

# 機械学習 第4回 識別（2）

立命館大学 情報理工学部

福森 隆寛

Beyond Borders

# 講義スケジュール

## □ 担当教員 1 : 福森 (第1回～第15回)

1	機械学習とは、機械学習の分類
2	機械学習の基本的な手順
3	識別 ( 1 )
4	<b>識別 ( 2 )</b>
5	識別 ( 3 )
6	回帰
7	サポートベクトルマシン
8	ニューラルネットワーク

9	深層学習
10	アンサンブル学習
11	モデル推定
12	パターンマイニング
13	系列データの識別
14	半教師あり学習
15	強化学習

## □ 担当教員 2 : 叶昕辰先生 (第16回の講義を担当)

# 今回の講義内容

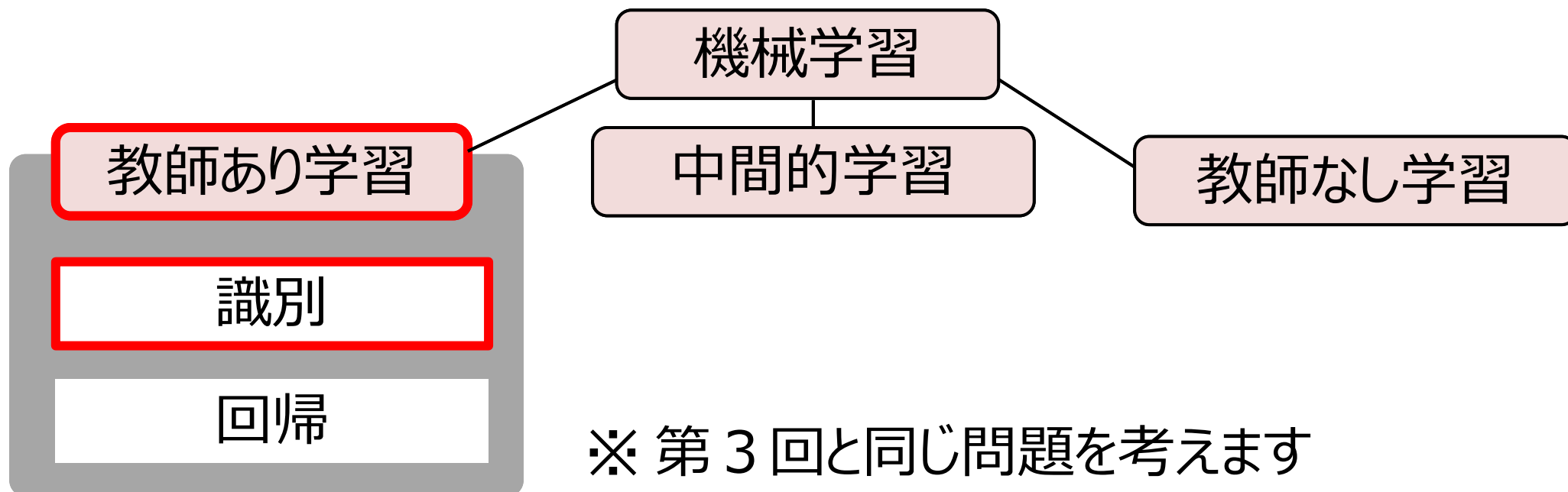
- 取り扱う問題の定義
- 最小二乗法
- さいきゅうこうかほう 最急降下法
- Widrow-Hoffの学習規則
- かくりつてき 確率的な最急降下法
- 演習問題

# 取り扱う問題の定義：教師あり・識別問題

- カテゴリデータ、または数値データからなる特徴ベクトルを入力して、それをクラス分けする識別器を作る

※ 教師あり学習の識別問題での学習データは、以下のペアで構成される

入力データの特徴ベクトル  $\leftarrow \{ \underline{x_i}, \underline{y_i} \}, \quad i = 1, 2, \dots, \underline{N} \longrightarrow$  学習データの総数  
(カテゴリデータ/数値データ)      カテゴリ形式の正解情報  $\rightarrow$  「クラス」と呼ぶ



# 区分的線形識別

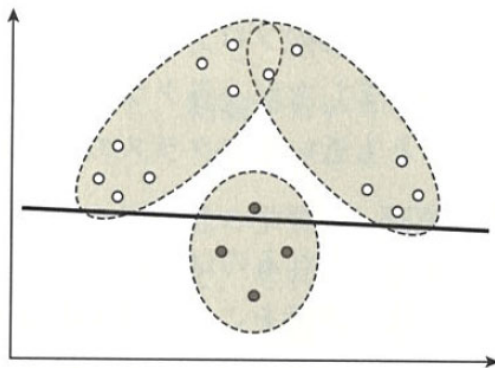
## □ パーセプトロンの学習規則

- 特徴空間上の学習データが線形分離可能ならば識別面を発見できる

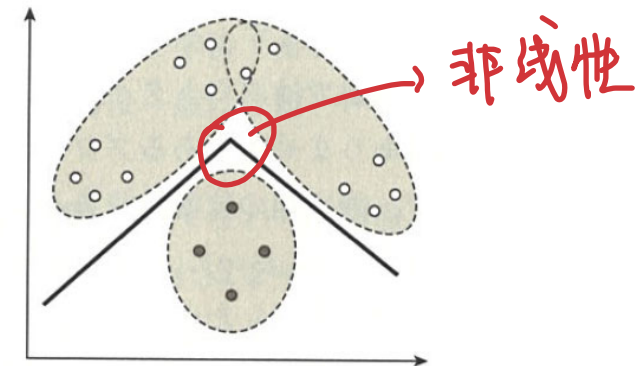
- 線形分離不可能な場合（1つの超平面で区切れない場合）は？
  - 識別面を折り曲げれば学習データを分離できる

## ■ 区分的線形 分段线性

- 折れ曲がっている部分だけが非線形で、それ以外の区間は線形



線形分離不可能なデータ



区分的線形識別を用いた場合

# 区分的線形識別面の定式化

使用两个代表点就可以

生成两个判别面

## □ 区分的線形識別面の実現

### ■ 2つの超平面 (2次元の場合は2本の直線<sup>ちよくせん</sup>) を繋ぐ

- 1つのクラスに対して、2個のプロトタイプを用意すれば  
2つの線形識別面ができる

– 右図の場合

» クラス $\omega_1$ のプロトタイプ :  $p_{11}$ 、 $p_{12}$

» クラス $\omega_2$ のプロトタイプ :  $p_{21}$ 、 $p_{22}$

» 1つ目の識別面 :  $p_{11}$ 、 $p_{21}$ から求める

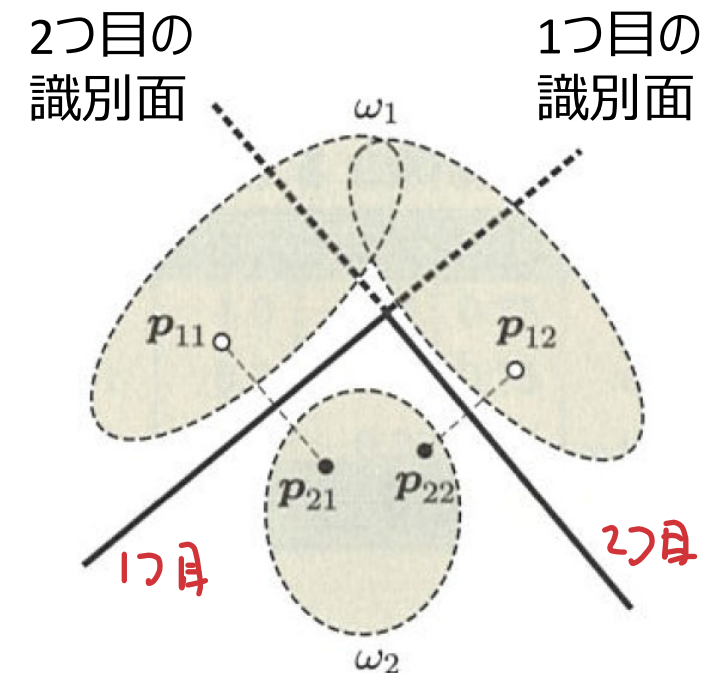
» 2つ目の識別面 :  $p_{12}$ 、 $p_{22}$ から求める

– 2つの超平面 (識別面) の内、

クラスの識別に関係する面を繋いで

区分的線形識別面 (右図の実線) を実現

じっせん



区分的線形識別面

# 区分的線形識別面の定式化

## □ 区分的線形識別面の定式化

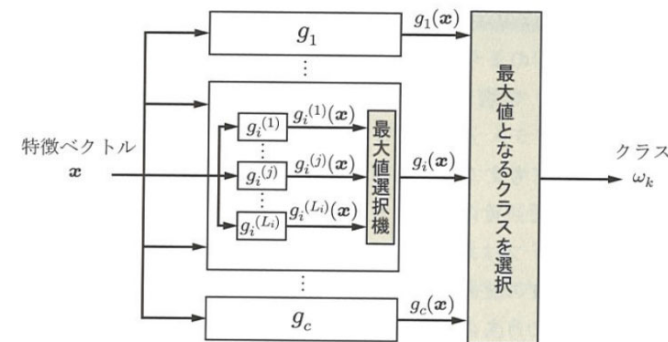
- クラス $\omega_i$ について、 $L_i$ 個の線形識別面を繋ぎ合わせれば、他のクラスと分離できると仮定

- クラス $\omega_i$ には  $L_i$  個のプロトタイプが必要

每个类有 $L_i$ 个次要判别函数，其判别函数会取其中最大值

- クラス $\omega_i$ における $L_i$ 個のプロトタイプに対応する  
副次識別関数<sup>ふくじ</sup>を  $g_i^{(l)}(x)$  ( $l = 1, \dots, L_i$ ) とする

- クラス $\omega_i$ の識別関数  $g_i(x)$  を  $L_i$  個の副次識別関数の最大値として表現
- 各クラスの識別関数  $g_1(x), \dots, g_c(x)$  の内、  
最大値をとる識別関数が  $g_k(x)$  なら、  
入力  $x$  はクラス  $\omega_k$  に識別される



区分的線形識別関数を用いた識別器

# 区分的線形識別関数の識別能力と学習

## □ 区分的線形識別関数の識別能力

- プロトタイプの数（副次識別関数の数）を増やせば、理論上は、非線形な曲面を任意の精度で近似できる
  - 学習データがどんな複雑な分布でも識別面を決められるが...

## □ 区分的線形識別関数の学習は難しい

- 副次識別関数の個数 $L_i$ （<sup>なんかい</sup>何回曲げればクラスを分離できるか？）とそれらの重みの<sup>りょうほう</sup>両方を学習しなければならない
  - これらを同時に学習することはできない
  - 副次識別関数の個数を変えると、重みの学習をやり直す<sup>なお</sup>
    - ただし、十分な個数の副次識別関数を用意しないと学習が終了しない



# 区分的線形識別関数の識別能力と学習

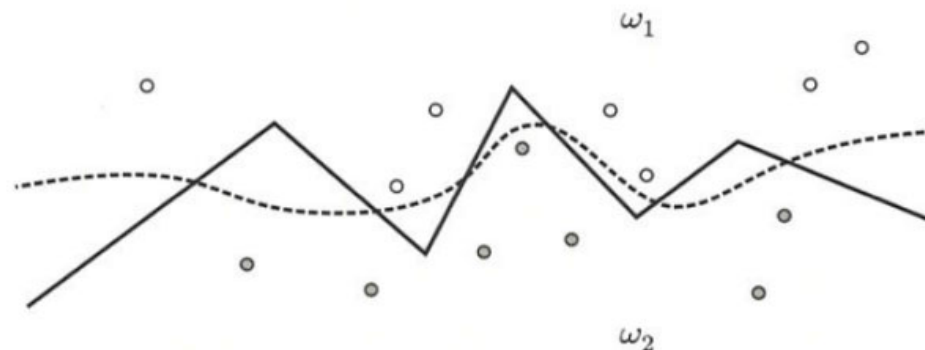
## □ 区分線形識別面の学習は難しい (つづき)

- $L_i$  をクラス  $\omega_i$  の学習データの個数とした場合、最も複雑な識別面が得られる (下図の実線)

- 「十分に多い学習データからプロトタイプを選ぶという前提で全ての学習データをプロトタイプした場合」と考えてよい

- クラスの識別面は滑らかな形の方が良さそう (下図の点線)

- それを実現するのがk-NN法

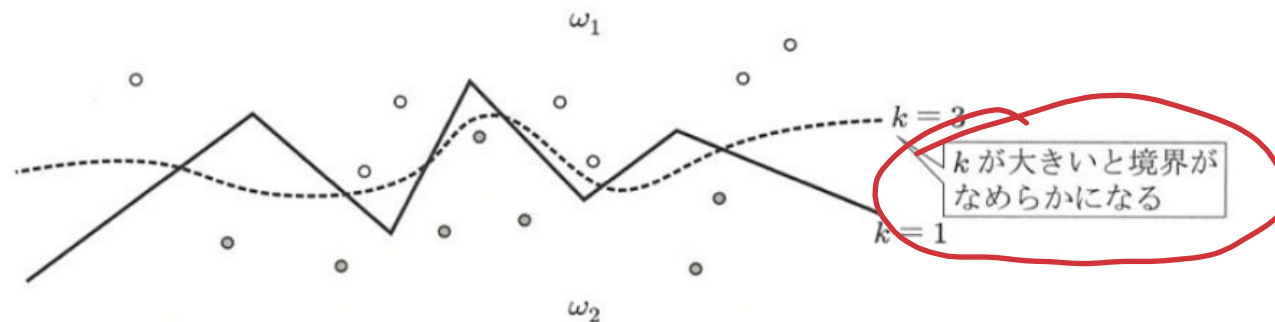


# k-NN法

□ **k-NN法** (※ NN法：最近傍決定則) 根据输入数据附近的  $k$  个训练数据来判别

■ 入力  $x$  に近い  $k$  個のデータからクラスを識別する方法

- 一般に  $k$  が大きいほど、識別面は滑らかになる傾向  $k$  越大, 判别面越光滑
- 1-NN法：入力に一番近いプロトタイプが識別結果
- 3-NN法：入力に近い3個のプロトタイプの多数決から識別結果が決まる

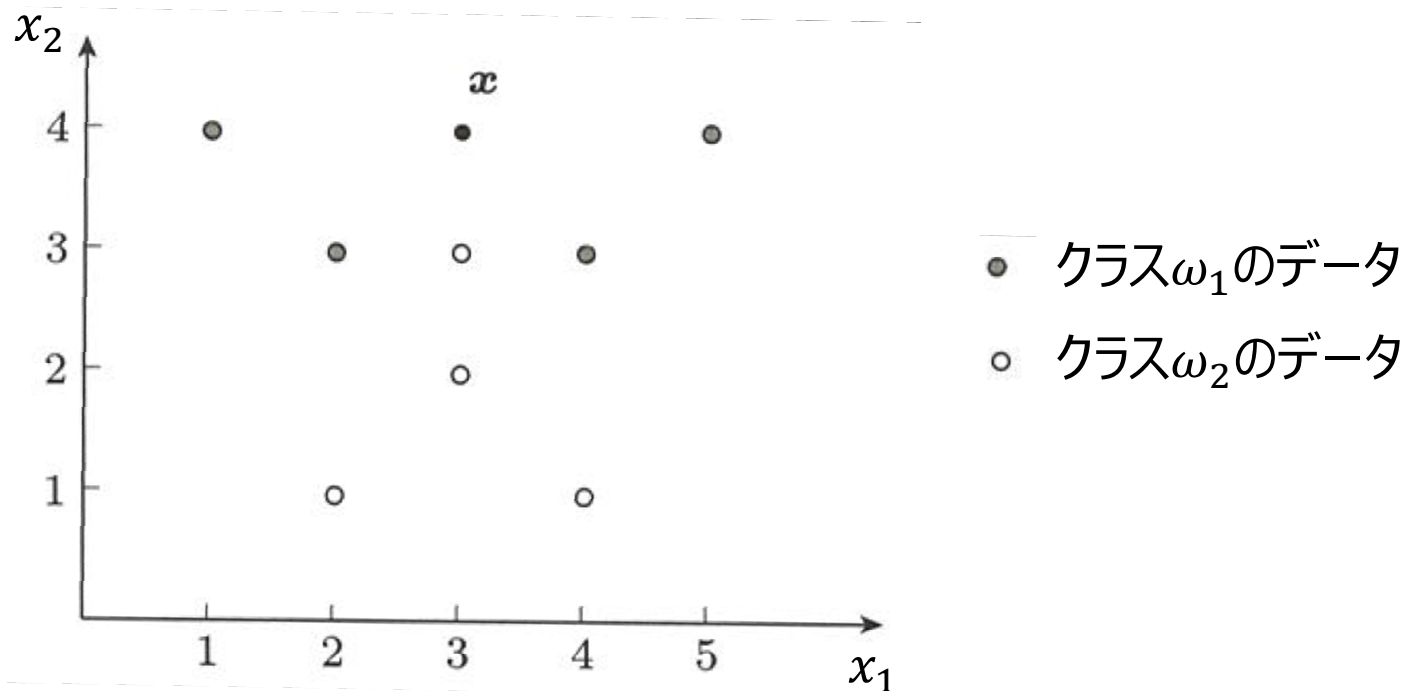


■ 識別結果の決め方は様々

- 単純な多数決、順位による重み付き多数決
- 副次識別関数の値を基準にする方法 など

# 演習問題4-1 (10分間)

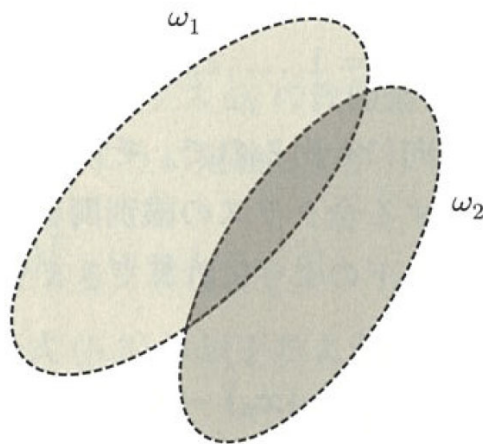
- 下図に示す学習データを用いて、  
3-NN法によって入力  $x = (3, 4)$  を識別せよ
- 識別結果の決め方は多数決とする  $\omega_1$



# 学習データを分離できない場合

- 先ほどまで、線形分離 or 区分的線形分離可能な学習データを取り扱ってきた
- 下図のように特徴空間上で複数のクラスの学習データが重なり合って分布している場合
  - 区分的線形関数でも誤識別を無くすることができない
  - 学習データに対する識別関数の誤差を定義して、その誤差を最小にする識別面を見つける方法が必要

→ その代表的な手法が最小二乗法  
当データ线性不可分时



2つのクラスの学習データの分布が重なっている  
(学習データを分離できない)

# 誤差評価に基づく学習：最小二乗法

## □ 最小二乗法 通过缩小误差的平方和来求判别函数的方法

■ 誤差の2乗和を最小にすることで識別関数を求める方法

■ 次のスライドで最小二乗法を説明するために使用する記号

- $\chi \stackrel{\text{def}}{=} \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  : 学習データの集合 (※  $A \stackrel{\text{def}}{=} B$  :  $A$ を $B$ と定義する)
- $\mathbf{x}_p$  : 集合 $\chi$ から取り出した $p$ 番目のデータ
- $c$  : クラス数
- $g_i(\mathbf{x}_p)$  : 学習データ $\mathbf{x}_p$ に対するクラス $\omega_i$ の識別関数の値
  - $g_i(\mathbf{x}_p) = \mathbf{w}_i^T \mathbf{x}_p$  (前回の講義資料より)
- $b_{ip}$  : 望ましい出力値 (教師信号)
  - $\mathbf{x}_p$ がクラス $\omega_i$ に属する場合 :  $b_{1p}, \dots, b_{cp}$  ( $b_{ip} = 1, b_{jp} = 0 (j \neq i)$ )
- $\varepsilon_{ip}$  : 識別関数の値  $g_i(\mathbf{x}_p)$ と教師信号  $b_{ip}$ の誤差
  - $\varepsilon_{ip} \stackrel{\text{def}}{=} g_i(\mathbf{x}_p) - b_{ip} (i = 1, \dots, c)$

# 誤差評価に基づく学習：最小二乗法

## □ 最小二乗法

- $J_p$  :  $\mathbf{x}_p$  に対する全クラスの識別関数の誤差の二乗和

$$J_p \stackrel{\text{def}}{=} \frac{1}{2} \sum_{i=1}^c \varepsilon_{ip}^2 = \frac{1}{2} \sum_{i=1}^c \{g_i(\mathbf{x}_p) - b_{ip}\}^2 = \frac{1}{2} \sum_{i=1}^c (\mathbf{w}_i^T \mathbf{x}_p - b_{ip})^2$$

- 上式は特定の学習データ  $\mathbf{x}_p$  に関する識別関数の誤差を評価

- $J$  : 最終的に評価すべき誤差

- 全学習データとの誤差の和

$$J \stackrel{\text{def}}{=} \sum_{p=1}^n J_p = \frac{1}{2} \sum_{p=1}^n \sum_{i=1}^c \varepsilon_{ip}^2 = \frac{1}{2} \sum_{p=1}^n \sum_{i=1}^c (\mathbf{w}_i^T \mathbf{x}_p - b_{ip})^2$$

最小二乗法は  $J$  を最小にするようにクラス  $i$  の識別関数の重み  $\mathbf{w}_i$  ( $i = 1, \dots, c$ ) を調整

# 最小二乗法の解析的な解法

□ 誤差 $J$ を最小にする重み $w_i$ の解析的な解法

1. 最小値を計算したい関数 $J$ を重み $w_i$ で偏微分
2. 1.から極小値を計算し、そのときの $w_i$ が誤差を最小とする $w_i$

■ 計算しやすくするために、以下の記号を定義

- $X = (x_1, \dots, x_n)^T$  : パターン行列
- $b_i$  : クラス $\omega_i$ の全ての教師信号を並べた $n$ 次元ベクトル  
–  $b_i \stackrel{\text{def}}{=} (b_{i1}, \dots, b_{in})^T \quad (i = 1, \dots, c)$
- 上記の記号を使って誤差 $J$ を書き換えると...

$$J = \frac{1}{2} \sum_{p=1}^n \sum_{i=1}^c (w_i^T x_p - b_{ip})^2 = \frac{1}{2} \sum_{i=1}^c \|X w_i - b_i\|^2$$

# 最小二乗法の解析的な解法

## □ 誤差 $J$ を最小にする重み $\mathbf{w}_i$ の解析的な解法 (つづき)

### 1. 誤差 $J$ を重み $\mathbf{w}_i$ で偏微分

$$\frac{\partial J}{\partial \mathbf{w}_i} = \mathbf{X}^T (\mathbf{X} \mathbf{w}_i - \mathbf{b}_i) \quad (i = 1, \dots, c)$$

### 2. 1.から極小値を計算 (上式が0となる $\mathbf{w}_i$ を計算)

$$\mathbf{X}^T (\mathbf{X} \mathbf{w}_i - \mathbf{b}_i) = 0 \quad (i = 1, \dots, c)$$

$$\Rightarrow \mathbf{X}^T \mathbf{X} \mathbf{w}_i = \mathbf{X}^T \mathbf{b}_i$$

$$\Rightarrow \mathbf{w}_i = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{b}_i$$

この計算によって誤差 $J$ を最小にする重み $\mathbf{w}_i$ を解析に求めることができた  
ただし、あくまでも全学習データに対して誤差を最小にするだけであり、  
線形分離可能な学習データでも適切な識別面を発見できない可能性があることに注意する



# 最急降下法 梯度下降法

为了使某个函数取最小值，将参数向函数值减小的方向变化

## □ 最小二乗法の解析的な解法

- データ数が多いと、ぎゃくぎょうれっえんざん逆行列演算に多くの時間が必要
- この問題を解消するのが、かいしょう最急降下法

## □ 最急降下法 (さいてきかパラメータ最適化手法の1つ)

- ある関数の値が最小値をとるように、そのパラメータを関数の値が減少する方向へ徐々に変化させる方法  
じょうじょう
  - 今回の場合、誤差  $J$  が小さくなる方向へ重み  $w$  を変化させる

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

こうしん  
更新後の  
重み

更新前の  
重み

誤差  $J$  が  
小さくなる方向

$w$  : 識別関数の重み  
 $w'$  : 更新後の重み  
 $\rho$  : 学習係数

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

# 最急降下法

## □ 最急降下法のイメージ

### ■ 例：重みを2次元 $\mathbf{w} = (w_1, w_2)$ として考える

- 誤差  $J$  は  $\mathbf{w}$  の2次式となるので、図のような2次曲面が得られる
- $\mathbf{w}_{(0)}$ ：重みの初期値

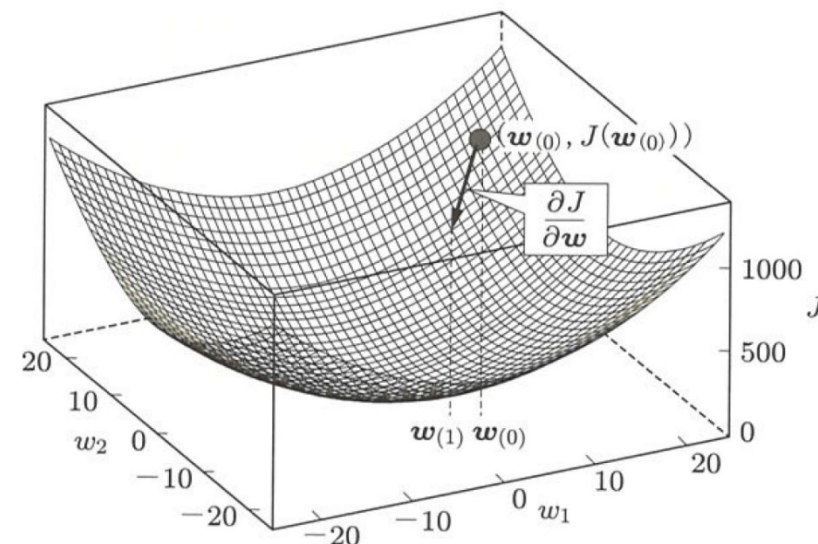
- $\frac{\partial J}{\partial \mathbf{w}} = \left( \frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2} \right)$ ：勾配ベクトル

– 勾配ベクトルの方向は  
点  $(\mathbf{w}_{(0)}, J(\mathbf{w}_{(0)}))$  にボールを  
置いたときに転がる方向と**真逆**

お                      ころ                      まぎやく

#### 【重み $\mathbf{w}$ の更新】

- 点  $(\mathbf{w}, J(\mathbf{w}))$  から  $-\rho \frac{\partial J}{\partial \mathbf{w}}$  だけ  $\mathbf{w}$  を移動させる
- 移動を繰り返すと、修正幅  $\rho \frac{\partial J}{\partial \mathbf{w}}$  が小さくなり、  
やがて  $\mathbf{w}$  は谷底付近に落ち着く



$\mathbf{w}$  を2次元として考えたときの  
最急降下法のイメージ

# Widrow-Hoffの学習規則

## □ Widrow-Hoffの学習規則 近似最速下降法

### ■ 下記の更新式を使った学習アルゴリズム

#### • 重みの修正量

– 全データに対する「学習係数・誤差・学習データの乗算結果」の合計

$$w'_i = w_i - \rho \sum_{p=1}^n (w_i^T x_p - b_{ip}) x_p$$

左の更新式は  
①～③を組み合わせ  
導出可能  
どうしゅつ

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

【①  $w \rightarrow w_i$ 】 重み $w$ をクラス $\omega_i$ の識別関数の重み $w_i$ に置換

ちかん

置換为  $w_i$

【② 誤差 $J$ を $w_i$ で偏微分】

$$J = \frac{1}{2} \sum_{p=1}^n \sum_{i=1}^c (w_i^T x_p - b_{ip})^2 \longrightarrow \frac{\partial J}{\partial w_i} = \sum_{p=1}^n (w_i^T x_p - b_{ip}) x_p$$

【③ 最急降下法の修正式】  $w' = w - \rho \frac{\partial J}{\partial w}$

# 確率的最急降下法

□ **batch** バッチ法 找出所有训练数据的误差并统一更新权重  
(※ 最急降下法はバッチ法の1つ)

- 全学習データに対して誤差を求め、一括で重みを更新 いっ かつ
- 学習データが多いと、1回の重み更新に長い時間が必要 なが
- ミニバッチ法 *mini-batch*.
  - ある程度、まとまったデータで最急降下法を実行する方法 じっ ころ

## □ 確率的最急降下法

- 個々のデータ  $x_p$  に対して、下式で重みを修正する手法
- データ  $x_p$  は学習データからランダム (確率的) に選択 随机选择一个

最急降下法

$$w'_i = w_i - \rho \sum_{p=1}^n (w_i^T x_p - b_{ip}) x_p$$

確率的<sup>な</sup>最急降下法

数据修改权重

$$w'_i = w_i - \rho (w_i^T x_p - b_{ip}) x_p$$

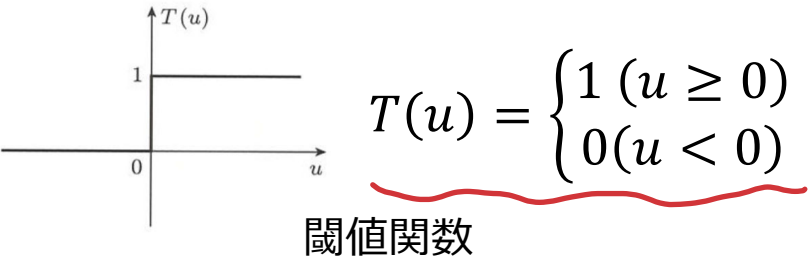
# パーセプトロンの学習規則との比較

## パーセプトロン

□ パーセプトロンの学習規則 は Widrow-Hoff 的特例

■ Widrow-Hoff の学習規則の特殊なケース

- 閾値関数を使って識別関数の出力を0または1に限定

$$\begin{cases} g_i(\mathbf{x}_p) = T(\mathbf{w}_i^T \mathbf{x}_p) = 1 \\ g_j(\mathbf{x}_p) = T(\mathbf{w}_j^T \mathbf{x}_p) = 0 \quad (j \neq i) \end{cases}$$


The graph shows a step function  $T(u)$  on a coordinate system with  $u$  on the horizontal axis and  $T(u)$  on the vertical axis. The function is 0 for  $u < 0$  and 1 for  $u \geq 0$ . The label '閾値関数' (threshold function) is written below the graph. The definition of the function is given as  $T(u) = \begin{cases} 1 & (u \geq 0) \\ 0 & (u < 0) \end{cases}$ .

- 正解のクラスは  $\mathbf{w}_i^T \mathbf{x}_p$  が正、  
それ以外のクラスは  $\mathbf{w}_i^T \mathbf{x}_p$  が負になるように  $\mathbf{w}_i$  を学習すれば良い
- このときの誤識別のパターン（重みを更新する条件）は2通りのみ

$$\begin{cases} g_i(\mathbf{x}_p) = 0, b_{ip} = 1 \\ g_j(\mathbf{x}_p) = 1, b_{jp} = 0 \quad (j \neq i) \end{cases}$$

# パーセプトロンの学習規則との比較

## □ パーセプトロンの学習規則

### ■ Widrow-Hoffの学習規則の特殊なケース（つづき）

- この設定では確率的最急降下法による  
Widrow-Hoffの学習規則は以下のように表現できる

$$\mathbf{w}'_i = \mathbf{w}_i - \rho \{g_i(\mathbf{x}_p) - b_{ip}\} \mathbf{x}_p$$

$$\Leftrightarrow \begin{cases} \mathbf{w}'_i = \mathbf{w}_i + \rho \mathbf{x}_p & (g_i(\mathbf{x}_p) = 0, b_{ip} = 1 \text{ のとき}) \\ \mathbf{w}'_i = \mathbf{w}_i - \rho \mathbf{x}_p & (g_i(\mathbf{x}_p) = 1, b_{ip} = 0 \text{ のとき}) \end{cases}$$

Widrow-Hoffの学習規則

パーセプトロンの学習規則の

1. 識別関数の値と教師信号との二乗誤差 重みの更新式と同じ の総和を最小化する
2. 線形分離不可能の場合でも収束する
3. 線形分離可能の場合でも、全ての学習パターンが正しく識別される重みは得られるとは限らない

# 演習問題4-2 (10分間)

□ 次の文は、それぞれ「パーセプトロンの学習規則」と「Widrow-Hoffの学習規則」のどちらについて説明しているか？

1,2: Widrow-Hoffの学習規則

3: パーセプトロンの学習規則

1. 識別関数の値と教師信号の二乗誤差の総和を最小化する

WH

2. 線形分離可能な場合でも、全ての学習パターンが正しく識別される重みが得られるとは限らない

WH

3. 線形分離不可能の場合は、学習は収束しない

perceptron

パーセプトロンの学習規則

1. 2クラスの場合、識別関の正負と教師信号の正負が一致するまで修正を繰り返す
2. 線形分離可能であれば、学習は収束し、全ての学習パターンが正しく識別される重みベクトルが得られる
3. 線形分離不可能の場合は、学習は収束しない