tag{38}$$

で計算される．

その際，$g(\mathbf{b})$ は b_t を次のように選ぶことによって最小化できる [Breiman 97]．

$$b_t = \log \frac{\varepsilon_t}{1-\varepsilon_t} - \log \frac{\phi_t}{1-\phi_t}. \tag{39}$$

Input: N examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
Initialize: $w_1(\mathbf{z}_i) = 1/N$ **for all** $i = 1 \ldots N$
Do for $t = 1, \ldots, T$,
(1) Train a base learner with respect to weighted example distribution w_t and obtain hypothesis $h_t : \mathbf{x} \mapsto \{-1, 1\}$
(2) calculate the training error ϵ_t of h_t: $\epsilon_t = \sum_{i=1}^{N} w_t(\mathbf{z}_i) I(h_t(\mathbf{x}_i) \neq y_i)$.
(3) set $b_t = \log \frac{\epsilon_t}{1-\epsilon_t} - \log \frac{\phi}{1-\phi}$.
(4) update example distribution w_t: $w_{t+1}(\mathbf{z}_i) = \frac{w_t(\mathbf{z}_i) \exp\{-b_t I(h_t(\mathbf{x}_i) = y_i)\}}{\sum_{j=1}^{N} w_t(\mathbf{z}_j) \exp\{-b_t I(h_t(\mathbf{x}_j) = y_j)\}}$
Output: final hypothesis: $f(\mathbf{x}) = \frac{1}{|\mathbf{b}|} \sum_{t=1}^{T} b_t h_t(\mathbf{x})$

図 5 アルゴリズム AdaBoost(ϕ)

これらの手順をアルゴリズム化したのが図5である*4.

3・2 AdaBoostの振舞い

以下では最小化される式（35）で表現される$g(\mathbf{b})$を損失関数と呼ぶ．興味深いことに，重み$w_{t+1}(\mathbf{z}_i)$について以下の補題が成立する[Onoda 98].

[補題1] t番目の学習仮説における学習サンプルの重み付け$w_{t+1}(\mathbf{z}_i)$は，marginに対する損失関数$g(\mathbf{b})$の規格化したgradientと等価である．つまり，次のように表現できる．

$$w_{t+1}(\mathbf{z}_i)=\frac{\partial g(\mathbf{b}_t)}{\partial \rho(\mathbf{z}_i,\mathbf{b}_t)}\Big/\sum_j \frac{\partial g(\mathbf{b}_t)}{\partial \rho(\mathbf{z}_j,\mathbf{b}_t)}. \quad (40)$$

ここで，$\rho(\mathbf{z}_i,\mathbf{b}_t)=|\mathbf{b}_t|\rho(\mathbf{z}_i,\mathbf{c}_t)$であり，$\mathbf{b}_t=[b_1\cdots b_t]$，$\mathbf{c}_t=[c_1\cdots c_t]$，$|\mathbf{c}_t|=\sum_{r=1}^{t}c_r$，$|\mathbf{b}_t|=\sum_{r=1}^{t}b_r$である．

この重み$w_t(\mathbf{z}_i)$によって，損失関数$g(\mathbf{b}_t)$を最小化する真の学習仮説h_t^*を近似する学習仮説が生成される．

この補題より次のことがわかる．AdaBoostは，marginに対する損失関数$g(\mathbf{b})$の勾配方向を計算し，その後，損失関数$g(\mathbf{b})$を最小化するようにb_tを決定する．これは，基本学習アルゴリズムの与える仮説空間でのgradient descent法と一致する．つまり，AdaBoostは学習仮説の集約によって，損失関数$g(\mathbf{b})$をmargin ρをパラメータとして，最小化するように動作するのである．

3・3 AdaBoostの漸近特性

本節では，margin ρの視点からAdaBoostの漸近特性について述べる．

式（34），式（35）および[補題1]より学習サンプルの重み付けの式を次のように書き換えることができる．

$$w_{t+1}(\mathbf{z}_i)=\frac{\exp\{-|\mathbf{b}_t|\rho(\mathbf{z}_i,\mathbf{c}_t)/2\}}{\sum_j \exp\{-|\mathbf{b}_t|\rho(\mathbf{z}_j,\mathbf{c}_t)/2\}}. \quad (41)$$

式（41）を詳細に観察してみると，AdaBoostの学習がパラメータ$|\mathbf{b}_t|$を有するsoftmax関数[Bishop 95]となっていることがわかる．したがって，パラメータ$|\mathbf{b}_t|$が小さい場合，すべての学習サンプルにほぼ均等な重み付けが行われる．パラメータ$|\mathbf{b}_t|$が大きくなるに従って，marginの小さな，言い換えれば，分類の難しい学習パターンに大きな重み付けが行われることになる．その極限では最も分類の難しい，つまり，最も小さいmarginをもつ学習パターンのみを考慮するようになる．このパラメータ$|\mathbf{b}_t|$については以下の補題が成立する[Onoda 98].

[補題2] $0<\phi<1$の関係を満たすパラメータϕを有するAdaBoostの学習過程において，重み付けされた学習誤差ε_tのおのおのが$\varepsilon_t\le\phi-\delta$の条件を満たしているとすると，パラメータ$|\mathbf{b}_t|$は繰返し回数$t$に対して少なくとも線形に増加する．ただし，$\delta$は$0<\delta<\phi$にあるとする．

ある程度表現力の豊かな学習モデルを適用すれば，**[補題2]**中のすべての学習誤差ε_tについて$\varepsilon_t\le\phi-\delta$が成立するという条件は，現実の分類問題において妥当な条件であると考えられる．図6に，4章の数値シミュレーションに用いた分類問題における$|\mathbf{b}|$と学習仮説生成繰返し回数tとの関係を示す．

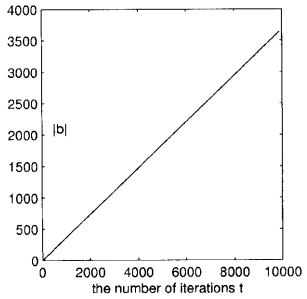

図6 $|\mathbf{b}|$と学習仮説生成繰返し回数tとの関係

式（39）より，学習誤差ε_tが小さい場合，$\mathbf{b}_t$は大きな値となる．つまり，AdaBoostの基本学習アルゴリズムが非常に高いモデル表現能力を有する場合は，学習誤差ε_tを非常に小さくすることができ，わずか数回，学習仮説の生成を繰り返すことで$|\mathbf{b}_t|$の値を大きくすることが可能である．また，$\phi>\varepsilon_t$という制約条件のもとで，パラメータϕの値を小さくすれば，学習が収束に向かう速さを抑えることができる．つまり，基本学習アルゴリズムのモデル表現能力とパラメータϕの値によって学習速度が決められるのである．

以下では，この学習速度とAdaBoostの形成するmarginとの関係について解説する．

Freundらによる[Freund 97]中の定理5から，以下の系が導ける[Onoda 98].

（系1） AdaBoostは漸近的に，つまり，$T\to\infty$で，次の不等式を満たすmargin ρを有するmargin分布を形成する．

$$\rho\ge\frac{\ln\left(\phi\varepsilon^{-1}\right)+\ln\left((1-\phi)(1-\varepsilon)^{-1}\right)}{\ln\left(\phi\varepsilon^{-1}\right)-\ln\left((1-\phi)(1-\varepsilon)^{-1}\right)}, \quad (42)$$

ここで，$\varepsilon=\max_t\varepsilon_t$であり，$\varepsilon\le(1-\rho)/2$を満たしているとする．

（系1）は，漸近的に，つまり$T\to\infty$で，すべての学習サンプルに対するmarginが不等式（42）を満たしていることを意味している．つまり，各学習サンプルのmargin ρのうち，最も小さいmargin ρ_{min}も不等式（42）を満たすような値になることを意味し，最小margin ρ_{min}の下界がϕとεに依存することを示している．

また，式（42）はmargin ρに関するϕとεとの相互

*4 Schapireらが[Schapire 98]で提案しているAdaBoostアルゴリズムでは$\phi=1/2$となるが，ここではϕを明示的に残している．本解説では，このアルゴリズムをAdaBoostと呼ぶことにする．

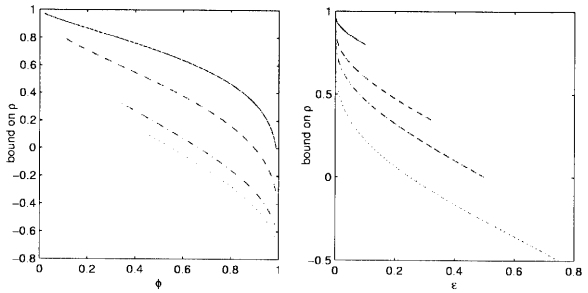

図 7　左図: $\varepsilon = 1/100$（実線），1/10（破線），1/3（一点鎖線），9/20（点線）における ρ と ϕ の関係，右図: $\phi = 1/10$（実線），1/3（破線），1/2（一点鎖線），3/4（点線）における ρ と ε の関係

関係を示している．図 7 に式 (42) から導かれる margin ρ と ϕ の関係および margin ρ と ε の関係を示す．

式 (42) および図 7 から ϕ を小さい値にすれば，最小 margin をより大きくできるという意味で，より大きな margin を生成できることがわかる．また，基本アルゴリズムのモデル表現能力をあげることで学習誤差 ε_t を小さくし，margin ρ の値を大きくし，最小 margin ρ_{min} をより大きくすることが可能である．

以上までの AdaBoost における margin にかかわる特徴をまとめると，次のようになる．

(1) 分類の難しい，判別面に近い学習サンプルの margin は，漸近的に，式 (42) を満たすある最小 margin ρ_{min} に収束する．

(2) この最小 margin ρ_{min} の下界は $|\mathbf{b}|$ の増加速度に依存する．

(3) この $|\mathbf{b}|$ の増加速度はパラメータ ϕ と学習誤差 ε_t の値を決める基本学習アルゴリズムのモデル表現能力に依存する．

結局，AdaBoost は最小 margin ρ_{min} の値を，できるだけ大きくするように動作するのである．

4. Support Vector Machines vs. AdaBoost

本章では，LMC として述べてきた SVMs と AdaBoost を，簡単な分類問題に適用した際の結果について述べるとともに，AdaBoost の margin 分布について紹介する．

図 8 に，利用した学習サンプルとそのサンプルに SVM および AdaBoost を適用して得られる判別面を示す．図 8 の学習サンプルは数種の円形正規分布で構成され，入力雑音として一様分布 $U(0, \sigma^2)$ で生成される雑音を有する．また，学習サンプル数は 300 である*5.

図 8 の上図に，kernel 関数として Radial Basis Function を用いた SVM を適用して得られる判別面と Support Vectors を示す．また，図 8 の下図に，13 のセンターを有する Radial Basis Function ネットワーク [Bishop 95] を基本学習アルゴリズムとした AdaBoost を適用して得られる判別面と判別の困難なサンプルを示す．ここでは，13 のセンターを有する Radial Basis Function（RBF）ネットワーク [Bishop 95] を AdaBoost の基本学習アルゴリズムとして用いた．

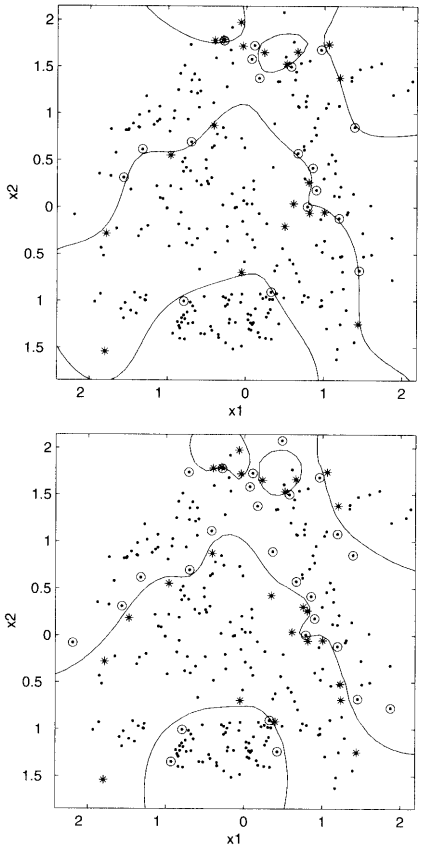

図 8　上図: 学習サンプルおよび SVM（RBF kernel 関数を使用）によって求められた判別面．●は学習サンプルを表し，＊は正の境界上のサンプル，◉は負の境界上のサンプルを表す．
下図: AdaBoost によって求められた判別面（13 のセンターを有する RBF ネットワークを学習機械として使用）．●は学習サンプルを表し，＊は正の境界上のサンプル，◉は負の境界上のサンプルを表す

図 8 上図と下図とを比較すると，その判別面および AdaBoost で強調される判別の難しいサンプルと SVM が生成する *Support Vectors* とが，非常に良く似ていることが観察できる．

この学習サンプルに AdaBoost を適用して得られる margin 分布を図 9 に示す．本シミュレーションでは $\phi = 1/2$ としている．図 9 は，13 のセンターを有する RBF ネットワークを使用し，学習サンプルの雑音 σ^2 を 0 %，9 %，16 % と変化させた場合の margin 分布である．本シミュレーションでは漸近特性が現れるように 10^4 回の学習仮説生成繰返しを行っている．

図 9 より，margin の分布が，いくつかの学習サンプルに対して，一つの決まった margin ρ_{min} の値でステップを形成することが確認できる．これは最小 margin ρ_{min} の最大化を図っていることとなり，その意味で

*5 このデータは，http://ida.first.gmd.de/~raetsch/data/benchmarks.htm から入手できる．

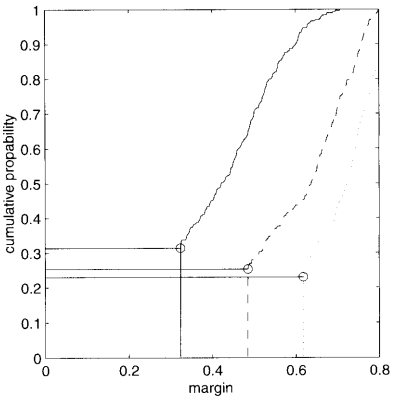

図9 雑音レベル $\sigma^2 = 0$ %（点線），9 %（破線），16 %（実線）に対して，学習機械を13のセンターを有するRBFネットワークとした場合のAdaBoostアルゴリズムが生成するmargin分布（繰返し回数10^4）

AdaBoostがmarginの最大化を図るアルゴリズムであることが観察できる．

5. おわりに

本稿で紹介したSVM，AdaBoostの学習の解析には，数理計画法が用いられている．特に，線形計画問題や，凸二次計画問題のように，確実に大局的最適解を求められる数理計画法が，学習の解析に好まれている．これは，かつてニューラルネットワークにおいて，目的関数が複雑になりすぎ，深刻な局所解の問題を生じたことの反省からである．数理計画法は，勾配降下法に比べて，理論的に明確であり，また，KKT条件など，数理計画法特有の道具を用いて，学習の改善などの問題に取り組むこともできる．しかし，最適化理論の立場から機械学習を扱う研究は，まだ未開拓であり，さまざまな研究課題が残されている．

また，SVM，AdaBoostに代表されるLMCを人工知能研究の立場から見ると，LMCはその起源が統計的機械学習にあるものの，ニューラルネットワークのように学習系自体が複雑であった手法の反省から，marginという概念を導入することによって，*Support Vectors*のように，判別のために必要となる情報を，非常に人間にわかりやすく提供する仕組みを実現しているとみることができる．しかし，人工知能研究の立場からLMCを扱う研究も，未だ未開拓であるので，さまざまな研究課題が考えられ，今後は，人工知能研究の専門家のLMC分野への貢献がますます重要になるであろうと思われる．

本稿での解説はLMCであるSVM，AdaBoostのさわりに過ぎない．ほかの日本語で読めるSVMに関する解説としては，参考文献 [前田01, 津田00] がある．SVM関連の研究についての詳細を知りたい方は，`www.kernel-machines.org`を参照されたい．このホームページにはSVMに関する研究の論文やSVM用の最適化手法のプログラムなどが掲載されている．また，AdaBoostに関連する研究の詳細について知りたい方は，`www.boosting.org`を参照されたい．このホームページには，最近のAdaBoostに関する研究の論文などが掲載されている．

◇ 参考文献 ◇

[Bishop 95] Bishop, C.: *Neural Networks for Pattern Recognition,* Clarendon Press, Oxford (1995)

[Boser 92] Boser, B., Guyon, I. and Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers, in Haussler, D., ed., *5th Annual ACM Workshop on COLT,* pp. 144-152, ACM Press, Pittsburgh, PA (1992)

[Boser 93] Boser, B., Guyon, I. and Vapnik, V.: Automatic capacity tuning of very large VC-dimension classifiers, in Hanson, S. J., Cowan, J. D. and Giles, C. L., eds., *Advances in Neural Information Processing Systems 5,* pp. 147-155, Morgan Kaufmann, San Mateo, CA (1993)

[Breiman 96] Breiman, L.: Bagging predictors, *Machine Learning,* Vol. 26, No. 2, pp. 123-140 (1996)

[Breiman 97] Breiman, L.: Prediction Games and Arcing Algorithms, Technical Report 504, Statistics Department, University of California (1997)

[Cortes 95] Cortes, C. and Vapnik, V.: Support Vector Networks, *Machine Learning,* Vol. 20, pp. 273-297 (1995)

[Drucker 93] Drucker, H., Schapire, R. and Simard, P.: Boosting performance in neural networks, *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 7, pp. 705-719 (1993)

[Freund 97] Freund, Y. and Schapire, R.: A Decision-theoretic Generalization of On-line Learning and an Application to Boosting, *Jounal of Computer and System Sciences,* Vol. 55, No. 1, pp. 119-139 (1997)

[Freund 96] Freund, Y. and Schapire, R.: Experiments with a new Boosting algorithm, in *Proc. 13th International Conference on Machine Learning,* pp. 148-146, Morgan Kaufmann (1996)

[LeCun 95] LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Müller, U., Säckinger, E., Simard, P. and Vapnik, V.: Learning Algorithms for Classification: A Comparison on Handwritten Digit Recognition, *Neural Networks,* pp. 261-276 (1995)

[前田01] 前田英作: 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, 情報処理学会誌, Vol. 42, No. 7, pp. 676-683 (2001)

[Onoda 98] Onoda, T., Rätsch, G. and Müller, K. -R.: An asymptotic analysis of AdaBoost in the binary classification case, in *Proceedings ICANN'98, Int. Conf. on Artificial Neural Networks,* Vol. VI of LNCS, Springer, Berlin (1998), In Press.

[Onoda 00] Onoda, T., Rätsch, G. and Müller, K.-R.: Applying Support Vector Machines and Boosting to a Non-Intrusive Monitoring System for Household Electric Appliances with Inverters, in *Proceedings NC'2000, Second Int. ICSC Symposium on NEURAL COMPUTATION* (2000)

[Schapire 98] Schapire, R., Freund, Y., Bartlett, P. and Lee, W.: Boosting the margin: A new explanation for the effectiveness of voting methods, *Annals of Statistics,* Vol.26, No. 5, pp. 1651-1686 (1998)

[Schölkopf 97] Schölkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V.: Comparing support vector machines with Gaussian kernels to radial basis function classifiers, *IEEE Trans. Sign. Processing,* Vol. 45, pp. 2758-2765 (1997)

[Schwenk 98] Schwenk, H. and Bengio, Y.: Training methods for adaptive boosting of neural networks, *Advances in Neural Information processing Systems,* Vol. 10, (1998)

[Smola 00] Smola, A., Bartlett, P., Schölkopf, B. and Schuurmans, D., eds.: *Advances in Large Margin Classifiers,*

The MIT Press (2000)
[津田 00] 津田宏治: サポートベクトルマシンとは何か,電子情報通信学会誌, Vol. 83, pp. 460-466 (2000)
[Vapnik 95] Vapnik, V.: *The Nature of Statistical Learning Theory,* Springer (1995)
[Vapnik 98] Vapnik, V.: *Statistical Learning Theory,* Wiley, New York (1998)

2001年10月31日 受理

著 者 紹 介

小野田 崇 (正会員)

1986年3月国際基督教大学教養学部理学科卒業. 1988年3月東京工業大学理工学研究科原子核工学専攻修士課程修了. 同年4月 (財) 電力中央研究所入所. 主としてニューラルネットワークの数理的側面およびEnsemble Learningの研究に興味を持つ. 1996年度人工知能学会論文賞受賞. 博士 (工学).