

アイリスデータを用いた クラス判別手法の比較

東京理科大学工学部第1部経営工学科
沼田研究室

4400083 中島理恵

4400118 分部理恵

発表構成

1. はじめに
 2. 研究目的
 3. 判別分析の基本
 - 3.1 ニューラルネット(NN)
 - 3.2 サポートベクターマシン(SVM)
 4. 実験
 5. 実験結果および考察
 6. まとめ
 7. 今後の課題
- 参考文献

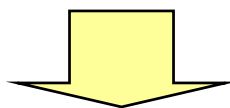
1. はじめに

外見だけでは区別できないものがある

ex) キノコに毒がある？ない？

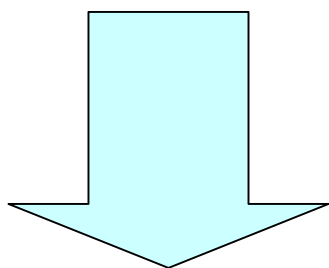


毒の有無がわかっているものの特徴から「学習」
して精度良く予測



クラス判別手法

- ・ベイズ識別規則
- ・重回帰分析
(線形関数, ロジスティック関数)



- ・ニューラルネット (NN)
- ・サポートベクターマシン (SVM)

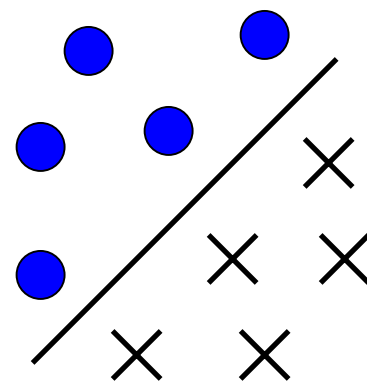


図1 線形分離境界

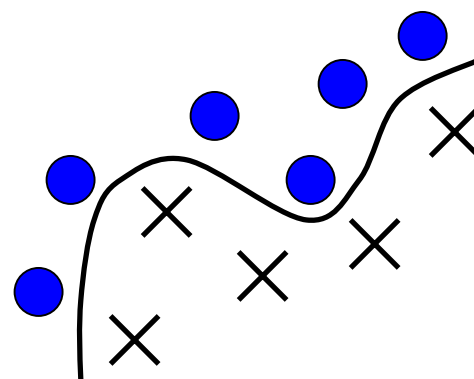


図2 非線形分離境界 4

2. 研究目的

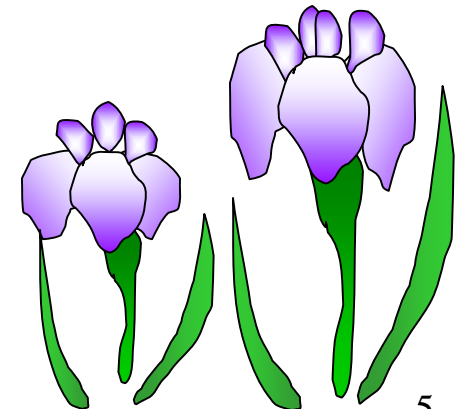
アイリスデータ

NN or SVM

- ・ 判別精度の比較
- ・ 各手法の特徴の理解

クラス(3)	特徴量(4)			
花の種類	花びら幅	花びら長さ	がく片幅	がく片長さ
セトサ	2	14	33	50
バーシクル	13	45	28	57
バージニカ	23	51	31	69

[1]



3. 判別分析の基本

特徴量 $\mathbf{x}_i = (x_{i1} \cdots x_{in})^T$, 帰属クラス d_i ($i = 1, \dots, N$)

学習データ:

$$\begin{bmatrix} x_{11} & \cdots & x_{1n} & d_1 \\ x_{21} & \cdots & x_{2n} & d_2 \\ \vdots & & \vdots & \vdots \\ x_{N1} & \cdots & x_{Nn} & d_N \end{bmatrix}^T \quad (d_i = 1 \text{ or } -1)$$

→ $\begin{cases} d_i = 1 \text{ のとき } \underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 > 0 \\ d_i = -1 \text{ のとき } \underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 < 0 \end{cases}$ となる重み $\underline{\mathbf{w}}, \underline{w}_0$

分離超平面: $\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 = 0$

線形判別素子ニューロン^[2]

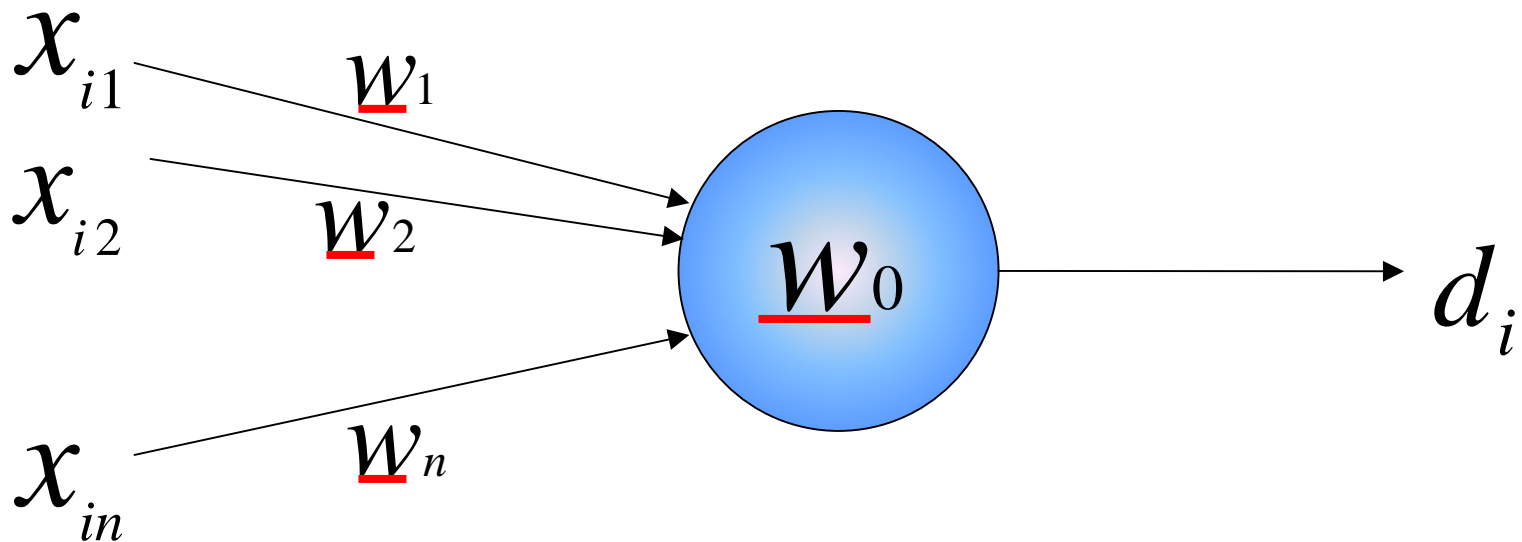


図3 ニューロン

ステップ関数 S の導入

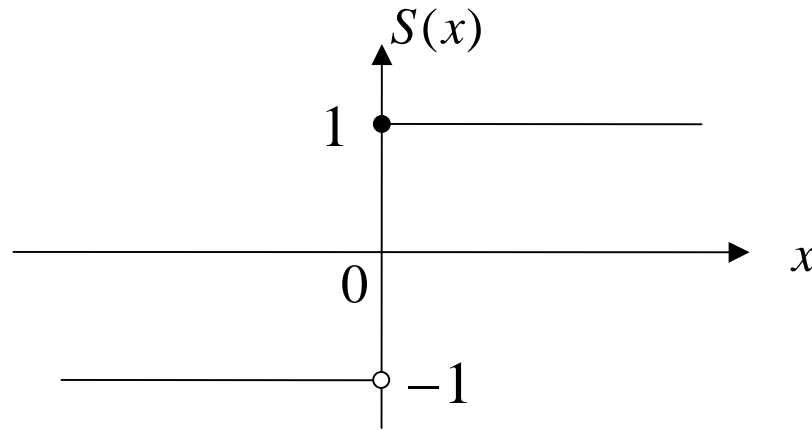


図4 ステップ関数

クラス未知サンプル: $\mathbf{u} = (u_1, \dots, u_n)^T$

これが属するクラスは $S(\underline{\mathbf{w}}^T \mathbf{u} - \underline{w}_0)$ で推定可能

NNの「学習」方法

教師データと出力の誤差  最小

となる重み w, w₀ を求める.

ここで、 $\left\{ \begin{array}{l} \text{教師データ: } d_i = 1 \text{ or } 0 \quad (i = 1, \dots, N) \\ \text{出力} \quad \quad \quad : 0 \text{以上} 1 \text{以下} \end{array} \right.$

シグモイド関数

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

とする.

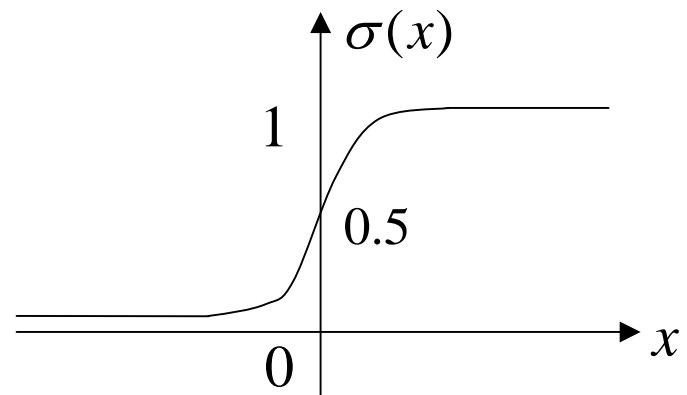


図5 シグモイド関数

SVMの「学習」方法

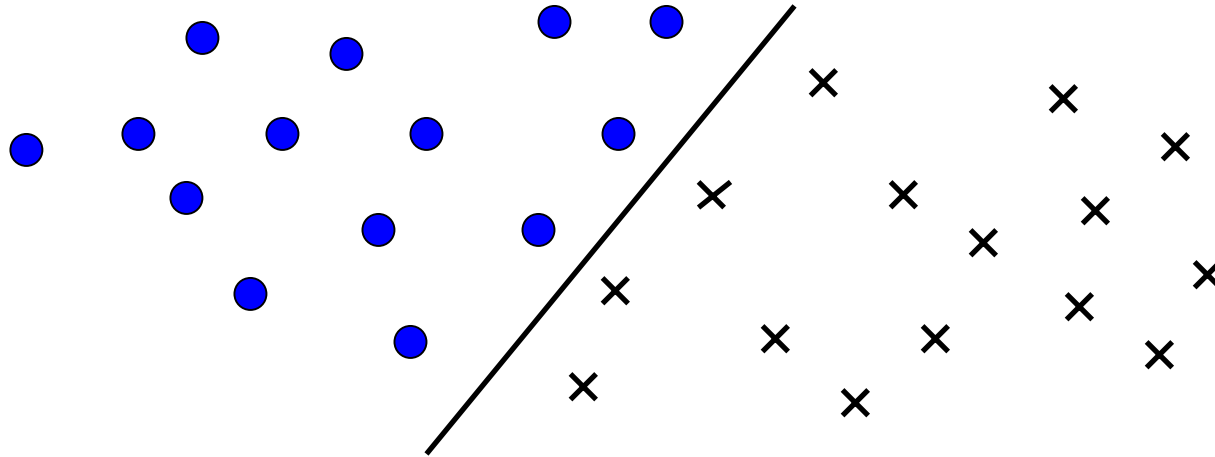
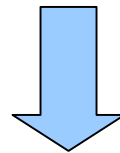


図6 2クラスの学習サンプル

2次計画問題に定式化



最適な重み w , w_0 を求める

3.1 ニューラルネット(NN)

多数のニューロンを組み合わせる . [2]

$$\begin{aligned} 0 &\leq x_1, x_2, x_3, x_4 \leq 1 \\ 0 &\leq y_1, \dots, y_m \leq 1 \\ 0 &\leq z_1, z_2, z_3 \leq 1 \end{aligned}$$

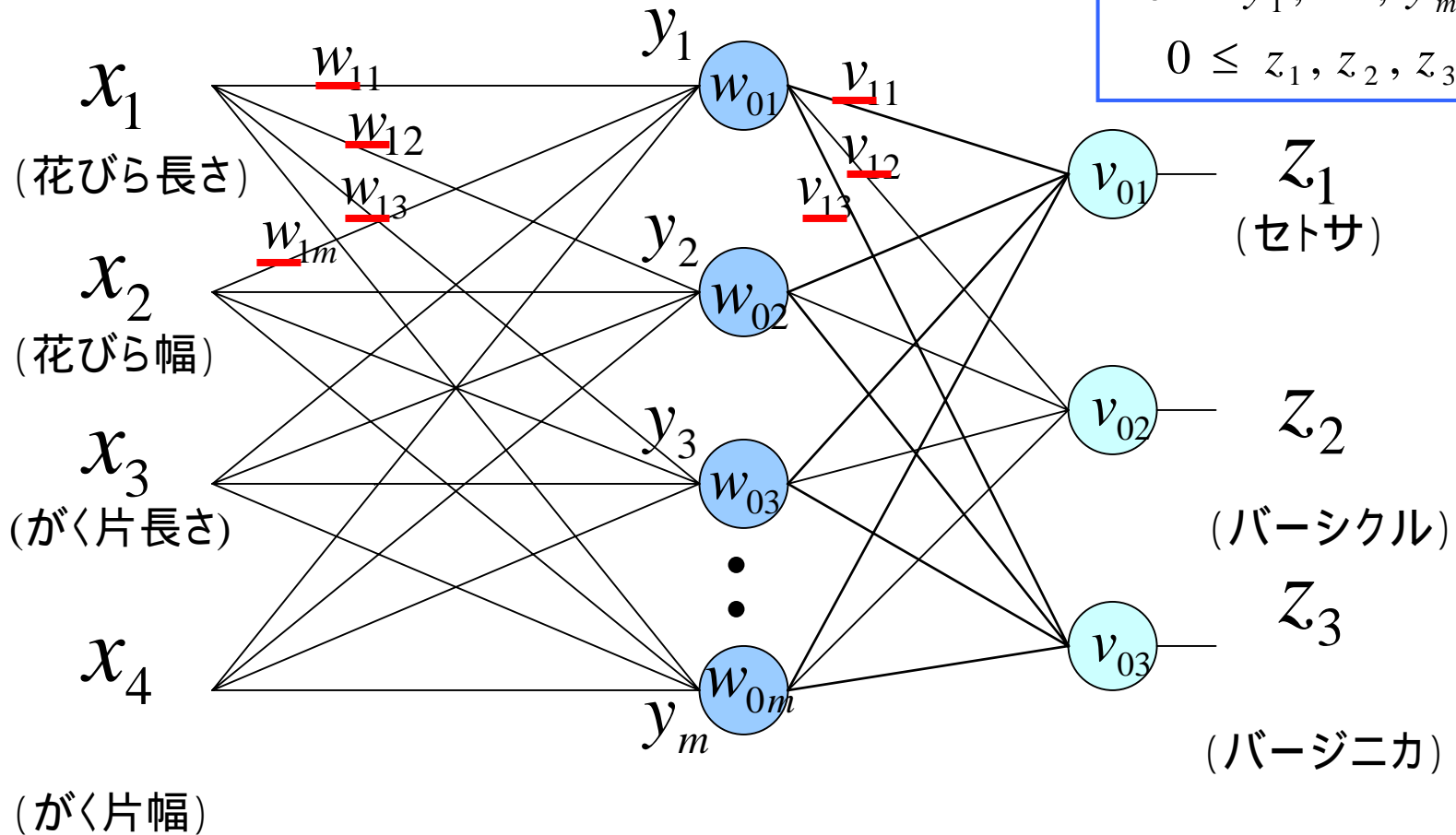


図7 ニューラルネットの構造

第2層での出力:

$$f_j(x) = y_j = \sigma(\underline{w_{1j}}x_1 + \underline{w_{2j}}x_2 + \underline{w_{3j}}x_3 + \underline{w_{4j}}x_4 - \underline{w_{0j}}) \quad (j = 1, \dots, m)$$

第3層(出力)での出力:

$$g_l(y) = z_l = \sigma(\underline{v_{1l}}y_1 + \dots + \underline{v_{ml}}y_m - \underline{v_{0l}}) \quad (l = 1, 2, 3)$$

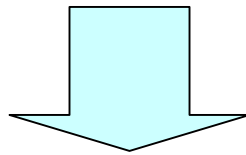
つまりニューラルネットは

ニューロンで表される関数の合成によって作られる関数：
$$\mathbf{h} = \mathbf{g} \circ \mathbf{f}$$

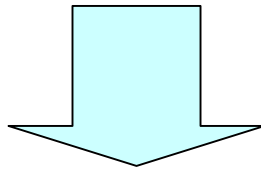
誤差逆伝搬学習法 (Back propagation) [2]

教師データ $\mathbf{d} = (d_1, \dots, d_N)$ と出力 $\mathbf{h} = (z_1, \dots, z_N)$

の差の2乗和(誤差): $E \equiv \frac{1}{2} \sum_{x \in X} \|h(x) - d(x)\|^2$



重み \mathbf{w} , \mathbf{v} に適当な初期値を代入



最急降下法により, 誤差 E を最小(極小)点へ向かわせる.

勾配ベクトルの計算

$$\nabla E(v) = \frac{\partial E}{\partial v_{jk}} \quad (j = 0, \dots, m, \quad k = 1, \dots, l)$$

$$= \sum_x \{z_k - d_k\} z_k (1 - z_k) \cdot y_j$$

$$= \sum_x \rho_{2k} y_j$$

$$\nabla E(w) = \frac{\partial E}{\partial w_{ij}} \quad (i = 0, \dots, n, \quad j = 1, \dots, m)$$

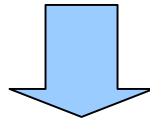
$$= \sum_x \sum_{p=1}^l \{z_p - d_p\} z_p (1 - z_p) v_{jp} \cdot y_j (1 - y_j) \cdot x_i$$

$$= \sum_x \left\{ \sum_{p=1}^l \rho_{2p} v_{jp} \right\} y_j (1 - y_j) \cdot x_i$$

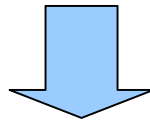
$$= \sum_x \rho_{1j} x_i$$

学習係数 $\alpha (> 0)$ を用いて漸化式をつくる。

$$\left\{ \begin{array}{l} \underline{v}_{jk}^{new} = \underline{v}_{jk}^{old} - \alpha \cdot \sum_x \underline{\rho}_{2k} y_j \quad (j=0, \dots, m, \quad k=1, \dots, l) \\ \underline{w}_{ij}^{new} = \underline{w}_{ij}^{old} - \alpha \cdot \sum_x \underline{\rho}_{1j} x_i \quad (i=0, \dots, n, \quad j=1, \dots, m) \end{array} \right.$$



新たな出力 $\mathbf{h} = (z_1, \dots, z_N)$ を得る。



誤差 E が最小(極小)になるまでくり返す。

3.2 サポートベクターマシン (SVM)

線形分離可能

クラス● : +1, クラス× : -1

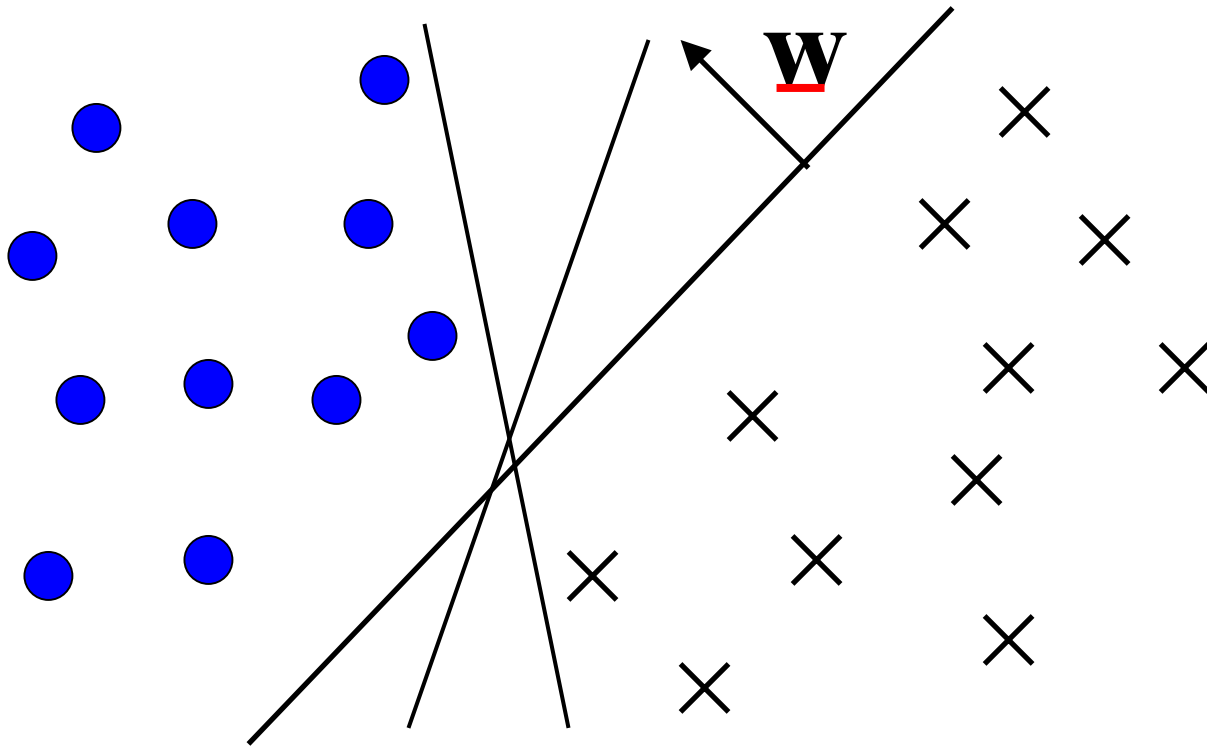


図8 線形分離可能

マージンの大きさ

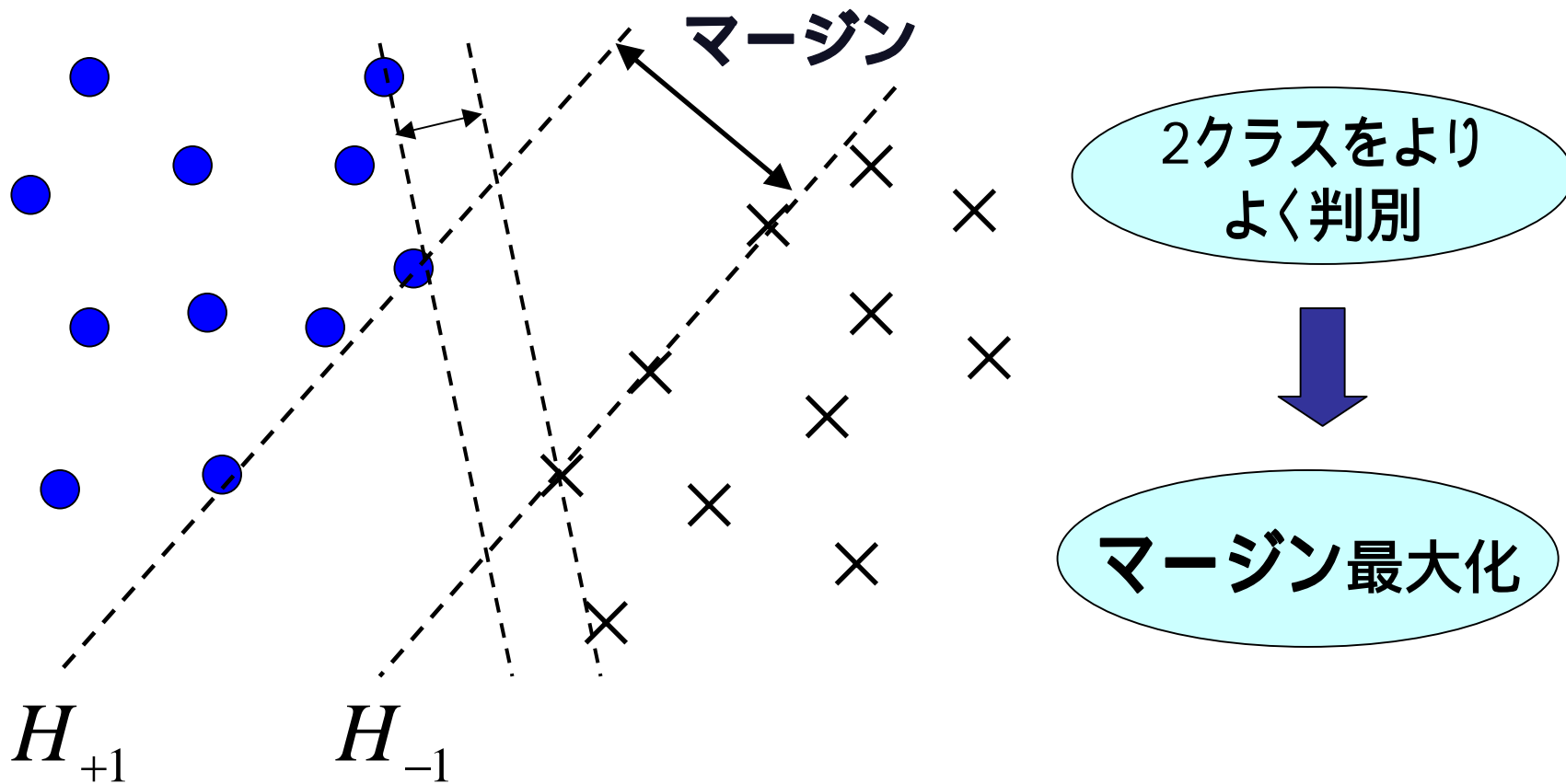


図9 サポートベクターとマージン

分離超平面の制約

・超平面 $\begin{cases} H_{+1} : \underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 = 1 \\ H_{-1} : \underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 = -1 \end{cases} \quad (i = 1, \dots, N)$

●のサンプル: $\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 \geq 1$

×のサンプル: $\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 \leq -1$

・マージンの距離 = $\frac{2}{\|\underline{\mathbf{w}}\|}$

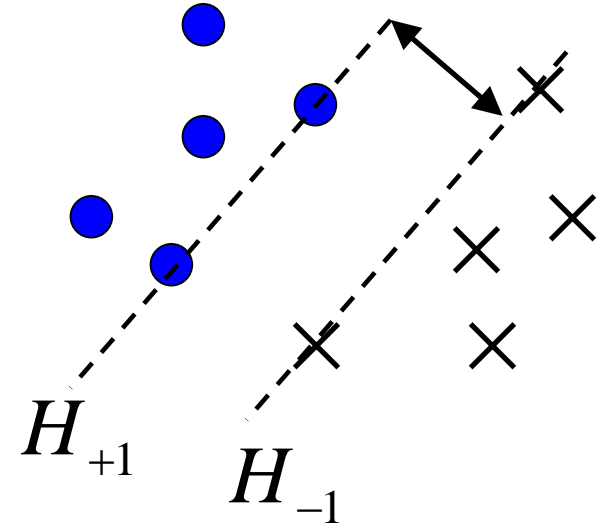
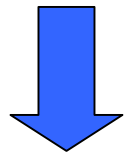


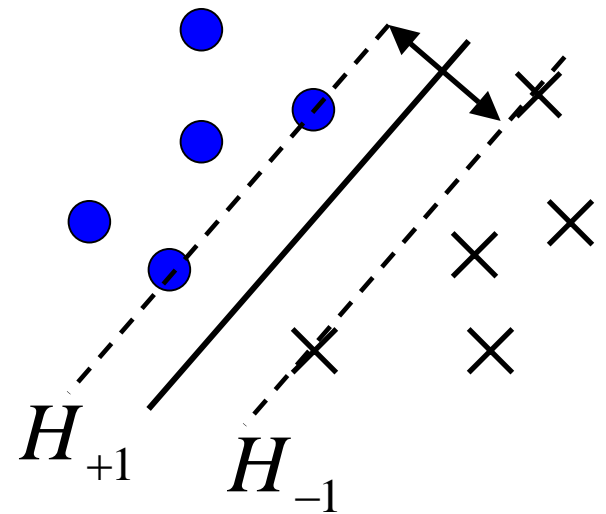
図10 超平面とサンプル

「学習」(\mathbf{w}, w_0 の決定) [5]

$$(HM) \left\{ \begin{array}{l} \text{Minimize}_{\mathbf{w}, w_0} \quad \frac{1}{2} \|\underline{\mathbf{w}}\|^2 \\ \text{s.t.} \quad \forall i, \quad d_i \cdot (\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0) \geq 1 \end{array} \right. \quad (i = 1, \dots, N)$$



分離超平面 $\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0 = 0$
(超平面 H_{+1}, H_{-1} の中間)



線形分離不可能

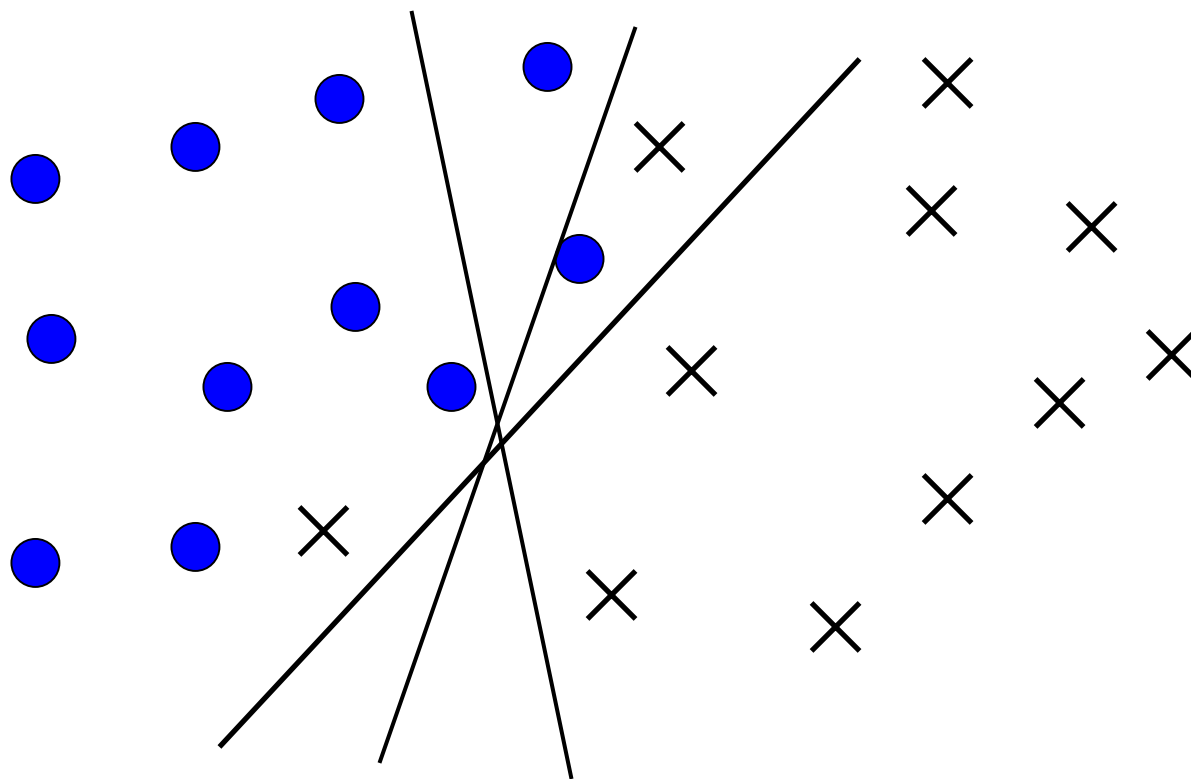
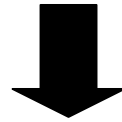


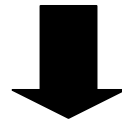
図11 線形分離不可能

[制約条件の弾力化]

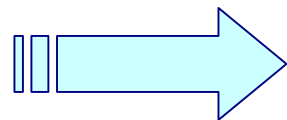
多少の誤りを許す



非負の補正量 $\xi_i (i=1, \dots, N)$ を導入



ξ_i を加えて $d_i \cdot (\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0) \geq 1$ の制約を満たす



$$d_i \cdot (\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0) + \xi_i \geq 1$$

「学習」(\mathbf{w}, w_0 の決定) [5]

Soft Margin

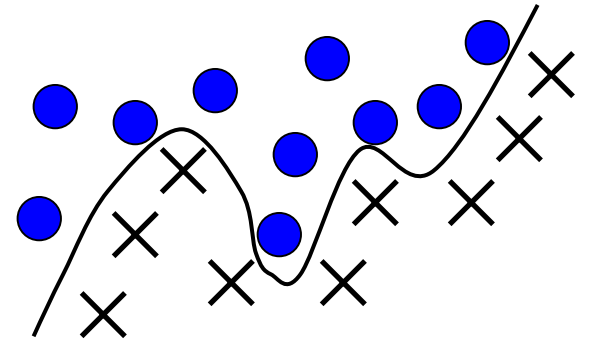
$$\begin{aligned}
 \text{(SM)} \quad & \left\{ \begin{array}{l} \text{Minimize}_{\mathbf{w}, w_0, \xi} \quad \frac{1}{2} \|\underline{\mathbf{w}}\|^2 + C \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad \forall i, \quad d_i \cdot (\underline{\mathbf{w}}^T \mathbf{x}_i - \underline{w}_0) + \xi_i \geq 1 \\ \quad \quad \forall i, \quad \xi_i \geq 0 \end{array} \right. \quad (i = 1, \dots, N)
 \end{aligned}$$

目的

補正量...小さく → マージン小さくなる

マージン...大きく → 補正量大きくなる

分離境界が非線形の場合



特徴量データ \mathbf{x}_i



高次空間 $\phi(\mathbf{x}_i)$

分離超平面を求める

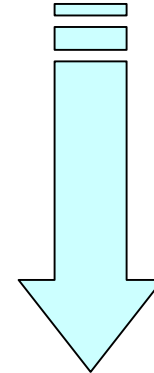
Soft Marginの2次計画問題

➡ 双対問題^[6]

$$(DSM) \quad \left\{ \begin{array}{l} \text{Maximize}_{\lambda} \quad \lambda^T \mathbf{1} - \frac{1}{2} \lambda^T (d_i d_j \mathbf{x}_i^T \mathbf{x}_j) \lambda \\ \text{s.t.} \quad \lambda^T \mathbf{d} = 0 \\ 0 \leq \lambda \leq C \mathbf{1} \end{array} \right.$$

$$(i, j = 1, \dots, N)$$

$$\lambda = (\lambda_1, \dots, \lambda_N)$$



$$\underline{\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)}$$

計算量の増加

カーネル関数

高次元での内積計算

$$\boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j)$$

||

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

カーネル関数

$$(i, j = 1, \dots, N)$$



元の次元での計算可能

ガウシアンカーネル: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\underline{\delta^2}}\right)$

判別関数(出力): $F(\mathbf{x}) = S\left(\sum_{i=1}^N d_i \cdot \lambda_i \cdot K(\mathbf{x}_i, \mathbf{x}) - w_0\right)$

4. 実験

アイリスデータ: 150個のサンプル

(花の種類: セトサ・バーシクル・バージニカ)

(1) 「学習」

終了時点で、最適な重みが得られたかどうか調べる。

) アヤメ2種類(出力)

・「セトサ・バーシクル」

特徴量(入力)	2次元	{ 「花びら長さ・花びら幅」 「がく片長さ・がく片幅」
	4次元	「花びら長さ・幅・がく片長さ・幅」

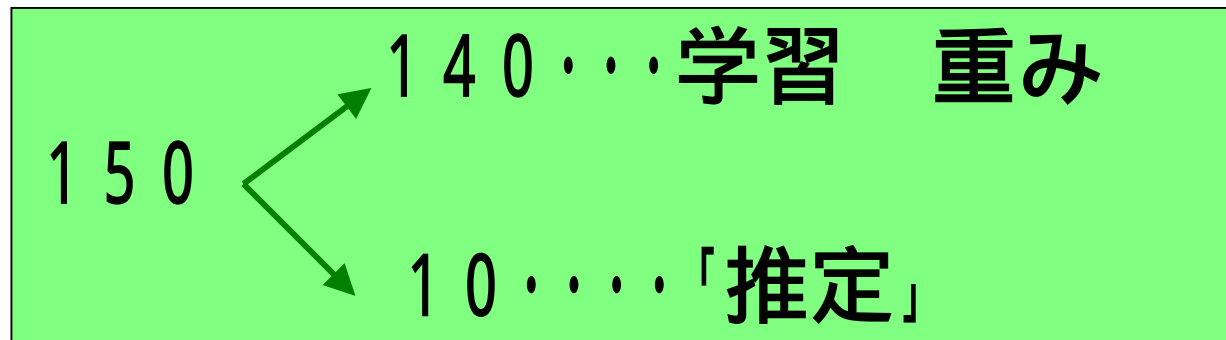
・「セトサ・バージニカ」

・「バーシクル・バージニカ」

) アヤメ3種類(出力)

(2) 「推定」

学習によって得られた重みを用いて、未知サンプルのクラスを推定し、その精度を調べる。



これを15回繰り返す

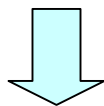
実験を行うにあたって(1)

[NN]

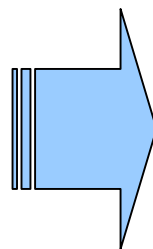
- ・「学習」と「推定」のプログラム作成

判別条件

特徴量 \mathbf{x} を入力



出力 { z_1 (セトサ)
 z_2 (バーシクル)
 z_3 (バージニカ)



最も「1」に近い花の
種類と判別

$$(0 \leq z_1, z_2, z_3 \leq 1)$$

実験を行うにあたって(2)

[SVM]

- ・「LINGO 8」の入力形式への変換プログラムを作成

判別条件

- ・Soft Marginの場合

$$\xi_i > 1 \quad \text{誤判別}$$

$$(i = 1, \dots, N)$$

- ・非線形の場合

$$d_i \cdot F(\mathbf{x}_i) < 0 \quad \text{誤判別}$$

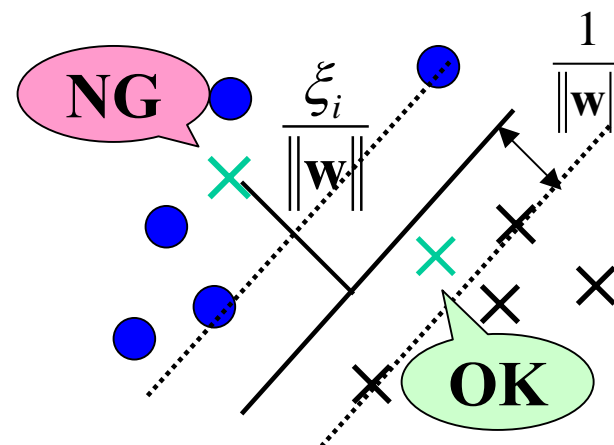


図12 SMの補正量

3クラス判別 場合分けが必要

カーネル関数 $\delta = 20$

5. 実験結果および考察(1)

(1) 「学習」の結果

) 出力が2種類

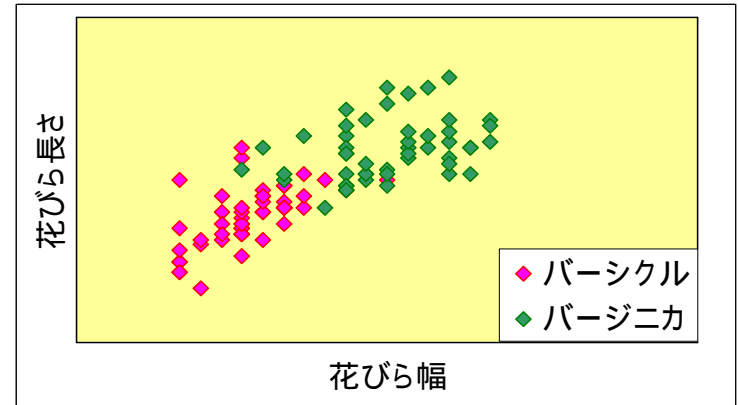
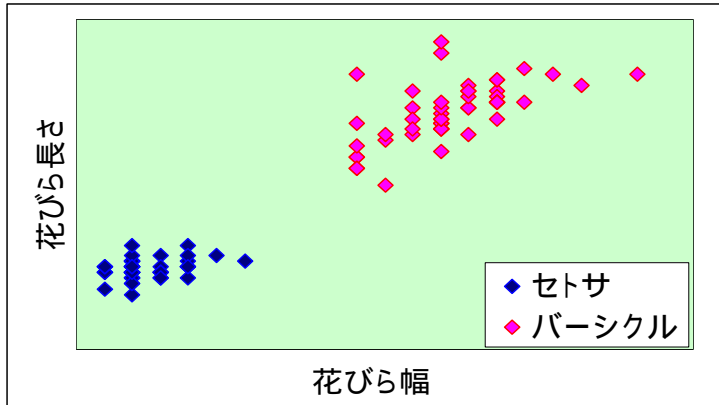


図13 「セトサとバーシクル」の分布

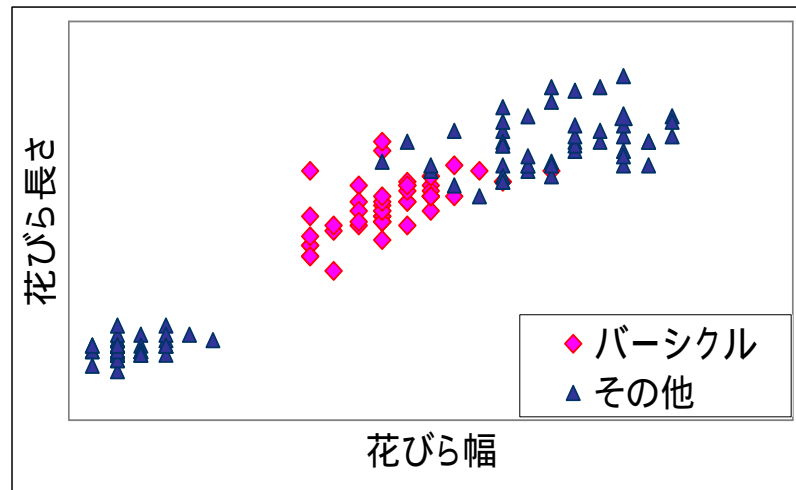
図14 「バーシクルとバージニカ」の分布

花びら長さ・花びら幅	正解数		
	NN	SVM	
		SoftMargin	Kernel
セトサ	50	50	
バーシクル	50	50	
セトサ	50	50	
バージニカ	50	50	
バーシクル	47	47	47
バージニカ	46	45	46

実験および考察(2)

) 出力が3種類

花びら長さ・花びら幅	正解数		
	NN	SVM	
		SoftMargin	Kernel
セトサ	50	50	50
バーシクル	46	0	48
バージニカ	47	46	46



がく片長さ・がく片幅 花びら長さ・花びら幅	図15 SVMによる「バーシクル or その他」の分布 正解数		
	NN	SVM	
		SoftMargin	Kernel
セトサ	50	50	50
バーシクル	47	26	49
バージニカ	49	48	48

実験および考察(3)

(2)「推定」の結果

総推定数: 150個(10個×15回)

[NN]

誤判別 5個/150

<教師データ>

<出力>

補正量 小

マージン 大

[SVM(kernel)]

$C = 10$

: 誤判別 4個/150

$C = 0.01 \sim 0.05$

: 誤判別 0個/150

誤判別サンプル: バーシクル, バージニカ

6. まとめ

〔 2つの判別手法 (NNとSVM)
の判別性能の比較評価 〕

1. クラス数

NN	3クラス以上の判別可	良
SVM	2クラスのみ判別 場合分け必要	悪

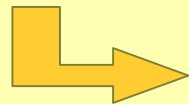
2. 手間

NN	特徴量データを入力するだけで非線形判別可能	良
SVM	C , δ , カーネル関数を 試行によって決定	悪

3. 推定精度

NN	「学習」結果をそのまま利用	悪
SVM	新たな分離超平面を得ることが可能	良

まず、NNを用いて推定



推定精度悪い



SVMで補う

7. 今後の課題

より数の多い学習データを用いてクラス判別を行い、2手法の比較をすることで、本研究の結果の信頼性を高めるといった課題が残っている。

このとき、NNでは誤差 E が極小値であるため、最小値でない可能性がある。そこで重みの初期値を様々に変えて試行する必要がある。

参考文献

- [1] Hiroshi Yadohisa: Data Representation System;
http://www.sci.kagoshimau.ac.jp/~drs/ja/data/fishers_iris/
- [2] 上坂吉則: ニューロコンピューティングの数学的基礎; 近代科学社
(1993)
- [3] 茨木俊秀, 福島雅夫: 最適化の手法; 共立出版, 情報数学講座第
14巻(1993)
- [4] 富田秀樹: 非線形多変量解析 ~ ニューラルネットによるアプローチ ~ ;
朝倉書店(1996)
- [5] 前田英作: 痛快! サポートベクトルマシン; 情報処理, 42巻7号,
pp. 676 - 683 (2001)
- [6] 栗田多喜夫: サポートベクターマシン入門;
<http://staff.aist.go.jp/takio-kurita/index-j.html>

付録

・ベイズ識別規則

特徴とクラスの確率的関係が完全にわかっている場合に、誤り確率が最小になるような識別境界を求める。

・重回帰分析

どの特徴がクラス判別に最も影響を与えているのかを知る。よって分布全体を考慮。

線形関数

分離境界決定に関して重要度

の低いサンプルをも考慮。

ロジスティック関数

非線形分離境界の場合、推定精度低下。

双対問題

2次計画問題にLagrange乗数 λ を導入すると，目的関数は

$$L(\mathbf{w}, w_0, \lambda) = \frac{1}{2} \sum_{k=1}^m w_k^2 + \sum_{i=1}^n \lambda_i (1 - d_i (\mathbf{w}^T \mathbf{x}_i - w_0))$$

である．KKT条件より最適点においては

$$\begin{cases} \mathbf{w} = \sum_{i=1}^N \lambda_i d_i \mathbf{x}_i \\ 0 = \sum_{i=1}^N \lambda_i d_i \end{cases}$$

が成り立たなくてはならない．

これらを $L(\mathbf{w}, w_0, \lambda)$ に代入して \mathbf{w}, w_0 を消去すると，

となる．

$$L(\mathbf{w}, w_0, \lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \lambda_i \lambda_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j$$

多項式カーネル関数

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \underline{q} \cdot \mathbf{x}_i^T \mathbf{x}_j)^{\underline{P}}$$

ガウシアンカーネル関数

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\underline{\delta}^2}\right)$$