

ASVM における例外の影響度を定めるパラメータについて

林 功朗†

† 東京工業大学大学院情報理工学研究科 〒152-8552 東京都目黒区大岡山 2-12-1

E-mail: †hayashi8@is.titech.ac.jp

あらまし サポートベクトル・マシン (SVM) とは、与えられた例集合から、2つのクラスを識別する関数を定義する方法の一つである。与えられた例集合から SVM を構成するのは、一般には二次計画問題で多項式時間計算可能であるが、その解法では大量なデータの処理には向いていない。そのため、いくつかの近似アルゴリズムが提案されているが、なかでも、Mangasarian と Musicant が提案した ASVM という近似アルゴリズムは、計算コストが少ないことで注目を集めている。しかし、SVM とは解く問題が異なるため、効果的な識別のためのパラメータ（具体的にはエラーの影響度を定めるパラメータ）の設定法が不明確である。本稿では、そのパラメータを決める方法を提案し、実験でその妥当性を検証する。

キーワード サポートベクトル・マシン, ASVM, エラー

On the Influence Parameter of Outliers in ASVM

Isao HAYASHI†

† Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Ookayama 2-12-1, Meguro-ku, Tokyo, 152-8552 Japan

E-mail: †hayashi8@is.titech.ac.jp

Abstract A Support Vector Machine (SVM) is one of the methods of defining a binary classification function from a given set of examples. The problem of constructing a SVM from a given example set is in general formalized as a quadratic programming problem. Though there is a polynomial-time algorithm for solving such a problem, it is not efficient enough for dealing with a large example set; thus, several approximation algorithms have been proposed. ASVM is a variation of SVM proposed by Mangasarian and Musicant, who have shown a very efficient algorithm constructing for ASVM. On the other hand, it has been unclear on the way to determine some important parameter (i.e., a parameter defining an influence of errors), and in fact, ASVM does not work if the parameter is not set appropriately. In this paper, through some computer experiments, we investigate this point and propose one reasonable way to determine the parameter.

Key words Support Vector Machines, ASVM, influence of errors

1. はじめに

本稿では、Mangasarian と Musicant によって提案された、サポートベクトル・マシン (以下、SVM とする) の近似アルゴリズムである ASVM [1], [6] の解析を行う。最初に、ASVM の問題の入力である例集合から出力される解の様子を実験的に解析を行った。次に ASVM の問題点を指摘し、誤り例に対する重みパラメータの設定を従

来よりも適切な値に変更し、ASVM を容易に利用できる方法として提案した。

SVM の問題

SVM [3] は、複数の特徴量をまとめた特徴ベクトル (サンプル) $x = (x_1, \dots, x_n)' \in R^n$ と、ラベル $y \in \{-1, 1\}$ の組の集合 $\{(x_1, y_1), \dots, (x_m, y_m)\}$ を入力として受け取り、2つのクラスを識別する識別器を構成する方法であり、その識別器は $y = \text{sign}(w'x - \gamma)$ というかたちを取

る関数で表される。ここで w は重みベクトル、 γ はしきい値と呼ばれる。SVM では「マージン」を最大化することにより、汎化能力を向上させることができると言われている。マージンとは、超平面 $w'x - \gamma$ から各サンプルまでの距離の最小値を意味する。マージンが最大化された超平面と、マージンが最大化されていない超平面の様子を、図 1 に示す。実線は超平面を表し、破線から実線までの距離はマージンを表す。

SVM の問題は、入力するサンプルの集合 $\{(x_1, y_1), \dots, (x_m, y_m)\}$ が分離可能、すなわちある識別関数によってサンプルが完全に識別されるとすると、以下のように定式化ができる。

[定義 1] (SVM の問題 1)

$$(P1) \min_{w, \alpha, \beta} \frac{1}{2} \|w\|^2 - (\alpha - \beta)$$

$$\text{s.t. } w'x_i - \alpha \geq 0 \text{ if } y_i = 1$$

$$w'x_i - \beta \leq 0 \text{ if } y_i = -1$$

しかし一般的に、サンプルの集合にはいくつかの誤り例 (以下、エラーサンプルとする) が含まれている。エラーサンプルが存在することを考慮した SVM の問題は、以下のように定義される。

[定義 2] (SVM の問題 2)

$$(P2) \min_{w, \alpha, \beta, \xi_1, \dots, \xi_m} \frac{1}{2} \|w\|^2 - (\alpha - \beta) + \nu \sum_{i=1}^m \xi_i$$

$$\text{s.t. } w'x_i - \alpha + \xi_i \geq 0 \text{ if } y_i = 1$$

$$w'x_i - \beta - \xi_i \leq 0 \text{ if } y_i = -1$$

$$\xi_1, \dots, \xi_m \geq 0$$

ここで ν は、出力された識別関数ではエラーサンプルに対するペナルティの重みを表す定数である。また、問題 (P2) の双対問題 (D2) は、以下のように表される。

$$(D2) \min_{u_1, \dots, v_1, \dots} \frac{1}{2} \left\| \sum_{i: y_i=1} u_i x_i - \sum_{i: y_i=-1} v_i x_i \right\|^2$$

$$\text{s.t. } \sum_{i: y_i=1} u_i = \sum_{i: y_i=-1} v_i = 1$$

$$0 \leq u_1, \dots, v_1, \dots \leq \nu$$

またここで、別の主問題を考えた SVM の問題 (P3) を紹介する。

[定義 3] (SVM の問題 3)

$$(P3) \min_{w, \gamma, \xi_1, \dots, \xi_m} \frac{1}{2} \|w\|^2 + \nu \sum_{i=1}^m \xi_i$$

$$\text{s.t. } y_i(w'x_i - \gamma) + \xi_i - 1 \geq 0, \xi_i \geq 0$$

(P3) の双対問題 (D3) は、以下のように与えられる。

$$(D3) \min_{u_1, \dots, u_m} \frac{1}{2} \left\| \sum_{i=1}^m u_i y_i x_i \right\|^2 + \sum_{i=1}^m u_i$$

$$\text{s.t. } \sum_{i=1}^m y_i u_i = 0$$

$$0 \leq u_1, \dots, u_m \leq \nu$$

問題 (P2), (P3) は互いに等価な問題であり、一方の問

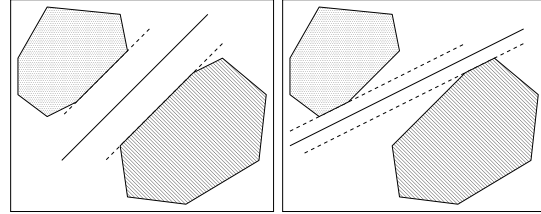


図 1 マージンが最大の超平面 (左図) と、マージンが最大ではない超平面 (右図)

題の最適解より、もう一方の最適解が容易に計算可能であることが知られている。SVM の問題を実際に解くのは、ほぼ不可能といえる。なぜなら、SVM の問題は二次計画問題であり、一般的にその規模は大きく、現実的な計算時間で最適解を出力するのは困難なためである。そのため、SVM の問題を解くいくつかの近似アルゴリズムが提案されている。なかでも有名な近似アルゴリズムとして、Joachims によって提案された SVM^{light} [5] が挙げられる。SVM^{light} は、小規模な二次計画問題を複数回解くことによって双対問題の近似解を計算し、計算時間の向上をはかったものである。しかし依然として、入力するサンプル数が多く、特徴ベクトルの次元数が多い場合には、現実的な計算時間では解を出力できないことも事実である。また、もうひとつの問題点として挙げられるのが、エラーサンプルの重みの適切な設定法が確立されていないということである。また、Dai, Tanaka, Watanabe らによって、問題 (P2) におけるエラーサンプルの重みに関する研究が行われた [4]。いずれの場合も、分離不可能なサンプルの集合に対する問題のうち、(P2) に注目した。そしてその問題をエラーサンプルの重みに応じた変換方法で (P1) に帰着させる方法を提案した。その方法とは、composed example という新しいサンプル構成する方法である。定数 k を用いて $\nu = 1/k$ とすると、composed example は、以下のように与えられる。

$$z_I = \frac{x_{i_1} + \dots + x_{i_k}}{k}$$

このようにして、composed example を構成することにより、問題 (P2) は問題 (P1) に帰着することができるこ

とを証明した。そして、与えられた分離不可能なサンプルの集合 $\{(x_1, y_1), \dots, (x_m, y_m)\}$ を、分離可能な composed example の集合に構成し直し問題を解き、 $w = 0$, $\alpha = \beta = 0$ を出力しない最小の k を使用することを提案した。また、Balcázar, Dai, Tanaka, Watanabeらは、この問題を、ランダムサンプリングによってサンプルの集合を選択し、選ばれた小規模なサンプルの集合による問題を複数回解くことによって近似解を求めることで計算時間の向上をはかった [2]。

2. 近似アルゴリズム ASVM

本稿では、Mangasarian と Musicant によって提案された、ASVM [1], [6] という SVM の近似アルゴリズムに注目した。その理由は、SVM^{light} では現実的な時間では計算不可能だった大規模な問題に対しても、現実的な時間で解を出力することができるという発表があったためである。Mangasarian と Musicant は SVM の双対問題に関して、以下の点が計算時間を悪化させる原因であると考え、問題の書き換えを行った。

- SVM の双対問題が、正定値行列の凸計画問題ではないので、解が唯一ではない可能性が有る。
- 等式条件や、各変数に対する上限が設けられていることで、大規模の問題になればなるほど、計算時間に悪影響を及ぼす。
- SVM の双対問題を解いた場合、 γ を求めるために、 γ のみに対する最適化問題を解く必要がある。準備として、各記号の定義を行う。ベクトル、行列の転置を、記号 $'$ を用いて表す。また、 $x \in R^n$ に対して、 x_+ を以下のように定義する。

$$x_{+i} = \begin{cases} x_i & \text{if } x_i > 0 \\ 0 & \text{if } x_i \leq 0 \end{cases}$$

$A \in R^{m \times n}$ に対して、 A_i を A の i 行目の成分をとりだした m 次ベクトル、全成分が 1 のベクトルを e 、全成分が 0 のベクトルを 0 で表し、単位行列を I で表す。また、 $u \in R^m$, $Q \in R^{m \times m}$ と $B \subset \{1, 2, \dots, m\}$ に対して、 u_B を $u_{i \in B} \in R^{|B|}$ とし、 Q_B を $Q_{i \in B} \in R^{|B| \times m}$ とし、 Q_{BB} を Q のうち、行 $i \in B$ 、列 $j \in B$ をとりだした部分行列とする。また、 $:=$ は定義を表す。最後に、入力されるサンプルの集合を、今後行列 $A \in R^{m \times n}$, $D \in R^{m \times m}$ を用いて表記し、それぞれを $A_i = x'_i$, $D_{ii} = y_i$, $D_{ij} = 0$ if $i \neq j$ とする。

上記にある計算時間を悪化させる原因を克服するために、ASVM の問題 (P4) と、その双対問題 (D4) は、以下のように定義された。

[定義 4] (ASVM の主問題)

$$(P4) \quad \min_{(w, \gamma, \xi) \in R^{n+1+m}} \frac{1}{2}(w'w + \gamma^2) + \nu \frac{\xi' \xi}{2}$$

$$\text{s.t. } D(Aw - e\gamma) + \xi \geq e$$

$$(D4) \quad \min_{0 \leq u \in R^m} \frac{1}{2} u' \left\{ \frac{I}{\nu} + D(AA' + ee')D \right\} u - e'u$$

行列 $I/\nu + D(AA' + ee')D$ は、明らかに正定値行列であり、(D4) は凸計画問題となる。また、等式条件や、 u に対して上限が設けられていない。また、(P4), (D4) の最適解の間には、以下に示す条件 (KKT 条件) が成り立っている。

[KKT 条件]

$$D(Aw - e\gamma) + \xi - e \geq 0$$

$$u' \{D(Aw - e\gamma) + \xi - e\} = 0$$

$$w - A'Du = 0$$

$$\gamma + e'Du = 0$$

$$\nu \xi - u = 0$$

$$u \geq 0$$

以上より、双対問題の最適解から直接 w , γ が計算可能である。このようにして、Mangasarian と Musicant は、計算時間を悪化させる原因と考えていたものを克服した。アルゴリズムの概要を紹介する。アルゴリズムの概要は以下のとおりである。表記を簡単にするため、行列 $I/\nu + D(AA' + ee')D$ を Q としている。

アルゴリズム ASVM の概要

- (0) $u^0 := (Q^{-1}e)_+$ とする。 $i = 0, \dots$ に対して、 u^i から u^{i+1} を以下のように計算する。
- (1) $B^i := \{j | u_j^i > 0\}$, $N^i := \{j | u_j^i = 0\}$
- (2) $u_{B^i}^{i+1} := (Q_{B^i B^i}^{-1} e_{B^i})_+$, $u_{N^i}^{i+1} := 0$ とする。
 $u^{i+1} \geq 0$, $Qu^{i+1} - e \geq 0$, $u^{i+1}(Qu^{i+1} - e) = 0$ を満たすならば停止し、 u^{i+1} を解として出力する。
- (3) $i := i + 1$ として、(1) へ進む。

アルゴリズムの停止条件 $u^{i+1} \geq 0$, $Qu^{i+1} - e \geq 0$, $u^{i+1}(Qu^{i+1} - e) = 0$ は、上記の KKT 条件 $D(Aw - e\gamma) + \xi - e \geq 0$, $u' \{D(Aw - e\gamma) + \xi - e\} = 0$, $u \geq 0$ を満たすことを意味する。さらに残りの KKT 条件より、 w , γ を計算する。以上が ASVM のアルゴリズムの概要で

ある。Mangasarian と Musicant は、ASVM の実行時間は、少ないサンプル数の場合、SVM^{light} と同等またはそれ以上であり、SVM^{light} では現実的な時間で計算ができなかったサンプル数が数百万規模の問題に対しても、現実的な時間で解を出力することができることを実験により示した。また、識別成功確率に関しても、SVM^{light} と同等の結果が得られることを実験により示した。

3. ASVM の問題点

本稿でも、サンプルの集合を生成するアルゴリズムを構成し、ASVM の出力する解について実験を行った。本実験では、サンプルの集合を以下のような方法で構成した。

サンプルの集合の構成アルゴリズム

- (0) 次元数 n , サンプル数 m , しきい値 γ , ランダムシード s , サンプルの集合のエラー率 e , 正例のうち、正しく識別されることを想定したサンプル数に占める割合 p_+ を入力として受け取る。
- (1) マージン ϕ を s に応じて生成する。
- (2) 重みベクトル w を s に応じて生成し, $\frac{k}{\|w\|} = \phi$ を満たす k を計算し, $w := kw$ とする。
- (3) $p_+ = 0$ ならば, 正例, 負例の割合は超平面 $w'x = \gamma$ によって分割される, 正例の存在する部分, 負例の存在する部分の体積に応じて, 正例, 負例の割合が決定されるとし, (4) へ進む。
 $p_+ \neq 0$ の場合は, (5) へ進む。
- (4) 正しく識別されることを想定したサンプルを生成する。各 x_i を s に応じて生成し, $w'x \geq \gamma + 1$ ならば, そのサンプルを正例として利用し, $w'x \leq \gamma - 1$ の場合は, 負例として利用する。それ以外の場合は, もう一度サンプルを生成し直す。 $m(1 - e)$ 回, この操作をくり返し, 終了後, (7) へ進む。
- (5) 正しく識別されることを想定した正例を生成する。各 x_i を s に応じて生成し, $w'x \geq \gamma + 1$ ならば, そのサンプルを利用する。それ以外の場合は, もう一度サンプルを生成し直す。 $mp_+(1 - e)$ 回, この操作をくり返す。

- (6) 正しく識別されることを想定した負例を生成する。各 x_i を s に応じて生成し, $w'x \leq \gamma - 1$ ならば, そのサンプルを利用する。それ以外の場合は, もう一度サンプルを生成し直す。
 $m(1 - p_+)(1 - e)$ 回, この操作をくり返し, 終了後, (7) へ進む。
- (7) 正しく識別されないことを想定したサンプルを生成する。各 x_i を s に応じて生成し, $w'x \leq \gamma$ ならば, そのサンプルを正例として利用する。それ以外の場合は, 負例として利用する。 me 回, この操作をくり返す。終了後, 生成された m 個のサンプルを出力して停止する。

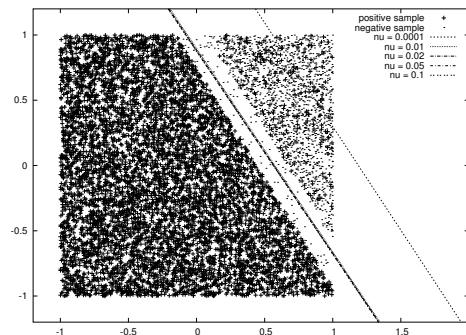


図2 エラーサンプルに対する重み ν を変化させた場合の超平面の様子

ASVM では, ν を手動で設定することも可能であるが, 設定が無い場合, 自動的に $\nu = 1/m$ (m はサンプル数) として, 計算を行う。その $\nu = 1/m$ という設定では, サンプルの集合の識別が不適切な場合があることがわかった。同時に, ν をある程度の大きさに設定すると, サンプルの集合を適切に識別できることが分かった。サンプル数 $m = 10000$ の実験結果の一部 (図2) を紹介する。サンプル数 $m = 10000$ の場合, ASVM によって自動的に設定される $\nu = 0.0001$ という値を用いた場合, 超平面が大きく負例よりに存在し, 正しい識別ができていない。しかし $\nu = 0.01, 0.02, 0.05, 0.1$ のような, 0.0001 よりも大きな値を ν として定めた場合, 識別成功確率が高まることがわかった。

本稿の目的は, ASVM を利用するにあたり, 適切なエラーサンプルに対する重み ν の解析である。本稿では, 超平面が近似的に安定するという基準を用いて, 以下の予想と仮定をもとに実験的解析を行った。

4. ν の設定に関する実験的解析

ASVM ではエラーサンプルに対する重み ν の設定によっては、サンプルの集合を分離できないことがあることがわかった。また、エラーサンプルに対する重み ν がある程度大きな場合、サンプルの集合を分離することができ、またその分離平面は、ほぼ等しいことが視覚的に確認できた。そこで本稿では、以下のような予想を立てた。
 [予想 1] ASVM によって、構成される超平面が近似的に同一となるような区間 $\nu \in [\nu_{\text{begin}}, \nu_{\text{end}}]$ が存在する。
 [予想 2] ASVM において、近似的に安定した超平面の識別成功確率は高い。

本稿では、ある重み ν によって出力される識別関数 (超平面) と、その近傍 ν' によって出力される超平面が、近似的に等しくなるようなエラーサンプルの重み ν の設定方法について実験的に解析を行い、そのときの識別成功確率を改めて確認することによって、識別成功確率が高くなるようなエラーサンプルに対する重み ν の設定方法を提案することを実験の目的とする。実験を行うにあたり、本稿では安定度を以下のように定義した。

[定義 5] (安定度) エラーサンプルに対する重み ν によって定まる超平面 $w'x = \gamma$ と、 ν の近傍内の任意の ν' によって定まる超平面 $(w')'x = \gamma$ に対して、安定度 δ を、以下のように定義する。

$$\delta = \frac{\min \left\{ \frac{w_{a1}}{w_{b1}}, \dots, \frac{w_{an}}{w_{bn}}, \frac{\gamma_a}{\gamma_b} \right\}}{\max \left\{ \frac{w_{a1}}{w_{b1}}, \dots, \frac{w_{an}}{w_{bn}}, \frac{\gamma_a}{\gamma_b} \right\}}$$

$w_i = 0, w'_i = 0, \gamma = 0, \gamma' = 0$ のいずれかが満たされる場合は安定度は定義されない。ある安定度 δ^* 以上の安定度を達成するエラーサンプルの重み ν を安定度 δ^* の安定点と呼ぶ。

本実験では、サンプルの集合を次元数 n 、しきい値 γ 、エラー率 e 、正例の割合 p_+ 、という特徴を用いて分類し、それぞれの場合における適切なエラーサンプルの重み ν の設定の変化をサンプル数ごとに実験を行うことにより調べ、どの特徴がエラーサンプルに対する重み設定に強く影響を及ぼすかを調べる。また本実験では、さまざまな安定度における安定点を実験により求め、さまざまなサンプルの集合に対して、高い安定度を達成することのできるエラーサンプルに対する重み ν の設定方法を提案する。次に、SVM が機能するどのようなサンプルの集合を受け取った場合にも対応するような重み設定の方法を提案する。なお、本稿では分離不可能なサンプルの集合のみを取扱う。

実験方法を紹介する。本稿では、以下のような実験を行い、安定度 $\delta^* \in \{0.999, 0.9999, 0.99999\}$ における安

定点を求める。本実験では、定義 5 で言う近傍を $1/m$ と定め、さらに実験を $1200/m$ で終了させることとした。次節より、実験結果を紹介する。

安定度 δ における安定度を求める実験方法

- (1) 安定度 δ^* を設定する。
- (2) $\nu = 1/m, \nu' = 2/m$ とする。
- (3) 安定度 δ が $\delta > \delta^*$ を満たすか調べる。
- (4) $\nu = \nu + 1/m, \nu' = \nu' + 1/m$ として、 $\nu' < \frac{1200}{m}$ までくり返す。

4.1 次元数 n に応じた、重み ν の変化

本節では、サンプルの集合を次元数 n という観点から分類し、安定度 δ における安定点の変化を調べる。実験対象とするデータは以下のとおりである。

- 次元数 n : 2, 5, 10, 15, 20, 25, 30
- サンプル数 m : 10000, 20000, ..., 200000
- しきい値 γ : 0
- ランダムシード s : 1000
- エラー率 e : 0.05
- 正例の割合 p_+ : 体積に比例

以上の実験データに対して、

安定度 $\delta^* \in \{0.999, 0.9999, 0.99999\}$ における安定点を計算し、その違いの比較を行った。

実験結果の一部を紹介する。本実験によって、次元数 n に応じた、重み ν の変化に対して、以下のような結果 (図 3) が得られた。グラフは、横軸はサンプル数、縦軸は

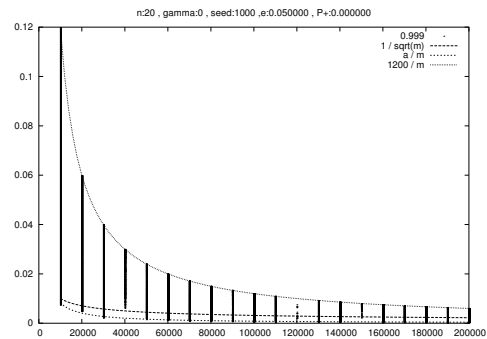


図 3 $n = 20$ の実験結果の一部

ν の大きさを表しており、各サンプル数に対して安定度 δ^* 以上の点をプロットした。また、曲線は $\nu = 1200/m$ 、 $\nu = 1/m$ の定数倍、 $\nu = 1/\sqrt{m}$ を表している。これらの結果より、超平面が近似的に安定するような重み ν は、ある程度長い間継続することが確認できた。また、次元数が高くなるにつれ、 $\delta^* = 0.99999$ 以上の安定度を達成

する最小の重み ν は、大きくなることがわかった。これは、安定度を計算する際の比較対象が、次元数に応じて増加することを考えると、当然の結果と考えることができる。しかし、 $\delta^* = 0.999, 0.9999$ の場合には、この傾向は顕著ではなかった。

次に、識別成功確率について議論をすすめる。本実験では、最小の安定点に対する識別成功確率を計算した。実験結果の一部を紹介する。

m	識別成功確率		
	$n = 10$	$n = 20$	$n = 30$
50000	0.9502	0.9502	0.9504
100000	0.9502	0.9502	0.9502
150000	0.9503	0.9502	0.9502
200000	0.9502	0.9501	0.9502

上の表は、安定度 $\delta^* = 0.9999$ における、最小の安定点に対する識別成功確率である。識別成功確率は、想定していたエラー率 e と比較すると、問題無くサンプルの集合を分離していると考えられる。

4.2 しきい値 γ に応じた、重み ν の変化

本節では、サンプルの集合をしきい値 γ 、という観点から分類し、安定度 δ^* における安定点の変化を調べる。実験対象とするデータは以下のとおりである。

- 次元数 $n : 10$
- サンプル数 $m : 10000, 20000, \dots, 200000$
- しきい値 $\gamma : 0, 1, 2, 3, 4, 5$
- ランダムシード $s : 1000$
- エラー率 $e : 0.05$
- 正例の割合 $p_+ : 体積に比例$

以上の実験データに対して、安定度 $\delta^* \in \{0.999, 0.9999, 0.99999\}$ における安定点を計算し、その違いの比較を行った。

実験結果の一部を紹介する。本実験によって、しきい値 γ に応じた、重み ν の変化に対して、以下のような結果 (図 4) が得られた。これらの結果より、超平面が近似

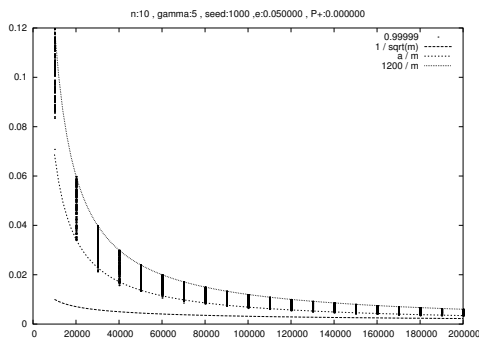


図 4 $\gamma = 5$ の実験結果の一部

的に安定するような重み ν は、ある程度長い間継続することが確認できた。また、しきい値が大きくなるにつれ、 $\delta^* = 0.99999$ 以上の安定度を達成する最小の重み ν は、大きくなることがわかった。しかし、 $\delta^* = 0.999, 0.9999$ の場合には、この傾向は顕著ではなかった。

次に、識別成功確率について議論をすすめる。本実験では、最小の安定点に対する識別成功確率を計算した。実験結果の一部を紹介する。

m	識別成功確率		
	$\gamma = 0$	$\gamma = 2$	$\gamma = 5$
50000	0.9502	0.9502	0.9512
100000	0.9502	0.9501	0.9512
150000	0.9503	0.9502	0.9512
200000	0.9502	0.9501	0.9512

上の表は、安定度 $\delta^* = 0.9999$ における、最小の安定点に対する識別成功確率である。識別成功確率は、想定していたエラー率 e と比較すると、問題無くサンプルの集合を分離していると考えられる。

4.3 エラー率 e に応じた、重み ν の変化

本節では、サンプルの集合をエラー率 e 、という観点から分類し、安定度 δ^* における安定点の変化を調べる。実験対象とするデータは以下のとおりである。

- 次元数 $n : 10$
- サンプル数 $m : 10000, 20000, \dots, 200000$
- しきい値 $\gamma : 0$
- ランダムシード $s : 1000$
- エラー率 $e : 0.01, 0.03, 0.05, 0.07, 0.10, 0.15$
- 正例の割合 $p_+ : 体積に比例$

以上の実験データに対して、安定度 $\delta^* \in \{0.999, 0.9999, 0.99999\}$ における安定点を計算し、その違いの比較を行った。

実験結果の一部を紹介する。本実験によって、エラー率 e に応じた、重み ν の変化に対して、以下のような結果 (図 5) が得られた。これらの結果より、超平面が近似

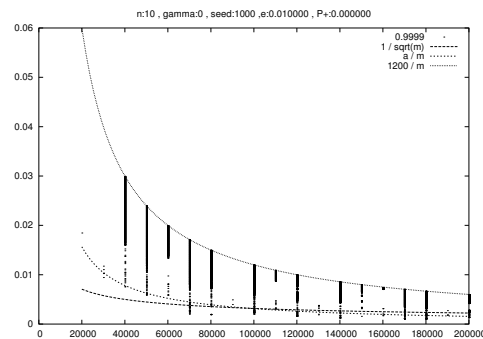


図 5 $e = 0.01$ の実験結果の一部

的に安定するような重み ν は、ある程度長い間継続することが確認できた。また、エラー率が小さくなるにつれ、 $\delta^* = 0.99999$ 以上の安定度を達成する最小の重み ν は、大きくなることがわかった。特に $e = 0.01$, $\delta^* = 0.99999$ では、実験中で超平面が安定する重み設定はほとんど発見できなかった。しかし、きわめて少ないデータであるが、 ν の設定を、本実験よりも大きな値 ($m = 10000$ に対して $0.1, 1$ など) とした場合には、その近傍で超平面が近似的に安定した。このことより、エラー率が低い場合には、 $\nu > 1200/m$ の部分に超平面が安定するような値 (区間) が存在すると予想できる。

次に、識別成功確率について議論をすすめる。本実験では、最小の安定点に対する識別成功確率を計算した。実験結果の一部を紹介する。

m	識別成功確率		
	$e = 0.01$	$e = 0.05$	$e = 0.10$
50000	0.9900	0.9502	0.9007
100000	0.9900	0.9502	0.9009
150000	0.9900	0.9503	0.9005
200000	0.9900	0.9502	0.9003

上の表は、安定度 $\delta^* = 0.9999$ における、最小の安定点に対する識別成功確率である。識別成功確率は、想定していたエラー率 e と比較すると、問題無くサンプルの集合を分離していると考えられる。

4.4 正例の割合 p_+ に応じた、重み ν の変化

本節では、サンプルの集合を正例の割合 p_+ 、という観点から分類し、実験対象とするデータは以下のとおりである。

- 次元数 $n : 10$
- サンプル数 $m : 10000, 20000, \dots, 200000$
- しきい値 $\gamma : 0$
- ランダムシード $s : 1000$
- エラー率 $e : 0.05$
- 正例の割合 $p_+ : 体積に比例, 0.30, 0.40, 0.50, 0.60$

以上の実験データに対して、安定度 $\delta^* \in \{0.999, 0.9999, 0.99999\}$ における安定点を計算し、その違いの比較を行った。

実験結果の一部を紹介する。本実験によって、正例の割合 p_+ に応じた、重み ν の変化に対して、以下のような結果 (図 6) が得られた。これらの結果より、超平面が近似的に安定するような重み ν は、ある程度長い間継続することが確認できた。また、サンプルの集合を構成する際に定めた正例が存在できる範囲、負例が存在できる範囲の体積の割合と、正例、負例の数の割合の間に開きがある、すなわち正例、負例に「密度差」がある場合、超平面が近似的に安定する重み ν の最小値は大きくなる傾

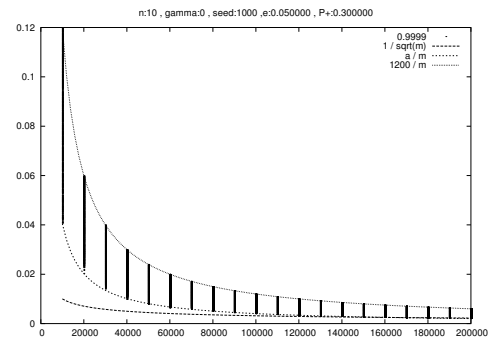


図 6 $p_+ = 0.30$ の実験結果の一部

向があった。それは、サンプル数が少ない場合に、より顕著である。さらに、次元数、しきい値、エラー率ではおおきな違いが見られなかった $\delta^* = 0.9999$ においても、その差が確認できた。

次に、識別成功確率について議論をすすめる。本実験では、最小の安定点に対する識別成功確率を計算した。実験結果の一部を紹介する。

m	識別成功確率		
	$p_+ : 比例$	$p_+ = 0.30$	$p_+ = 0.40$
50000	0.9502	0.9512	0.9515
100000	0.9502	0.9523	0.9515
150000	0.9503	0.9528	0.9515
200000	0.9502	0.9524	0.9515

上の表は、安定度 $\delta^* = 0.9999$ における、最小の安定点に対する識別成功確率である。識別成功確率は、想定していた正例の割合 p_+ と比較すると、問題無くサンプルの集合を分離していると考えられる。

4.5 実験結果

予想 1, 2 に対する検証結果を述べる。本実験により、超平面が近似的に安定する区間の存在を確かめることができた。また同時に、近似的に安定した超平面における識別成功確率が高いことと、近似的に安定していない超平面では、高い識別成功確率が保障されないことが、 $\nu = 1/m$ を例にとることでわかった。以下に、識別成功確率の実験結果の一部を紹介する。

m	識別成功確率		識別成功確率	
	$\gamma = 0$	$\gamma = 5$	$P_+ = 0.30$	$P_+ = 0.50$
50000	0.9538	0.8095	0.8134	0.9537
100000	0.9536	0.8092	0.8786	0.8899
150000	0.9539	0.8091	0.8785	0.8879
200000	0.9538	0.8098	0.8797	0.8864

この実験より、超平面が原点を通らない場合、正例と負例の密度の差が大きな場合には、識別成功確率が、想定

しているエラー率を考慮した場合、大きく下回ってしまうことが確認できた。以上より、以下を実験結果とする。
 [実験結果 1] ASVM によって、構成される超平面が近似的に同一となるような区間が存在した。

[実験結果 2] 近似的に安定した超平面では、高い識別成功確率が保障される。また、近似的に安定していない超平面では、高い識別成功確率が保障されない。

次に本稿では、最小の安定点 ν の近似を行う。エラーサンプルに対する重みが余りに大きい場合、極端にエラーサンプルに注目した学習を行ってしまい、これは良いことではないと考えることが当然であろう。本稿では、 $\nu = a/m$ の定数 a での近似を試みた。するとその定数は、サンプルの集合によりおおきくばらつきがあることがわかった。実験結果の一部を以下に紹介する。表は、各 p_+ に対する、安定度 $\delta \in \{0.999, 0.9999, 0.99999\}$ における定数 a を表している。

p_+	定数 a		
	0.999	0.9999	0.99999
比例	74.1643	179.347	621.448
0.30	120.321	404.697	1037.30
0.40	135.199	360.725	894.250
0.50	155.643	264.395	731.873
0.60	116.718	370.453	838.488

また、 $\nu = 1/\sqrt{m}$ という設定は、高い安定度を達成することがわかった。安定度の計算実験の結果の一部を以下に紹介する。表は、各サンプル数 m において、重み設定を $1/m, 1/4\sqrt{m}, 1/2\sqrt{m}, 1/\sqrt{m}$ としたときの安定度を表している。 $(n = 30, \gamma = 0, e = 0.05, p_+ : \text{比例})$

m	安定度			
	$1/m$	$1/4\sqrt{m}$	$1/2\sqrt{m}$	$1/\sqrt{m}$
50000	0.788058	0.996345	0.998656	0.999346
100000	0.735404	0.985769	0.992123	0.999072
150000	0.701324	0.967361	0.986741	0.996095
200000	0.579656	0.997900	0.999096	0.999773

$\nu = 1/4\sqrt{m}$ では、高い安定度が期待できず、また、 $\nu = 1/2\sqrt{m}$ でも、 $\nu = 1/\sqrt{m}$ の場合に比べて、高い安定度が期待できない。以上の結果より本稿では、 $\nu = 1/\sqrt{m}$ という設定は、ASVM をサンプルの集合を与えるだけで利用する (容易に利用する) ための有効な設定方法のひとつであると考え、以下を提案した。

[提案]

ASVM を容易に利用して、識別成功確率の高い超平面を得るためには、

$\nu = 1/\sqrt{m}$ と設定すればよい。

5. ま と め

本稿では、ASVM が苦手とする問題の条件のひとつを発見した。その条件とは、正例と負例の密度の差が大きい場合とエラー率が小さい場合である。また、サンプルの密度差が少ない場合には、エラーサンプルに対する重み $\nu = 1/\sqrt{m}$ に対する安定度は高く、その値は、ほとんどの場合 $\delta = 0.999 \sim 0.9999$ の安定度を達成することがわかった。さらに、サンプル数 m が大きくなるほど、各サンプルの集合に対する ν_{begin} の値のばらつきが小さくなることもわかった。ASVM が他の SVM の近似アルゴリズムと比較して特にすぐれている点は、今まで現実的な時間で解くことが不可能であったような、サンプル数 m の多い問題を解くときである。入力するサンプル数が、数万規模の場合は、SVM^{light} の計算時間と比べても差が少なく、かつ識別成功確率も高いため、ASVM よりも SVM^{light} を選択したほうがよいと考えられる。したがって、大規模な問題であるほど安定開始点のばらつきが小さくなるという結果は、好意的に受け止めることができるかと判断した。

今後の課題を述べる。本研究では、エラーサンプルに対する適切な重み設定の実験的な解析を行った。しかし、ASVM と SVM の問題の対応関係を明確にし、本研究で提案した重み設定と、SVM で提案されている重み設定との対応関係を調べることにより、理論的に ASVM の重み設定を理論的に解析できる可能性があると考えられる。

文 献

- [1] Active Support Vector Machine Home Page, <http://www.cs.wisc.edu/dmi/asvm/>
- [2] J. Balcázar, Y. Dai, J. Tanaka and O. Watanabe. Provably Fast Training Algorithms for Support Vector Machines Technicl Reports on Mathematical and Computer Sciences: TR-C160. 2002.
- [3] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press. 2000.
- [4] Y. Dai, J. Tanaka, and O. Watanabe. On the Influence of Outliers in the Support Vector Machine Framework Technicl Reports on Mathematical and Computer Sciences: TR-C163. 2002.
- [5] T. Joachims, Making Large-Scale SVM Learning Practical. Advances in Kernel Methods: Support Vector Machines. 1998.
- [6] O. L. Mangasarian and D. R. Musicant. Active Support Vector Machine Classification. Advances in Neural Information Processing Systems (NIPS 2000). 2000.