「わかりやすいパターン認識」

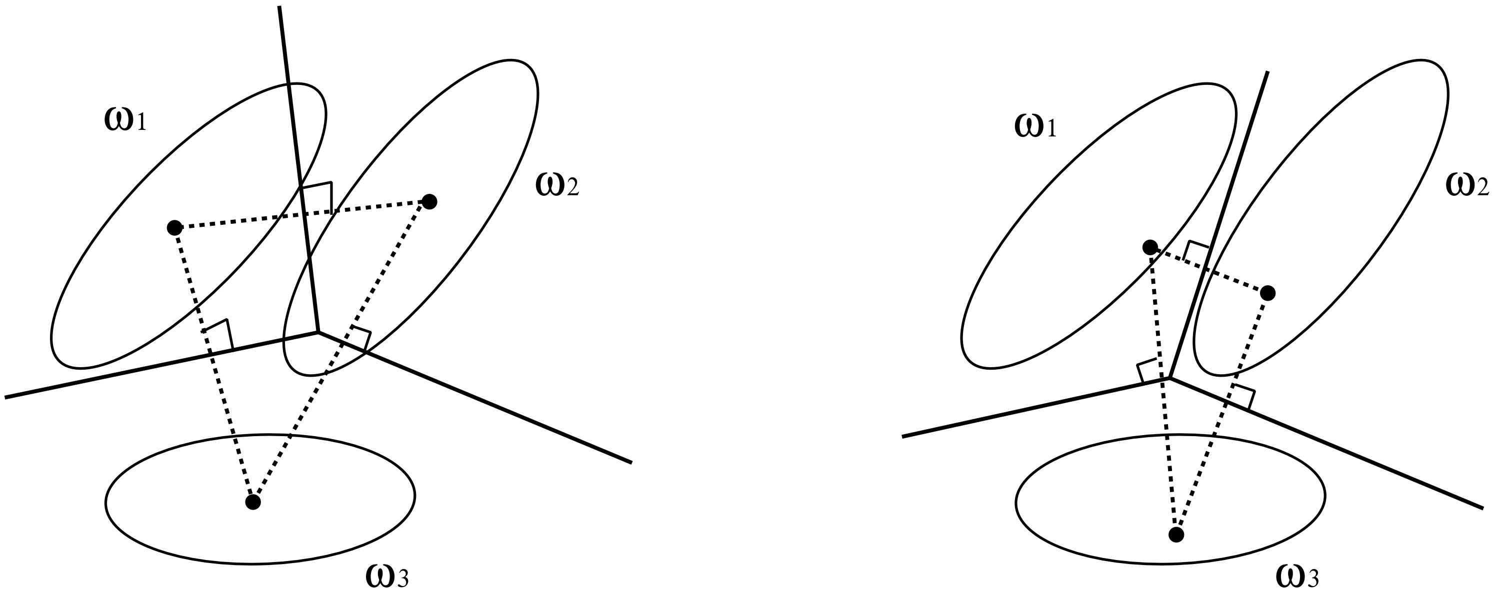
石井健一郎 他著 (オーム社出版局, 1998)

発表日: 平成15年4月18日
担当者: 岩崎 唯史
担当箇所: 第2章 学習と識別関数

- 2.1 学習の必要性
- 2.2 最近傍決定則と線形識別関数
- 2.3 パーセプトロンの学習規則
  - 1 重み空間と解領域
  - 2 パーセプトロンの収束定理
  - 3 学習とプロトタイプの移動

第 2 章 学習と識別関数

2.1 学習の必要性



(a) プロトタイプを重心に一致　　(b) プロトタイプを重心からずらす

図 2.1 プロトタイプの設定方法とクラス間分離の関係

■クラスの分離 (決定境界の設定)

プロトタイプを重心から直感的にずらす: 低次元○, 高次元×

➡ プロトタイプの位置を自動的に決定できないか?

➡ "学習 (learning)" or "訓練 (training)"

■学習パターン

識別部設計用 (図 1.1 参照) に収集されたパターン．訓練パターン，設計パターンともいう．

■ (統計的パターン認識における) 学習

学習パターンを用いて，学習パターンをすべて正しく識別できるようなクラス間分離面 (特徴空間の分割) を自動的に見出すこと．

学習パターンの数 $\geq$ プロトタイプの数

※等号時は全数記憶方式に対応

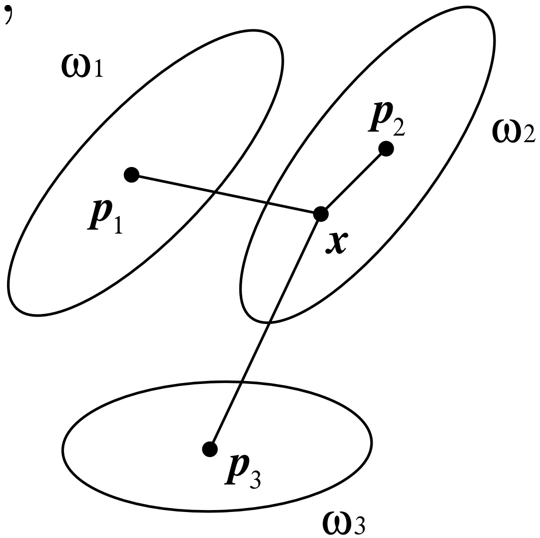
2.2 最近傍決定則と線形識別関数

■ 1 クラス当り 1 プロトタイプの NN 法 (最小距離識別法)

c 個のクラス $\omega_1, \omega_2, \ldots, \omega_c$ のプロトタイプを $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_c$ ($\boldsymbol{p}_i \in R^d$), 入力パターン (特徴ベクトル) を $\boldsymbol{x} \in R^d$ とする.
NN 法とは $\boldsymbol{x}$ と $\boldsymbol{p}_i$ との距離

$$\|\boldsymbol{x} - \boldsymbol{p}_i\|^2 = \|\boldsymbol{x}\|^2 - 2\boldsymbol{p}_i^{\mathrm{t}}\boldsymbol{x} + \|\boldsymbol{p}_i\|^2,$$

を最小にするプロトタイプ $\boldsymbol{p}_i$ を見出すこと.



$\Updownarrow$ 等価

■ (線形) 識別関数法

$$\text{識別関数:} \quad g_i(\boldsymbol{x}) \stackrel{\text{def}}{=} \boldsymbol{p}_i^{\mathrm{t}}\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{p}_i\|^2,$$

を各クラスに対応させ, 入力パターン x の属するクラスを判定.

$$\text{識別関数法:} \quad \max_{i=1,\ldots,c} \{g_i(\boldsymbol{x})\} = g_k(\boldsymbol{x}) \Longrightarrow x \in \omega_k.$$

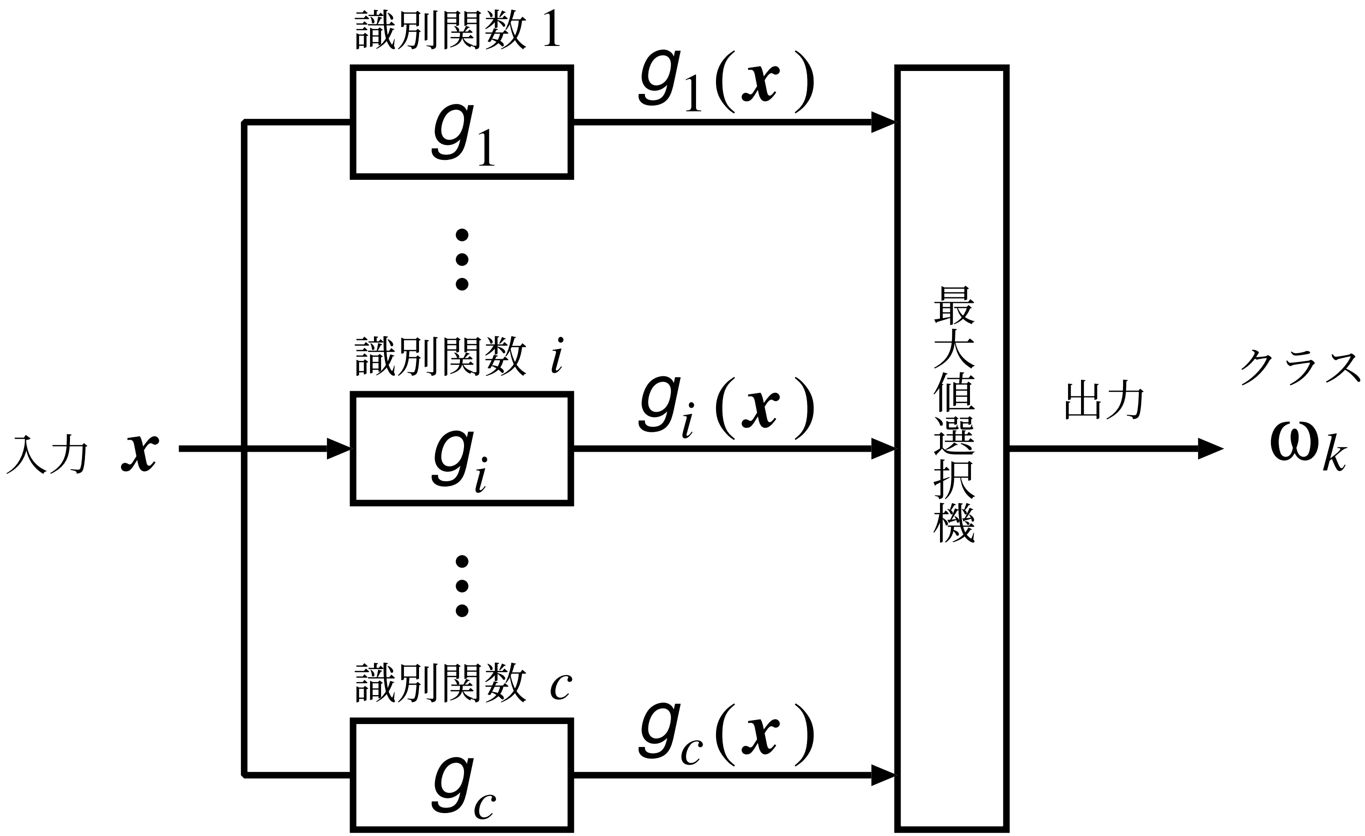


図 2.2 識別関数法による識別

⟨⟨ 一般的な線形識別関数 $g(\boldsymbol{x})$ ⟩⟩

$$g(\boldsymbol{x}) = w_0 + \sum_{j=1}^{d} w_j x_j = w_0 + \boldsymbol{w}^{\mathrm{t}}\boldsymbol{x} = \mathbf{w}^{\mathrm{t}}\mathbf{x}.$$

重みベクトル: $\boldsymbol{w} = (w_1, w_2, \ldots, w_d)^{\mathrm{t}}.$

拡張特徴ベクトル: $\mathbf{x} = (x_0, x_1, \ldots, x_d)^{\mathrm{t}} = \begin{pmatrix} x_0 \\ \boldsymbol{x} \end{pmatrix}, \quad x_0 \equiv 1.$

拡張重みベクトル: $\mathbf{w} = (w_0, w_1, \ldots, w_d)^{\mathrm{t}} = \begin{pmatrix} w_0 \\ \boldsymbol{w} \end{pmatrix}.$

クラス ω_i の線形識別関数: $g_i(\boldsymbol{x}) = \mathbf{w}_i^{\mathrm{t}}\mathbf{x}.$

クラス ω_i の拡張重みベクトル: $\mathbf{w}_i = \begin{pmatrix} w_{i0} \\ \boldsymbol{w}_i \end{pmatrix} = \begin{pmatrix} -\frac{1}{2}\|\boldsymbol{p}_i\|^2 \\ \boldsymbol{p}_i \end{pmatrix}.$

と置くと, 1 クラス当り 1 プロトタイプの NN 法 = 線形識別関数法.

■パーセプトロン*: 入力の線形和と最大値選択からなる識別系

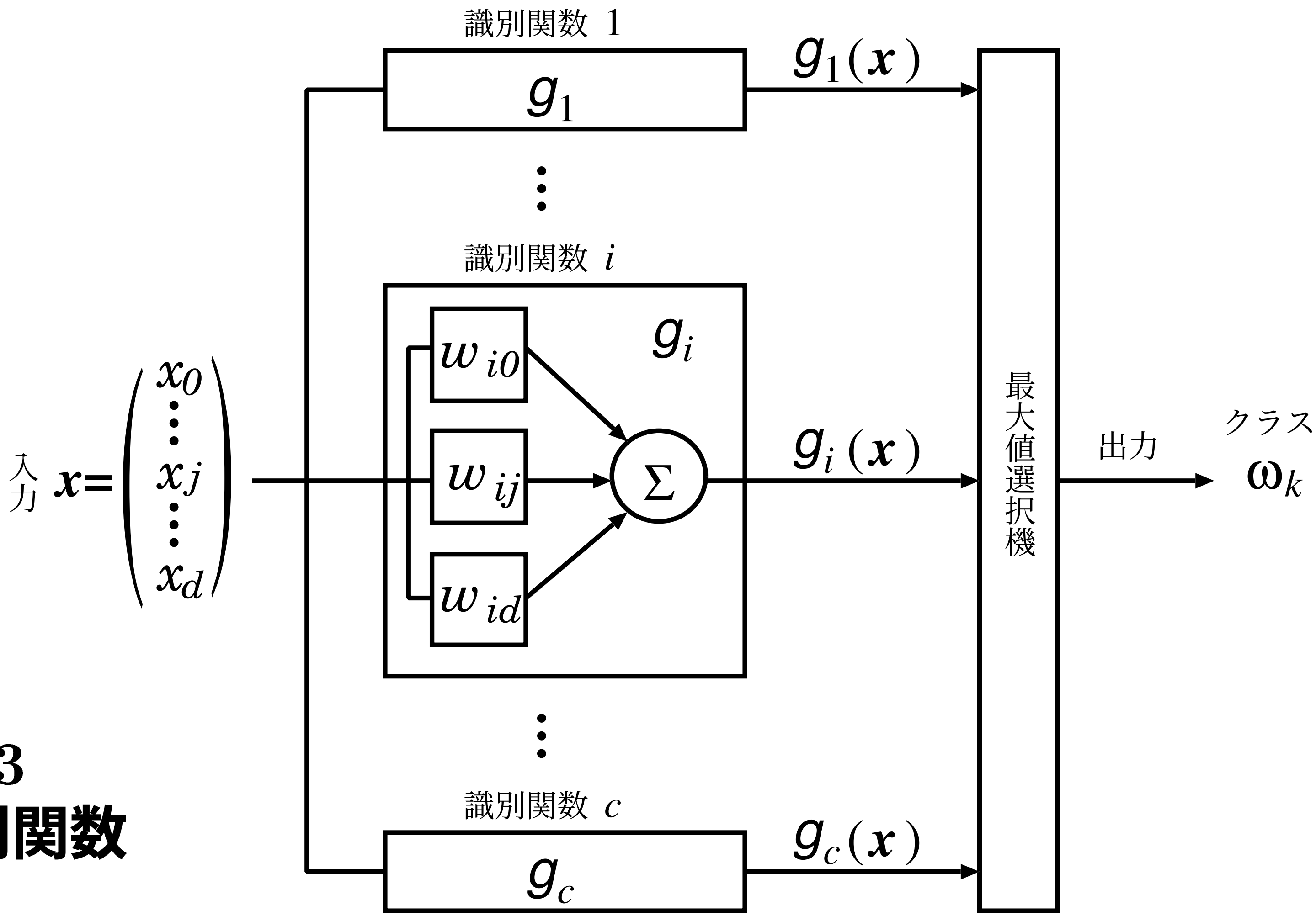


図 2.3
線形識別関数

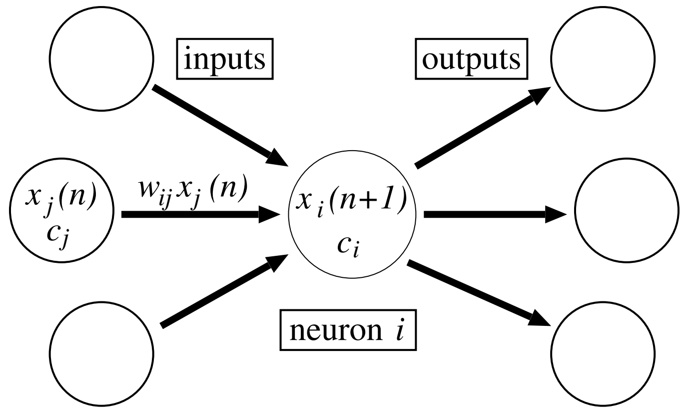
(*) 脳を模した学習能力のあるパターン認識機械. ある入力パターンからある出力パターンが出てくるように重み w_{ij} を学習.

ニューラルネットワークによる知的情報処理の歴史 (初期)

■ 1943 年 McCulloch & Pitts: 神経細胞モデル

線形閾値関数 (入出力関係は非線形)

$$x_i(n+1) = \theta(\sum_j w_{ij} x_j(n) - c_i).$$



状態 $x_i(n) = \{0, 1\}$, 結合強度 (重み)w_{ij}

■ 1949 年 Hebb: シナプス強化に関する仮説 (Hebb 則)

刺激を受けた神経細胞は結合強度が増大 $\Rightarrow$ 神経回路の可塑性

■ 1958 年 Rosenblatt: パーセプトロン (MP モデル+Hebb 則)

ある入力からある出力が得られるように重み w_{ij} を調節 (学習)

カテゴリー分類の線形分離可能性と学習収束定理

■ 1969 年 Minsky & Papert: パーセプトロンの限界を示す

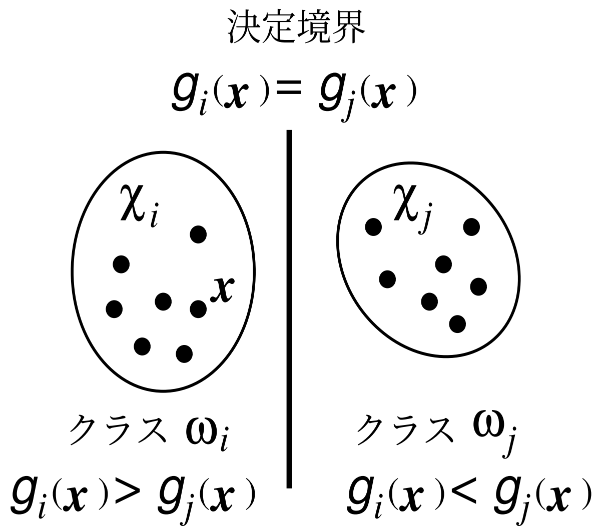
2.3 パーセプトロンの学習規則
[1] 重み空間と解領域

■**線形分離性**: 各クラスを超平面で分離できるか?

クラス $\omega_i (i = 1, 2, \ldots, c)$ に属する学習パターンの集合を χ_i とする.
線形識別関数 $g_i(\boldsymbol{x}) = \mathbf{w}_i^{\mathrm{t}}\mathbf{x} = w_{i0} + \boldsymbol{w}_i^{\mathrm{t}}\boldsymbol{x}$ に関して

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}), \quad \mathbf{for}\ {}^\forall \boldsymbol{x} \in \chi_i,\ {}^\forall j \neq i,$$

が成り立つような重み $\mathbf{w}_i = (w_{i0}, \boldsymbol{w}_i)$ を全てのクラス ω_i に対して決定できるとき, 「学習パターンは**線形分離性可能**」であるという.

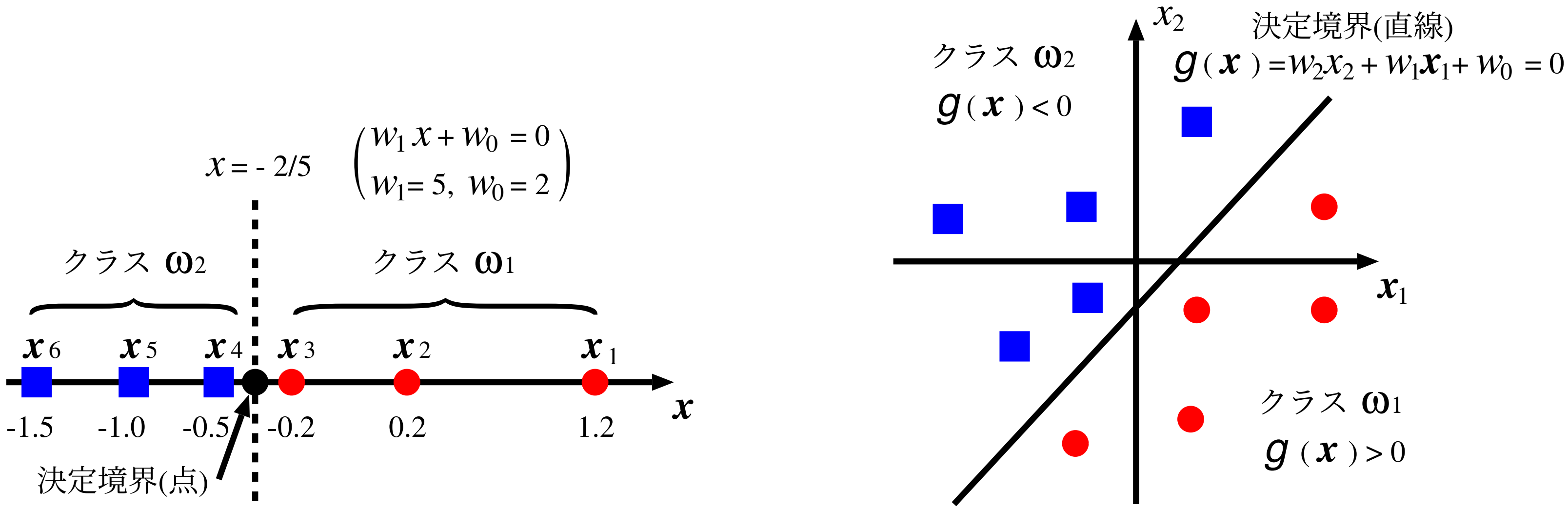


■**決定境界**

$g_i(\boldsymbol{x}) = g_j(\boldsymbol{x})$ $(\boldsymbol{x} \in R^d)$, すなわち

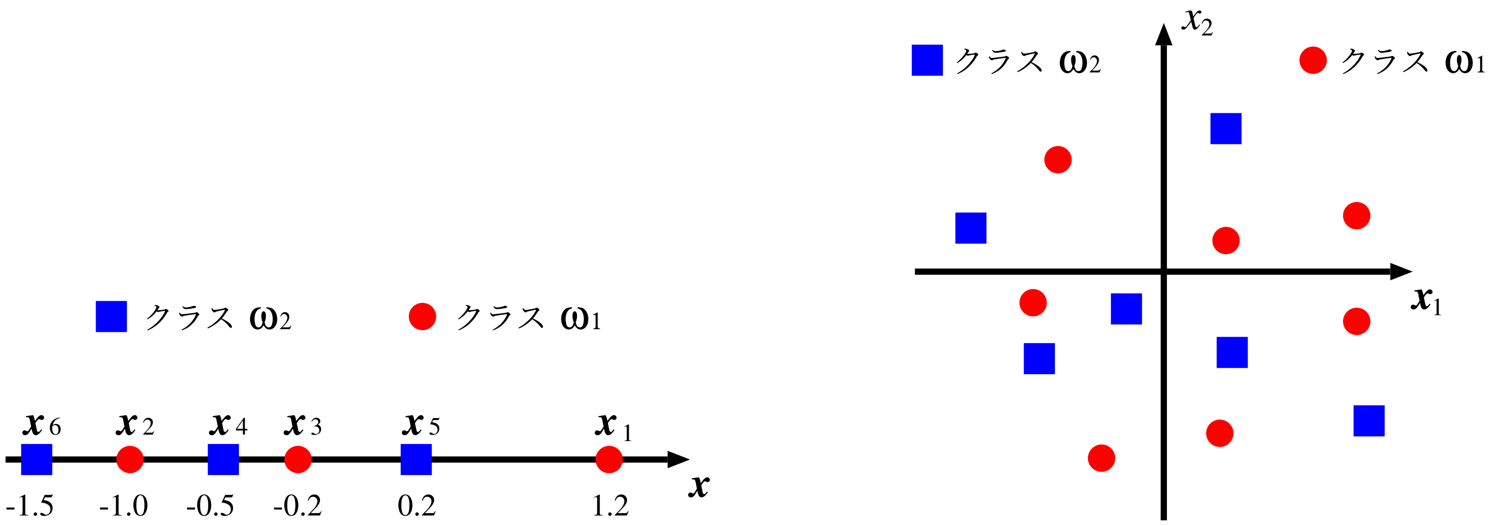
$$g(\boldsymbol{x}) \stackrel{\mathrm{def}}{=} g_i(\boldsymbol{x}) - g_j(\boldsymbol{x}) = (\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{t}}\mathbf{x} \stackrel{\mathrm{def}}{=} \mathbf{w}^{\mathrm{t}}\mathbf{x} = 0,$$

はクラス ω_i と ω_j の間の**決定境界** ($(d-1)$ 次元上の超平面) を与える.



$d = 1$ $d = 2$

図．線形分離性可能な学習パターンと決定境界の例 $(c = 2)$



$d = 1$ $d = 2$

図．線形分離性不可能な学習パターンの例 $(c = 2)$

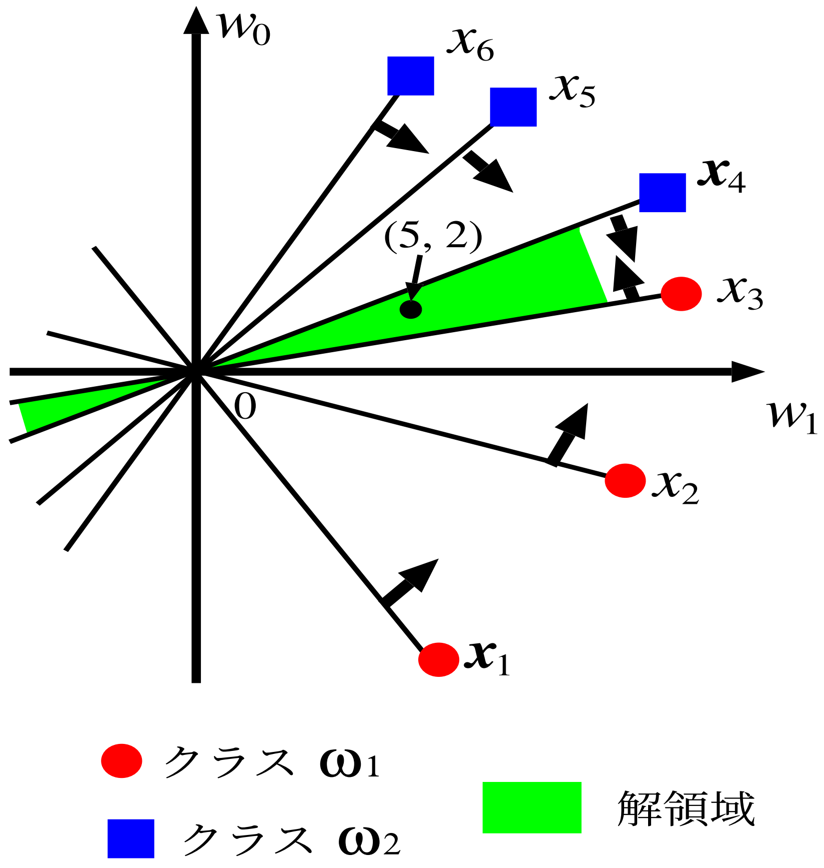
■重み空間: 重み $\mathbf{w} = (w_0, \boldsymbol{w})^{\mathrm{t}}$ の張る $(d+1)$ 次元空間

■解領域: 学習パターン $\mathbf{x}_i (i = 1, 2, \ldots, n)$ に対して定まる重み空間内の n 個の超平面 $\mathbf{w}^{\mathrm{t}}\mathbf{x}_i = 0$ で指定される $\mathbf{w}$ の存在すべき領域.

— $d = 1$, $c = 2$, $n = 6$ のときの重み空間と解領域の例 —

クラス	学習パターン	超平面*
ω_1	$x_1 = 1.2$	$w_0 = -1.2w_1$
ω_1	$x_2 = 0.2$	$w_0 = -0.2w_1$
ω_1	$x_3 = -0.2$	$w_0 = 0.2w_1$
ω_2	$x_4 = -0.5$	$w_0 = 0.5w_1$
ω_2	$x_5 = -1.0$	$w_0 = 1.0w_1$
ω_2	$x_6 = -1.2$	$w_0 = 1.2w_1$

(*) $d = 1$ のとき $\mathbf{w}^{\mathrm{t}}\mathbf{x}_i = w_1 x_i + w_0 = 0$



[2] パーセプトロンの収束定理

■パーセプトロンの学習規則 (誤り訂正法)

(1) 重みベクトル $\mathbf{w}$ の初期値を設定.

(2) χ の中から学習パターンを 1 つ選ぶ.

(3) $g(x) = \mathbf{w}^t\mathbf{x}$ により識別をし, 誤識別の場合 $\mathbf{w} \rightarrow \mathbf{w}'$ と修正.

$$\mathbf{w}' = \mathbf{w} + \rho\mathbf{x} \quad (\omega_1 \text{ を } \omega_2 \text{ と誤識別の場合})$$

$$\mathbf{w}' = \mathbf{w} - \rho\mathbf{x} \quad (\omega_2 \text{ を } \omega_1 \text{ と誤識別の場合})$$

(4) 処理 (2), (3) を χ の全パターンに対して繰り返す.

(5) χ の全パターンを正しく識別できたら終了．誤りがある場合には (2) に戻る.

■パーセプトロンの収束定理

χ が線形分離可能ならば, 上記のアルゴリズムは有限回の繰り返しで解領域内の重みベクトルに到達.

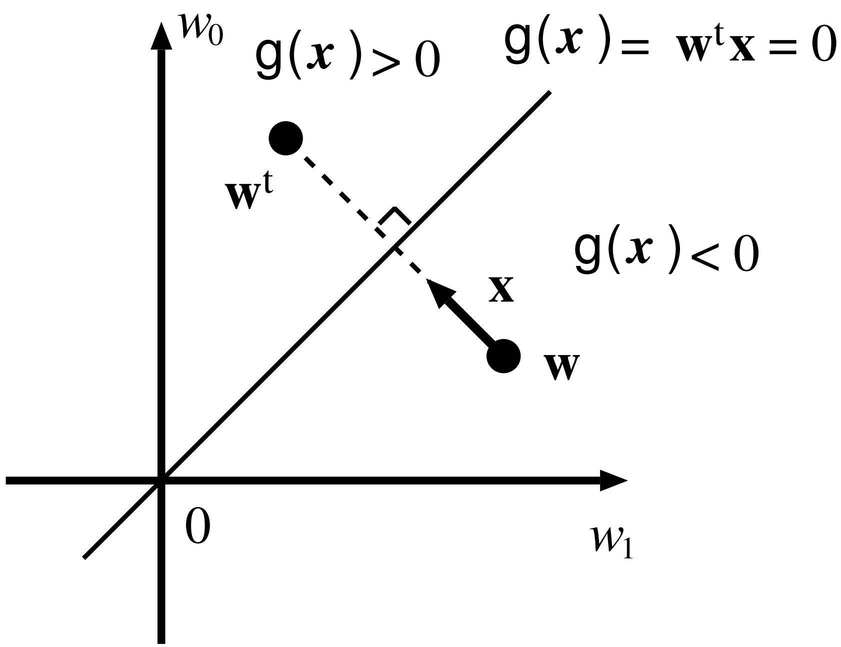


図 2.6 重みベクトルの修正

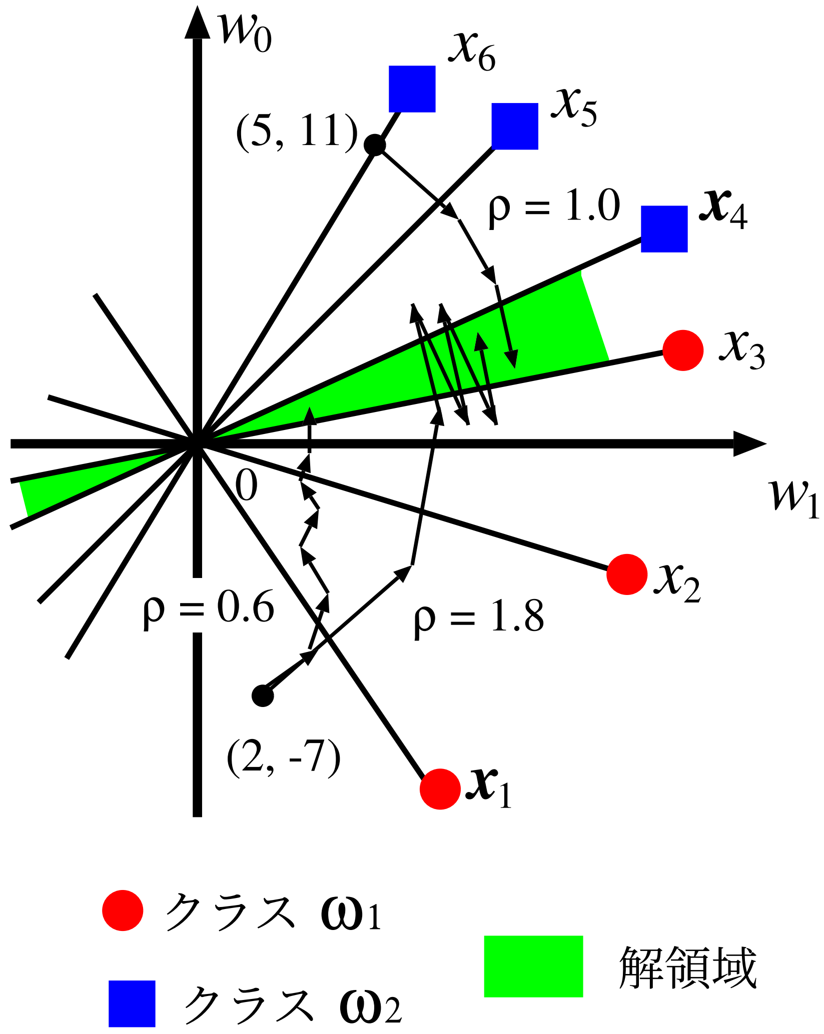


図 2.7 学習による重みベクトルの移動

■固定増分法

パーセプトロンの学習規則で ρ の値を固定

- ρ 小: 小刻みな修正で効率悪
- ρ 大: 振動修正で収束悪
- ρ 可変: 解領域への到達?

■多クラス $(c > 2)$ の場合の重みベクトルの修正

クラス ω_i に属する学習パターン x をクラス $\omega_j (i \neq j)$ に属すると誤った場合, それぞれの重みベクトルを

$$\begin{cases} \mathbf{w}_i' = \mathbf{w}_i + \rho \mathbf{x}, \\ \mathbf{w}_j' = \mathbf{w}_j - \rho \mathbf{x}. \end{cases}$$

とし, 識別関数 $g_i(\boldsymbol{x})$, $g_j(\boldsymbol{x})$ を修正.

[3] 学習とプロトタイプの移動

$w_{i0} = -\frac{1}{2}\|\boldsymbol{w}_i\|^2$を満たしながら重みベクトル$\mathrm{w}_i = (w_{i0}, \boldsymbol{w}_i)^t$を学習

➡ $\boldsymbol{w}_i$ の変化 ＝ 特徴空間上でのプロトタイプの移動

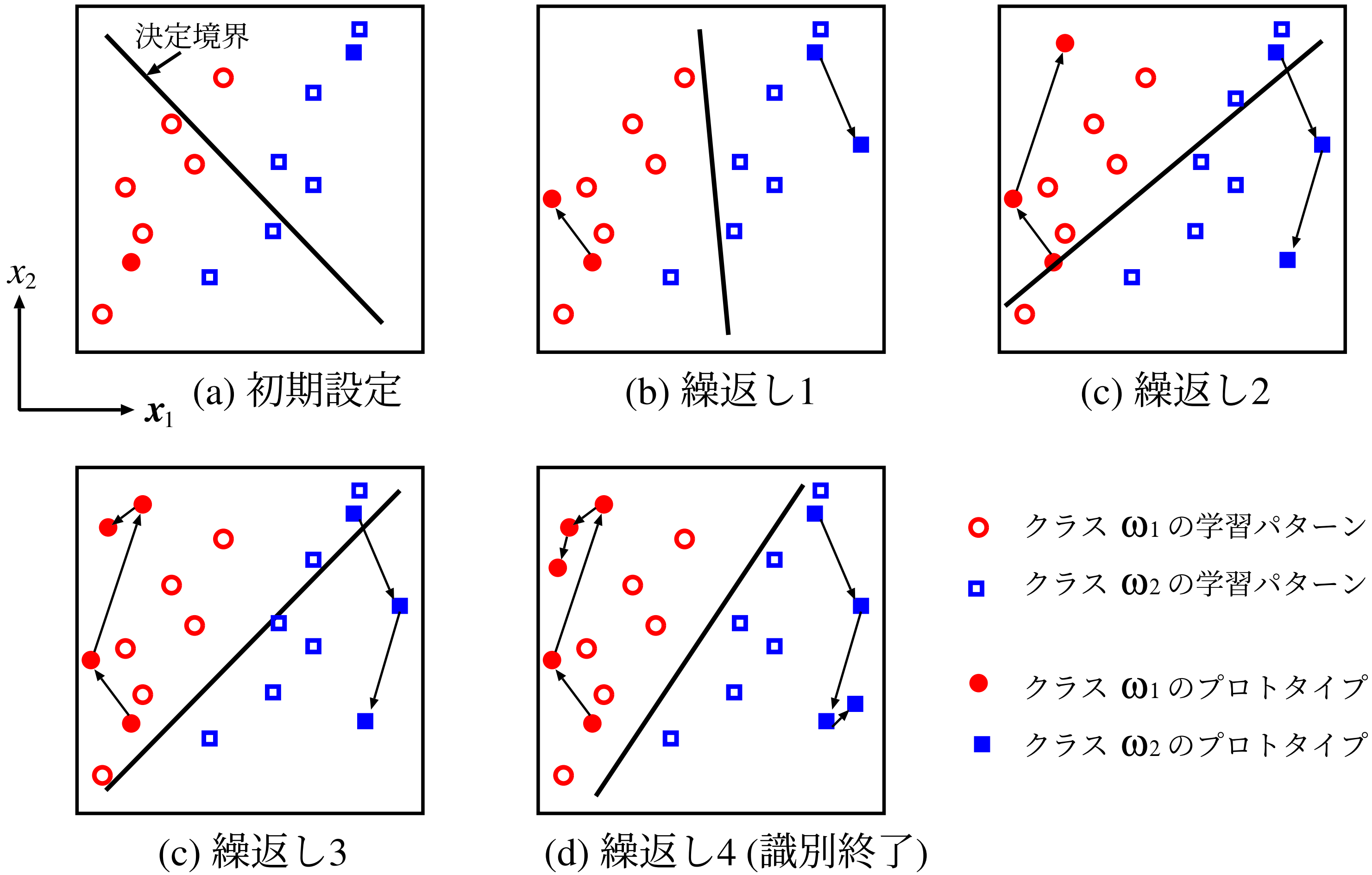


図 2.8 学習によるプロトタイプの移動