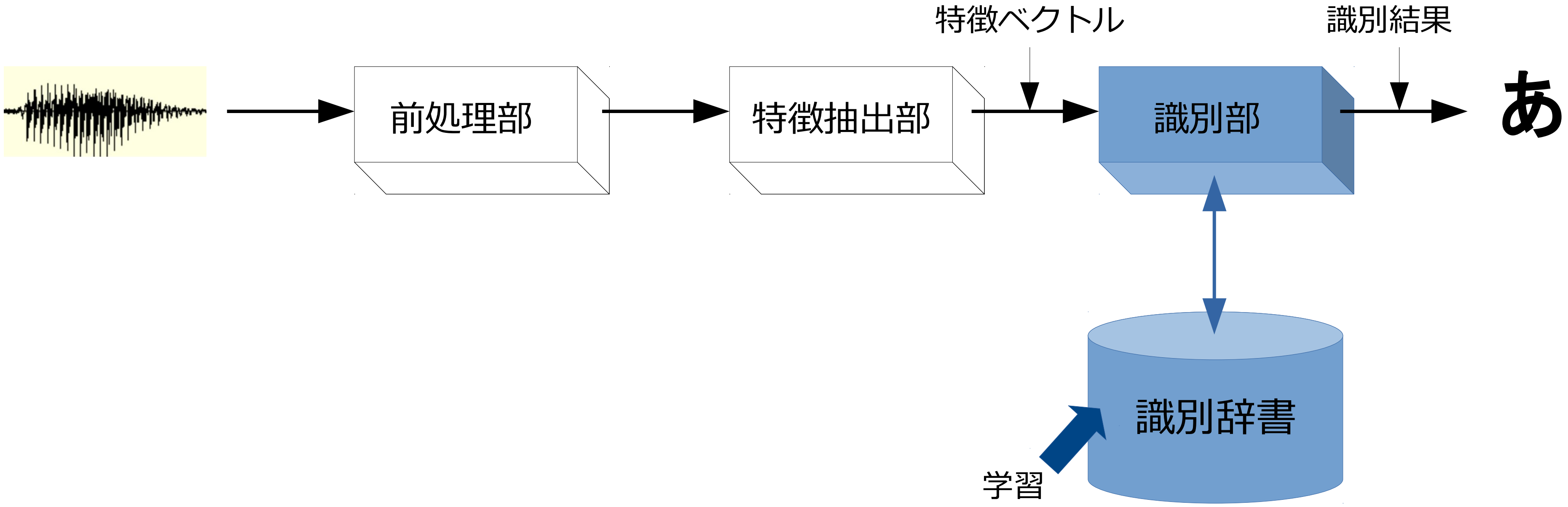
Section 2

- 基本的な識別手法 (4,5章)

4. パターンを識別しよう



4.1 NN 法の定式化と問題設定

4.1.1 「もっとも近い」の定義

- 識別対象のクラス：$\omega_1, \ldots, \omega_c$
- プロトタイプ
  - 各クラスの代表となる点

$$\boldsymbol{p}_i = (p_{i1}, \ldots, p_{id})^T \quad (i = 1, \ldots, c)$$

- 識別したい入力データ

$$\boldsymbol{x} = (x_1, \ldots, x_d)^T$$

4.1.1 「もっとも近い」の定義

- 入力ベクトルとプロトタイプとの距離

$$D(\boldsymbol{x}, \boldsymbol{p}_i) = \sqrt{(x_1 - p_{i1})^2 + \cdots + (x_d - p_{id})^2}$$

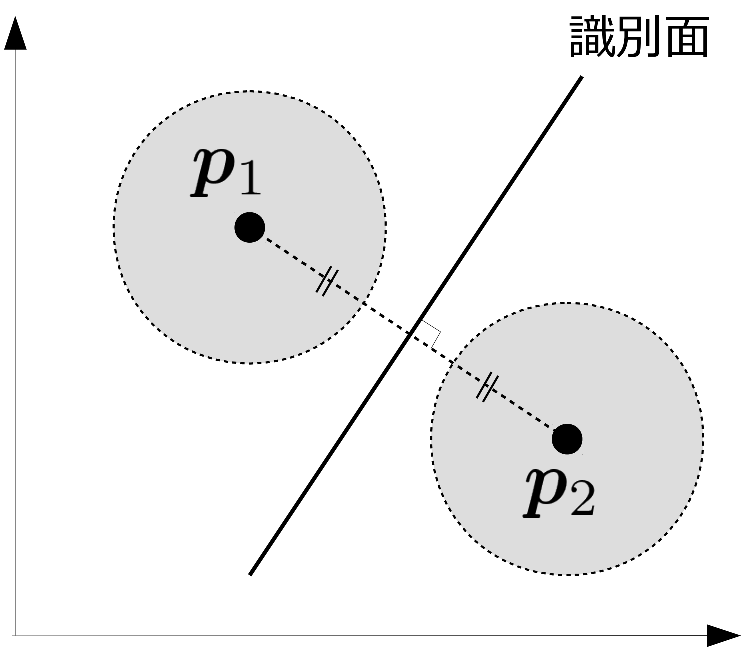
- NN 法の判定式

$$\arg \min_{i=1,\ldots,c} D(\boldsymbol{x}, \boldsymbol{p}_i) = k \quad \Rightarrow \quad \boldsymbol{x} \in \omega_k$$

最小値のときの
パラメータを返す
関数

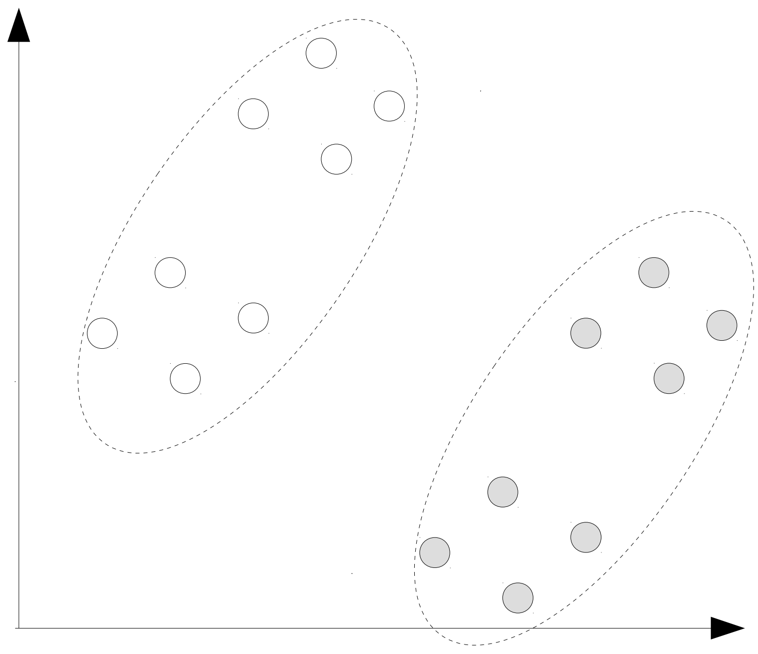
4.1.2 プロトタイプと識別面の関係

- 特徴空間の分割
  - 2 次元特徴の 2 クラス問題（$d=2,\ c=2$）を考える
  - クラスを分離する境界
    …プロトタイプから等距離にある領域
    - 2 次元の NN 法では垂直 2 等分線
    - 多次元では超平面
    - **決定境界**あるいは**識別面**と呼ぶ

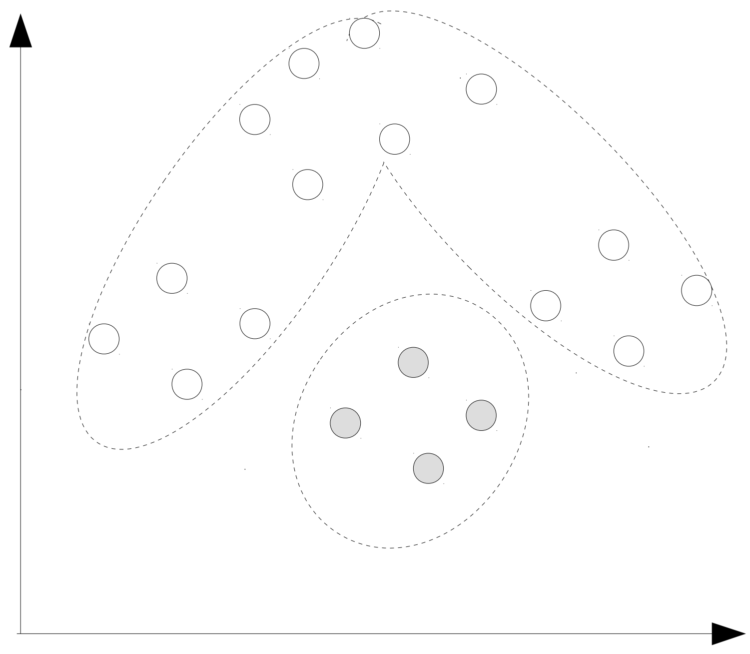


4.1.2 プロトタイプと識別面の関係

- 線形分離可能性
  - 直線（超平面）で 2 つのクラスが誤りなく分割できる場合を**線形分離可能**と呼ぶ



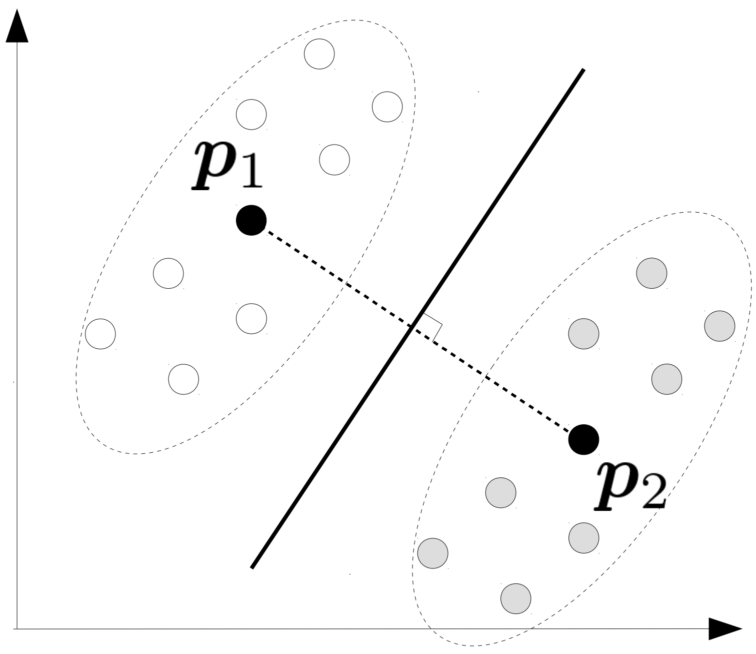
線形分離可能なデータ

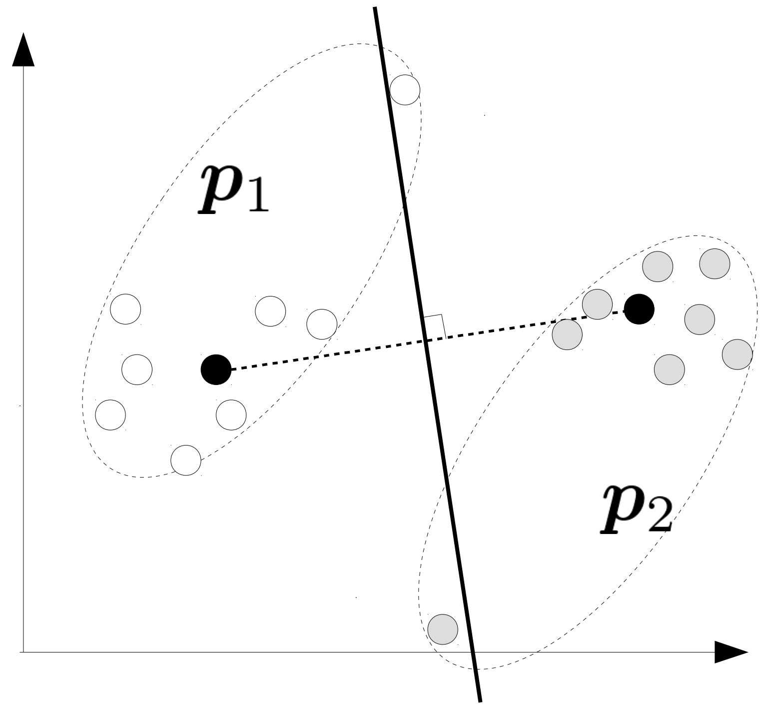


線形分離不可能なデータ

4.1.3 プロトタイプの位置の決め方

- クラスを代表するプロトタイプの設定法
  - 例）クラスの分布の重心

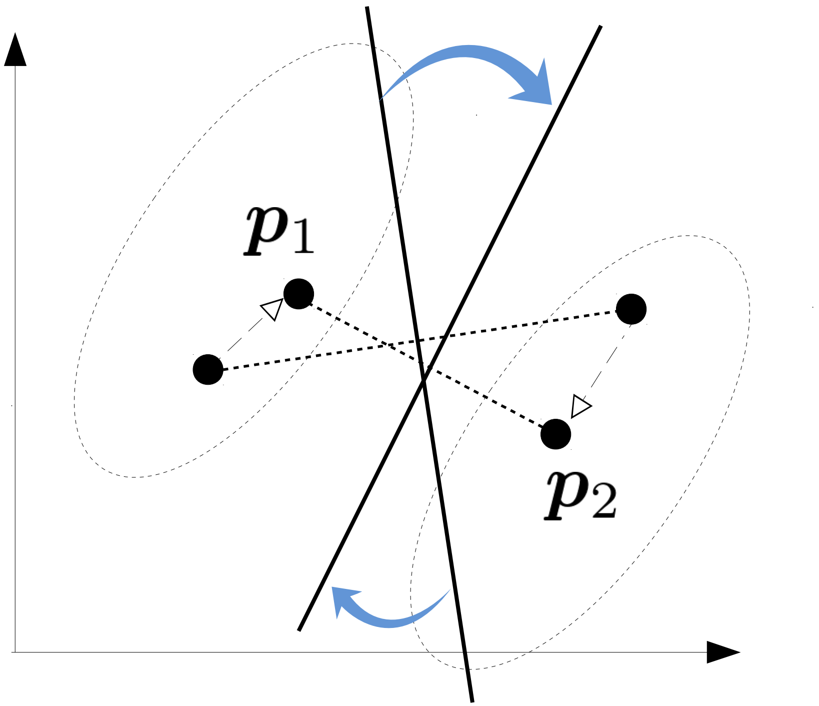




重心ではうまくゆかないことがある

4.2 パーセプトロンの学習規則

- 識別面の学習
  - 線形分離可能なデータに対して、識別誤りが生じない位置にプロトタイプを設定する



4.2.1　識別関数の設定

- 学習とは
  - プロトタイプの正しい位置を自動的に求めること
- 学習パターン
  - 識別部設計（特徴空間の分割）用に収集されたパターン
- 一般的な学習の定義
  - 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識別面を見いだすこと

4.2.1　識別関数の設定

- １クラス１プロトタイプの NN 法の定式化
  - クラス：$\omega_1, \ldots, \omega_c$
  - プロトタイプ： $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_c$
  - 入力パターン： $\boldsymbol{x}$ （特徴ベクトル）
  - NN 法：$D(\boldsymbol{x}, \boldsymbol{p}_i) = \|\boldsymbol{x} - \boldsymbol{p}_i\|$ を最小にする i を探す

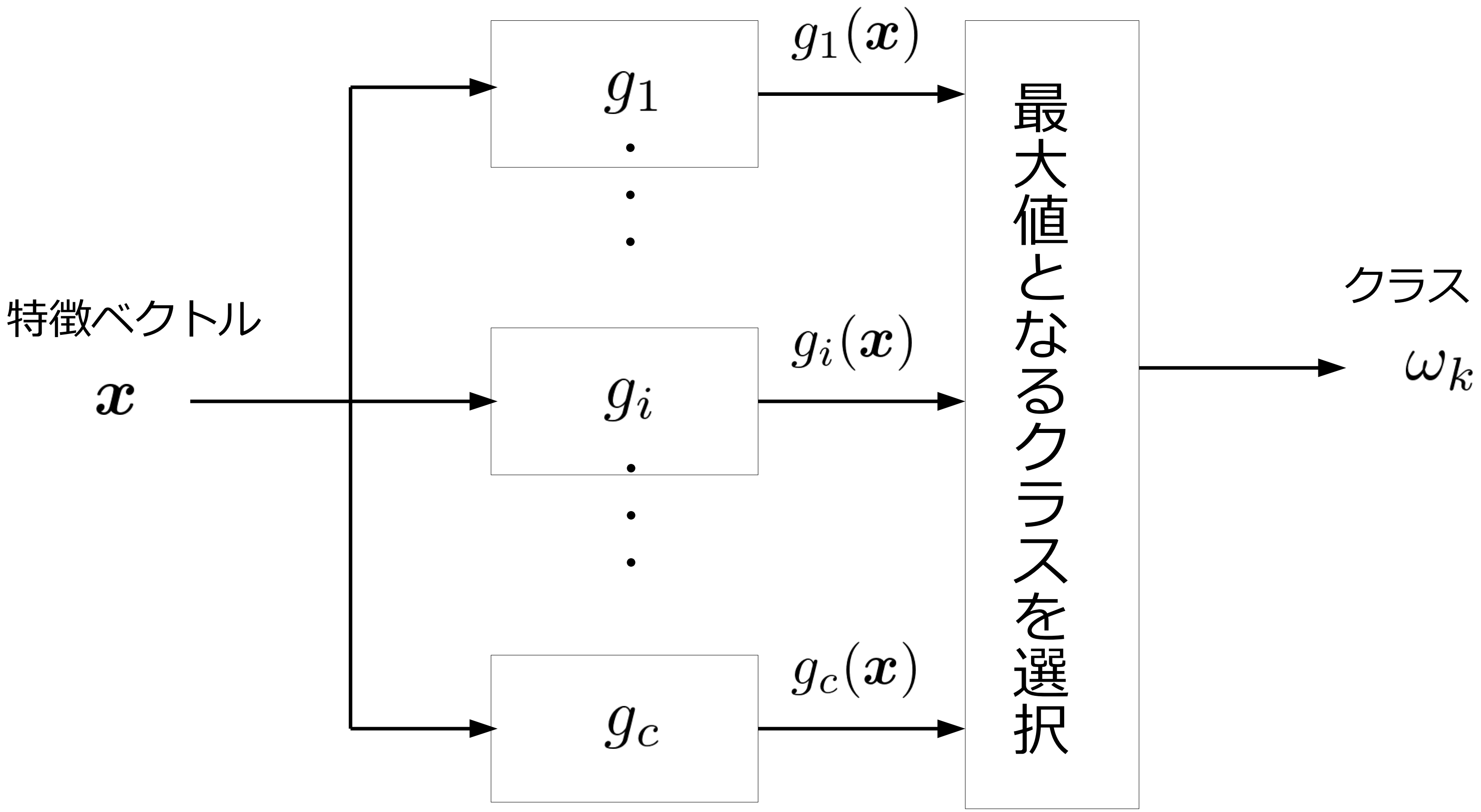
  $\rightarrow \; \|\boldsymbol{x} - \boldsymbol{p}_i\|^2 = \|\boldsymbol{x}\|^2 - 2\boldsymbol{p}_i^T \boldsymbol{x} + \|\boldsymbol{p}_i\|^2$

  $\rightarrow \; g_i(\boldsymbol{x}) = \boldsymbol{p}_i^T \boldsymbol{x} - \dfrac{1}{2}\|\boldsymbol{p}_i\|^2$ を最大にする i を探す

識別関数

4.2.1　識別関数の設定

- NN 法による識別部の実現



4.2.2　識別関数とパーセプトロン

- 線形識別関数
  - 識別関数の係数を

$$p_{ij} = w_{ij} \quad (j = 1, \ldots, d), \qquad -\frac{1}{2}\|\boldsymbol{p}_i\|^2 = w_{i0}$$

と置き換える

$$\begin{aligned} g_i(\boldsymbol{x}) &= w_{i0} + \sum_{j=1}^{d} w_{ij} x_j \\ &= \sum_{j=0}^{d} w_{ij} x_j \quad (x_0 \equiv 1) \\ &= \boldsymbol{w}_i^T \boldsymbol{x} \end{aligned}$$

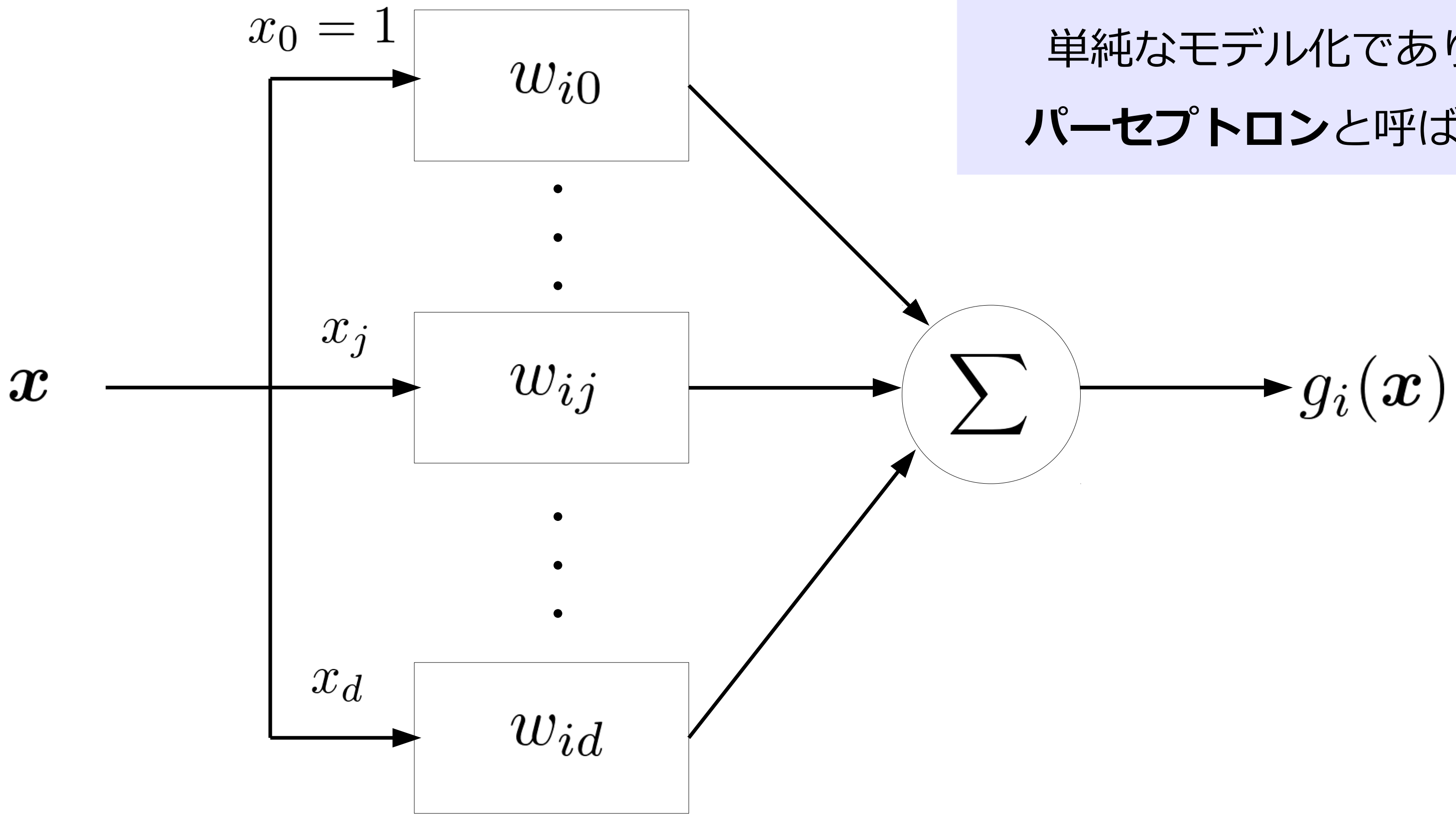
重みベクトル

$\boldsymbol{w}_i$, $\boldsymbol{x}$ は $d+1$ 次元

4.2.2　識別関数とパーセプトロン

- 線形識別関数の計算法

この計算機構は、神経細胞の
単純なモデル化であり、
パーセプトロンと呼ばれる



4.2.3 2 クラスの識別関数の学習

- 線形識別関数の学習
  - 学習パターン全体：χ
  - クラス ω_i に属する学習パターンの集合 χ_i の全ての要素 $\boldsymbol{x}$ に対して

$$g_i(\boldsymbol{x}) > g_j(\boldsymbol{x}) \quad (j = 1, \ldots, c,\ j \neq i)$$

が成り立つように重み $\boldsymbol{w}_i$ を決定する

4.2.3 2 クラスの識別関数の学習

- 2 クラスの場合
  - １つの識別関数

$$g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$$

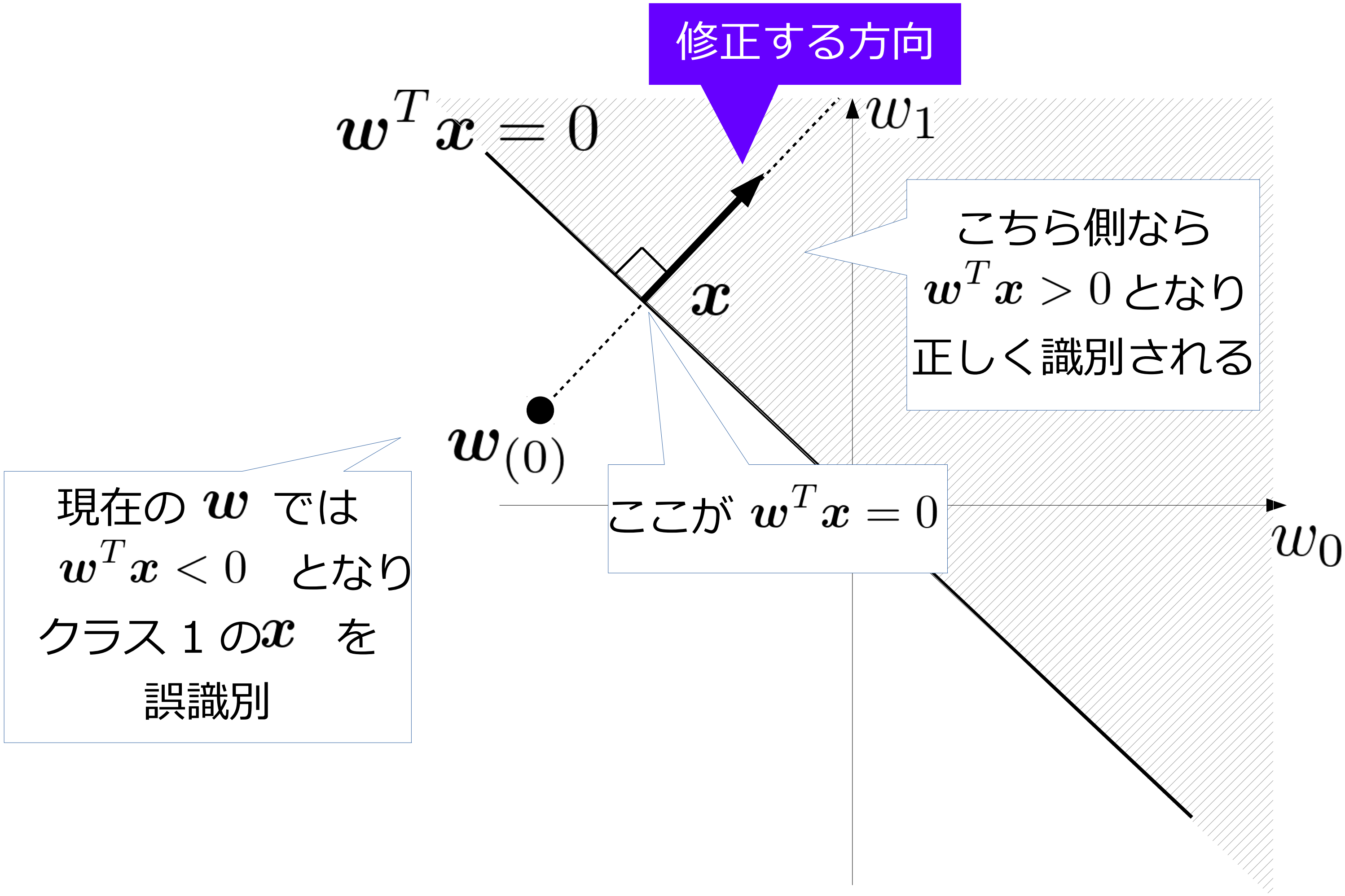
の正負を調べ、

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} \begin{cases} > 0 & (\boldsymbol{x} \in \chi_1) \\ < 0 & (\boldsymbol{x} \in \chi_2) \end{cases}$$

となる $\boldsymbol{w}$ を求める

4.2.3 2 クラスの識別関数の学習

- 重み空間での重みの修正



4.2.4 パーセプトロンの学習アルゴリズム

- パーセプトロンの学習規則

1. $\boldsymbol{w}$ の初期値を適当に決める
2. 学習パターンからひとつ $\boldsymbol{x}$ を選び、$g(\boldsymbol{x})$ を計算
3. 誤識別が起きたときのみ、$\boldsymbol{w}$ を修正する

$\boldsymbol{w}' = \boldsymbol{w} + \rho \boldsymbol{x}$ （クラス 1 のパターンをクラス 2 と誤ったとき）

$\boldsymbol{w}' = \boldsymbol{w} - \rho \boldsymbol{x}$ （クラス 2 のパターンをクラス 1 と誤ったとき）

学習係数

4. 2,3 を全ての学習パターンについて繰り返す
5. すべて識別できたら終了。そうでなければ 2 へ

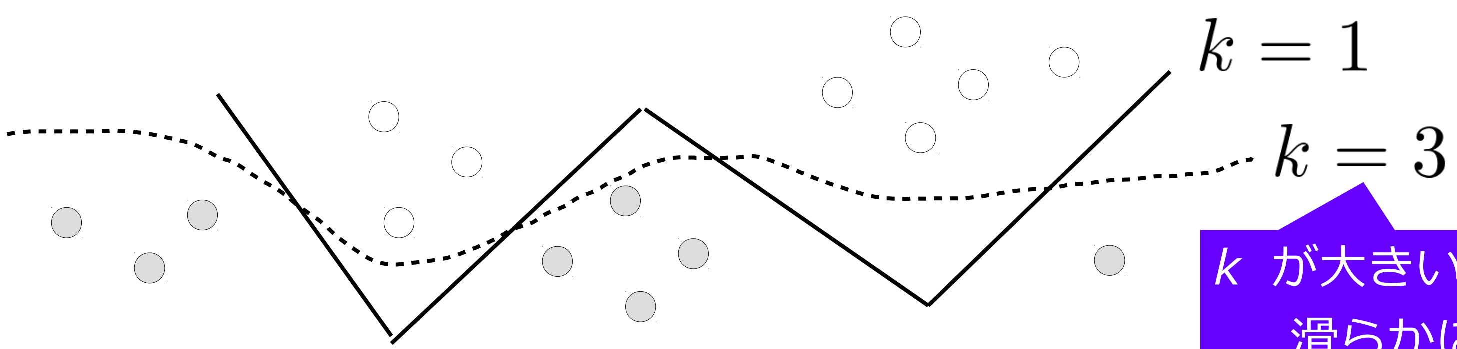
4.2.4　パーセプトロンの学習アルゴリズム

- パーセプトロンの収束定理
  - データが線形分離可能であれば、パーセプトロンの学習規則は有限回の繰り返しで終了する
- 学習係数 ρ の設定
  - 大きすぎると重みの値が振動する
  - 小さすぎると収束に時間がかかる

一般には小さい値が無難

4.3.4 学習をあきらめるのも一手　k-NN 法

- k- ＮＮ法とは
  - 全ての学習データをプロトタイプとする
  - 入力に近い順からｋ個のプロトタイプのクラスを調べ、多数決を取る
  - 実験の際のベースラインとして用いられる

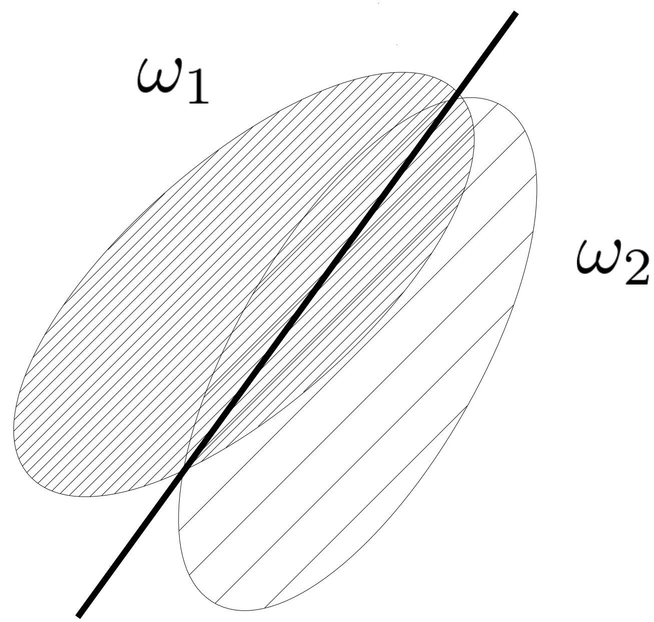


k が大きいと境界が滑らかになる

5. 誤差をできるだけ小さくしよう

- パーセプトロンの学習規則の欠点
  - 線形分離不可能である場合には利用できない
  - 一般に線形分離可能性を事前に確認するのは困難

- 評価関数最小化法
  - 線形分離不可能な場合にも適用可能



5.1　誤差評価に基づく学習とは

- 学習パターン $\chi = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$
- $\boldsymbol{x}_p \quad (1 \leq p \leq n)$　に対する c 個の識別関数の出力

  $(g_1(\boldsymbol{x}_p), \ldots, g_c(\boldsymbol{x}_p))^T$
- $\boldsymbol{x}_p$ に対する教師ベクトル（教師信号）

  $(b_{1p}, \ldots, b_{cp})^T$
  - 正解クラスの要素が１、他は０
- 入力パターン $\boldsymbol{x}_p$ に対する識別関数の出力と、教師信号との誤差 $\epsilon_{ip} \quad (i = 1, \ldots, c)$ が小さくなるように重みベクトル $\boldsymbol{w}$ を定める

5.1　誤差評価に基づく学習とは

- 誤差　$\epsilon_{ip} = g_i(\boldsymbol{x}_p) - b_{ip}$
- ϵ_{ip} の全クラスに対する二乗和を評価関数 J_p とする

$$
\begin{aligned}
J_p(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c) &= \frac{1}{2}\sum_{i=1}^{c} \epsilon_{ip}^2 \\
&= \frac{1}{2}\sum_{i=1}^{c} (g_i(\boldsymbol{x}_p) - b_{ip})^2 \\
&= \frac{1}{2}\sum_{i=1}^{c} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip})^2
\end{aligned}
$$

$\boldsymbol{x}_p$ に対する誤差

5.1　誤差評価に基づく学習とは

- 全パターンに対する二乗誤差 J

$$
\begin{aligned}
J(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c) &= \sum_{p=1}^{n} J_p(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c) \\
&= \frac{1}{2} \sum_{p=1}^{n} \sum_{i=1}^{c} (g_i(\boldsymbol{x}_p) - b_{ip})^2 \\
&= \underline{\frac{1}{2} \sum_{p=1}^{n} \sum_{i=1}^{c} (\boldsymbol{w}_i^T \boldsymbol{x}_p - b_{ip})^2}
\end{aligned}
$$

この値を最小にする $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_c$ を求める

5.1　誤差評価に基づく学習とは

- 2 クラス問題を考える場合
  - 識別関数を 1 つにできる

$$\mathrm{g}(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$$

  - 教師信号はクラス 1 のとき 1 、クラス 2 のとき -1
  - 誤差関数

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{p=1}^{n}(\boldsymbol{w}^T\boldsymbol{x}_p - b_p)^2$$

5.2 解析的な解法

- パターン行列

$$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^T$$

教師信号ベクトル

$$\boldsymbol{b} = (b_1, \ldots, b_n)^T$$

とすると

$$J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{b}\|^2$$

$$\frac{\partial J}{\partial \boldsymbol{w}} = \underbrace{\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{b}) = 0}_{\text{解くべき式}}$$

5.2 解析的な解法

- 解くべき式

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{b}$$

$\boldsymbol{X}^T\boldsymbol{X}$が正則であるとき

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{b}$$

正則：
逆行列が存在すること
逆行列：
$n \times n$ の正方行列 A に対して、
$AB = BA = I$ となる B

最小二乗法

- 解が求まらない可能性
  - $\boldsymbol{X}^T\boldsymbol{X}$が正則であるとは限らない
  - d が大きい場合は逆行列を求める計算が大変

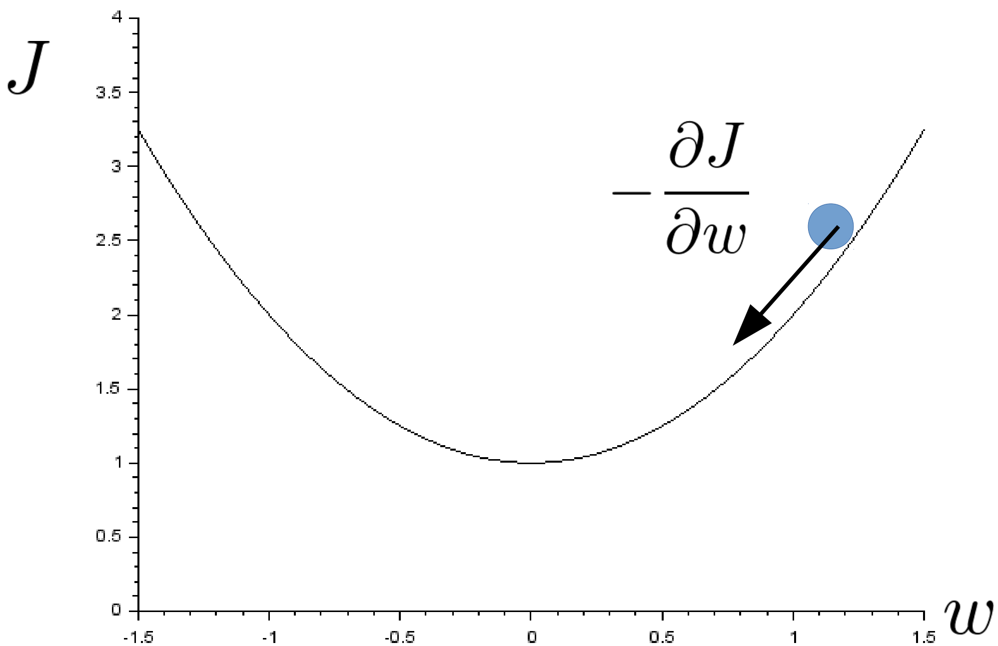
5.3 最急降下法

5.3.1 最急降下法による最適化

- $\boldsymbol{w}$ を $\boldsymbol{J}$ の傾きの方向に徐々に修正する

$$\boldsymbol{w}' = \boldsymbol{w} - \rho \frac{\partial J}{\partial \boldsymbol{w}}$$

- 最急降下法のイメージ



5.3.2 Widrow-Hoff の学習規則

- 勾配ベクトルの定義
  - 重みベクトル

$$\boldsymbol{w} = (w_0, \ldots, w_d)$$

の関数 $J(\boldsymbol{w})$ に対して、勾配ベクトルを

$$\nabla J = \frac{\partial J}{\partial \boldsymbol{w}} = (\frac{\partial J}{\partial w_0}, \ldots, \frac{\partial J}{\partial w_d})^T$$

と定義する

5.3.2 Widrow-Hoff の学習規則

- 修正式の導出

$$\frac{\partial J}{\partial \boldsymbol{w}} = \sum_{p=1}^{n} \frac{\partial J_p}{\partial \boldsymbol{w}}$$

$$= \sum_{p=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}_p - b_p) \boldsymbol{x}_p$$

$$\boldsymbol{w}' = \boldsymbol{w} - \rho \frac{\partial J}{\partial \boldsymbol{w}}$$

$$= \underline{\boldsymbol{w} - \rho \sum_{p=1}^{n} (\boldsymbol{w}^T \boldsymbol{x}_p - b_p) \boldsymbol{x}_p}$$

重みの修正式

5.3.3 確率的最急降下法

- 最急降下法の問題点
  - データ数やパラメータ数が多いと、重み更新に時間がかかる
- 確率的最急降下法
  - 個々のデータの識別結果に基づき、重みを更新
  - データが来る毎に学習するオンライン学習が可能
  - 更新式

$$\boldsymbol{w}' = \boldsymbol{w} - \rho(\boldsymbol{w}^T \boldsymbol{x}_p - b_p)\boldsymbol{x}_p$$

Section2 のまとめ

- パーセプトロンの学習規則
  - データが線形分離可能な場合、有限回の重みの修正で線形識別面を見つけることができる
- 誤差最小の識別面の学習
  - 最小二乗法：解析的に識別面を求めることができる
  - 最急降下法：誤差を徐々に小さくする