RITSUMEIKAN

機械学習 第３回 識別（１）

立命館大学 情報理工学部

村上 陽平

Beyond Borders

1

講義スケジュール

- 担当教員１：村上、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	識別（２）
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	強化学習
15	半教師あり学習

- 担当教員２：叶昕辰先生（第16回の講義を担当）

2

今回の講義内容

- 取り扱う問題の定義
- 決定木(けっていぎ)の学習
  - ID3アルゴリズム
- 最近傍決定則(さいきんぼうけっていそく)
  - 特徴空間
  - 特徴ベクトル
  - プロトタイプ
  - パーセプトロンの学習規則
- 演習問題

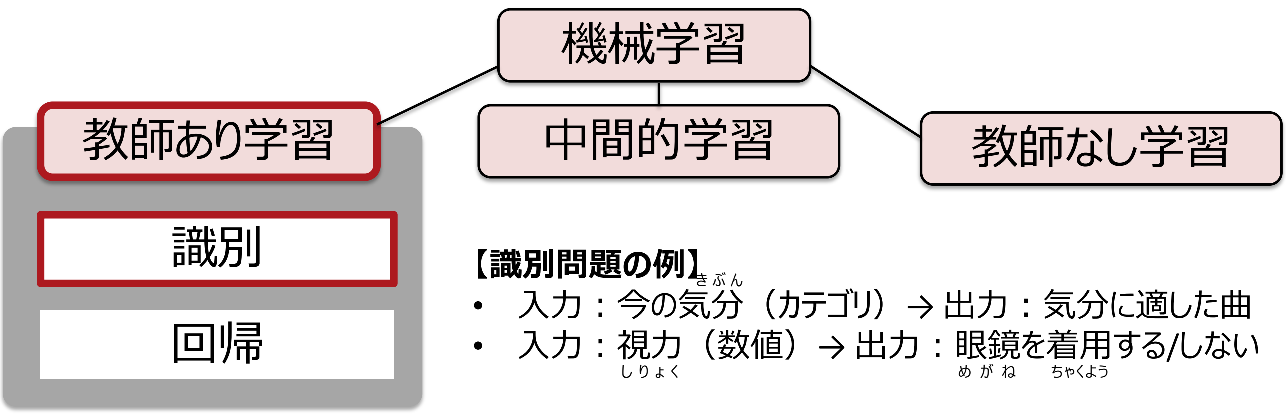
取り扱う問題の定義：教師あり・識別問題

- 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（カテゴリデータ/数値データ） ← $\{\boldsymbol{x}_i, y_i\}, \quad i = 1,2,\ldots,N$ → 学習データの総数

カテゴリ形式の正解情報 →「クラス」と呼ぶ



【識別問題の例】

- 入力：今の気分(きぶん)（カテゴリ）→ 出力：気分に適した曲
- 入力：視力(しりょく)（数値）→ 出力：眼鏡(めがね)を着用(ちゃくよう)する/しない

決定木

□ 概念学習

- 個々の事例から、あるクラスについて共通点をみつける学習手法
- その学習手法の代表例が**決定木**

□ 決定木

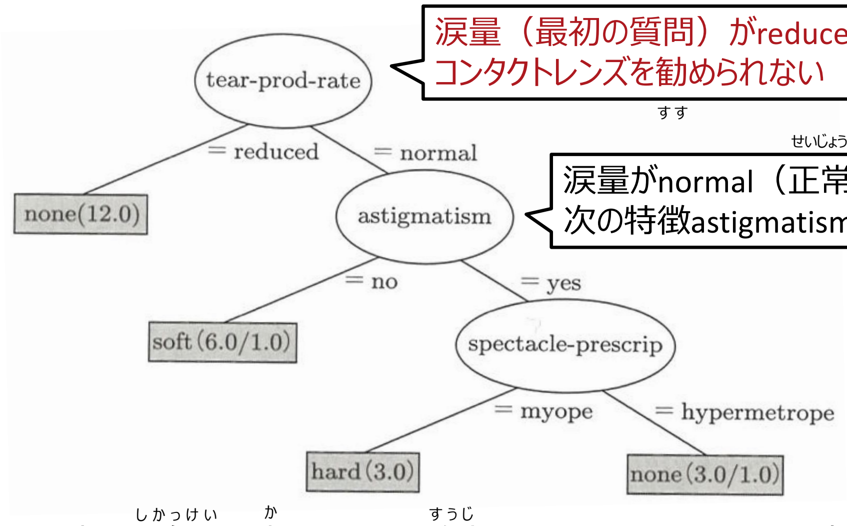
- データを分類する質問をノード(節)とし、
  分類結果をリーフ(葉)とする木構造の概念表現
  - 根から分類結果が正である葉に至る節の分岐の値を **AND条件** で結合
  - 全ての正の葉に関して、そのように得られた論理式を **OR条件** で結合
- 木構造は、人間の目から見て学習結果がわかりやすい

決定木の学習：決定木

□ 決定木の例（コンタクトレンズデータの場合）

- 得られる情報が多い質問（特徴）を木の上方のノードに
  配置するように構成

涙量（最初の質問）がreduced（減少）であると、
コンタクトレンズを勧められない



涙量がnormal（正常）であれば、
次の特徴astigmatism（乱視）を調べる

← 以下の3種類に分類
- ハードコンタクトレンズ
- ソフトコンタクトレンズ
- コンタクトレンズを勧めない

※ リーフを表す四角形に書かれている数字：そのリーフに分類された事例数
※ スラッシュと数字が続いている場合は、リーフに書かれたクラスに当てはまらない事例数

決定木の学習：ID3アルゴリズム

- 決定木を作成する基本的な手順が、
  ID3（Iterative Dichotomiser 3）アルゴリズム

1. 全ての特徴の中から、**得られる情報が最も多そうな特徴**を最初の質問として選択
   - 「得られる情報が最も多そうな特徴」
     ＝「特徴集合の中で最も分類能力の高い特徴」
   - 言い換えると、乱雑さが少なくなるように分類できる特徴

2. 1. の質問によって、学習データをいくつかの部分集合に分割
   - **同クラスのデータ**からなる部分集合は、それ以上の質問を続ける必要なし
   - **異なるクラスのデータ**が混在する部分集合は、
     得られる情報が多そうな質問を選んでデータをさらに分割

決定木の学習：ID3アルゴリズム

- エントロピーを用いた
  データ集合の乱雑さ（分類能力の高さ）の評価

$$E(D) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

※ $E(D)$：エントロピー（データ集合Dの乱雑さ）、P_+：正例の割合、P_-：負例の割合

- エントロピー$E(D)$は、
  $\boldsymbol{P_+ = 1}$ または $\boldsymbol{P_- = 1}$のとき**最小値0**、
  $\boldsymbol{P_+ = P_- = 0.5}$のとき**最大値1**となる

- エントロピーの値が**小さい**ほど、集合が**乱雑ではない**
  （同じクラスのものが大半を占めている）ということを示す

決定木の学習：ID3アルゴリズム

□ 情報獲得量

- 特徴値に基づく分類による**エントロピーの減少量**
- その特徴を選択することによって得られる**情報の大きさ**
- 情報獲得量が大きい特徴を選んでデータを分割する

情報獲得量

$$\mathrm{Gain}(D,a) \equiv E(D) - \sum_{v \in \mathrm{Values}(a)} \frac{|D_v|}{|D|} E(D_v)$$

分類前のエントロピー（$E(D)$）

分類後のエントロピー（$\sum_{v \in \mathrm{Values}(a)} \frac{|D_v|}{|D|} E(D_v)$）

- 分類後の集合数＝特徴値の種類数
- 集合の要素数の割合で重み付け

※ $E(\cdot)$：エントロピー、$|\cdot|$：集合「$\cdot$」の要素数、$\mathrm{Values}(a)$：特徴aのとりうる値
D_v：値$v \in \mathrm{Values}(a)$をとる学習データの集合

決定木：例題（補足資料）

□ 以下は、ある人がゴルフに参加するかを気象（きしょう）条件を特徴として決定するデータである。

	天候（てんこう）	気温	湿度（しつど）	風（かぜ）	ゴルフ
1	晴	高温	多湿	なし	不参加
2	晴	高温	多湿	あり	不参加
3	曇	高温	多湿	なし	参加
4	雨	適温	多湿	なし	参加
5	雨	低温	標準	なし	参加
6	雨	低温	標準	あり	不参加
7	曇	低温	標準	あり	参加
8	晴	適温	多湿	なし	不参加
9	晴	低温	標準	なし	参加
10	雨	適温	標準	なし	参加
11	晴	適温	標準	あり	参加
12	曇	適温	多湿	あり	参加
13	曇	高温	標準	なし	参加
14	雨	適温	多湿	あり	不参加

- このデータのエントロピー$E(D)$を求めたい。
- 各特徴量の情報獲得量を計算して、最初のデータ分割に用いる特徴を見つけたい。

決定木：例題（補足資料）

エントロピー$E(D)$

- 全14事例中、参加が9事例、不参加が5事例なので

$$E(D) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} \fallingdotseq \mathbf{0.94}$$

情報獲得量

$$\mathrm{Gain}(D,\text{天候}) = E(D) - \frac{5}{14}E(\overset{\text{はれ}}{\text{晴}}) - \frac{4}{14}E(\overset{\text{くもり}}{\text{曇}}) - \frac{5}{14}E(\overset{\text{あめ}}{\text{雨}})$$

$$\fallingdotseq 0.94 - 0.357\times\left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)$$

$$-0.286\times\left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right) - 0.357\times\left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

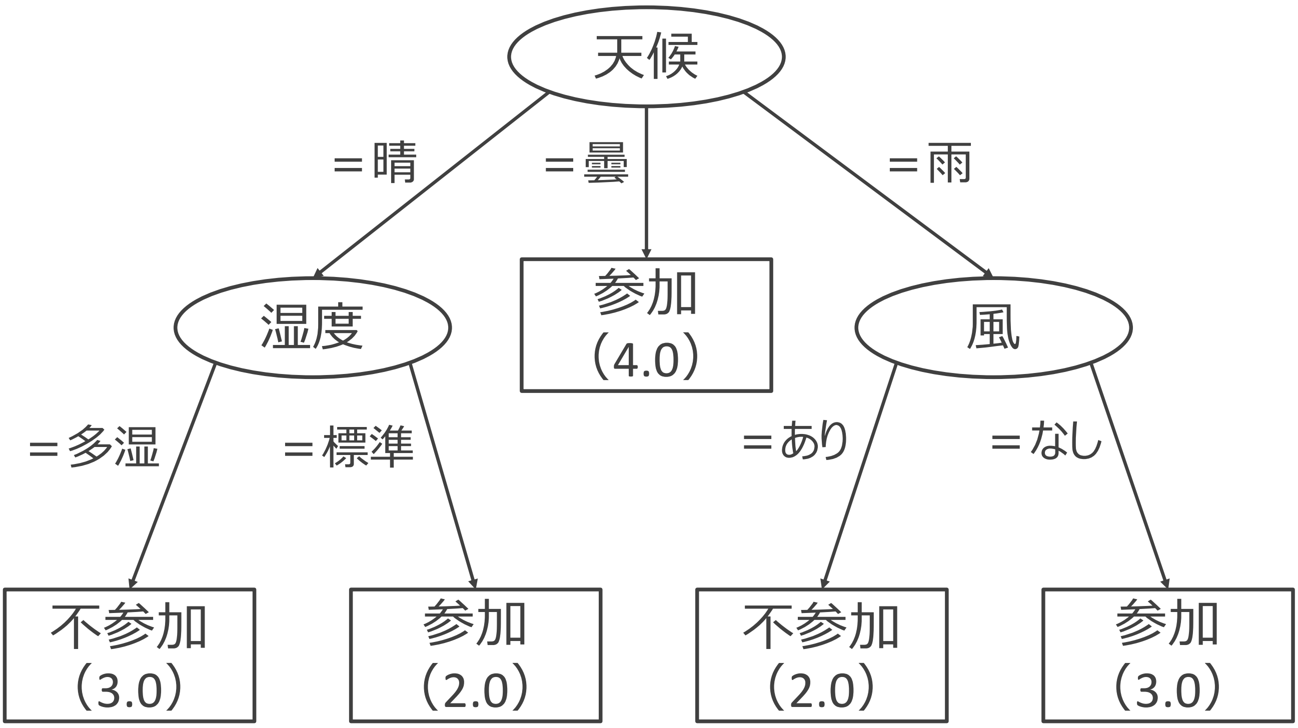
$$\fallingdotseq 0.247$$

最も情報獲得量の多い特徴は「天候」
この「天候」を特徴（質問）として
データを分割すれば良い

同じ手順で、気温、湿度、風の情報獲得量を求めると、

$\mathrm{Gain}(D,\text{気温}) \fallingdotseq 0.029$、$\mathrm{Gain}(D,\text{湿度}) \fallingdotseq 0.151$、$\mathrm{Gain}(D,\text{風}) \fallingdotseq 0.048$

決定木：例題（補足資料）



決定木の学習：ID3アルゴリズム

◻ 過学習

- ■ モデルが学習データに特化しすぎたために、未知データに対して性能が下がる現象

◻ ID3アルゴリズムの過学習

- ■ 一般的に小さい木を実現させることは難しい
- ■ 全ての事例のエラーが無くなるように決定木を作成すると、その木が成長しすぎて、学習データに適応しすぎた過学習になりがち。
- ■ 「適当なところで決定木の成長を止める方法」や「完全に学習させたあと、**枝刈り**する方法」で過学習に対処する必要がある。

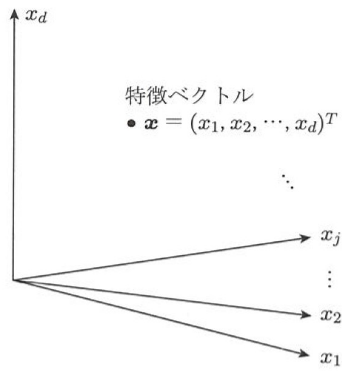
最近傍決定則：特徴空間・特徴ベクトル

◻ 人間が顔画像から人物を識別することを考える

- ■ 目だけを切り出した情報を用いて識別することは困難
- ■ 髪型・顔の輪郭・肌の色などの**複数の特徴**も使うはず
  - • 機械学習では、このような複数の特徴をどのように表現するのか？

◻ 特徴空間・特徴ベクトル

- ■ 特徴ベクトル $\boldsymbol{x} = (x_1, x_2, \dots, x_d)^T$
  - • d個の特徴の並びを表現した **d次元ベクトル**
- ■ このd次元空間を 特徴空間 と呼ぶ
- ■ 特徴ベクトルは特徴空間上の1点を表す



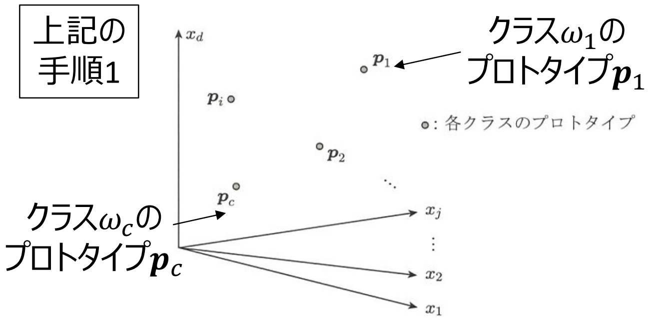
特徴空間と特徴ベクトル

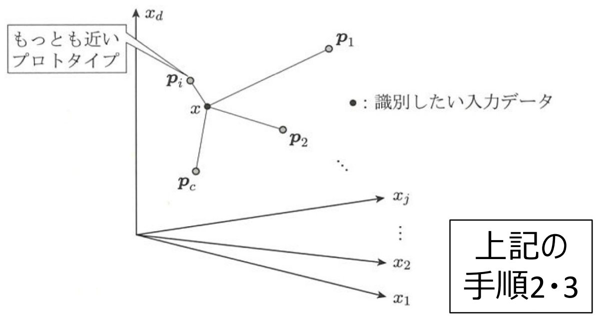
最近傍決定則

最近傍決定則（nearest neighbor法、NN法）

- 入力特徴ベクトル$\boldsymbol{x}$が属するクラスを判定する簡単な方法

1. 各クラスにつき1つのプロトタイプ$\boldsymbol{p}_i$（代表となるベクトル）を用意
   - プロトタイプ： $\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_c$ （※ $\boldsymbol{c}$ **：クラスの総数**）
2. 入力特徴ベクトルと各クラスのプロトタイプの **距離を比較**
3. **最も近い**プロトタイプ$\boldsymbol{p}_i$の属するクラスω_iを識別結果とする





最近傍決定則の定式化

以下のようにクラス、特徴ベクトル、プロトタイプを定義

- $\boldsymbol{c}$**個** のクラス：$\omega_1, \omega_2, \dots, \omega_c$
- $\boldsymbol{d}$**次元** の入力特徴ベクトル：$\boldsymbol{x} = (x_1, x_2, \dots, x_d)^T$
- **クラス**$\boldsymbol{\omega_i}$ のプロトタイプ：$\boldsymbol{p}_i = (p_{i1}, p_{i2}, \dots, p_{id})^T$

入力ベクトル$\boldsymbol{x}$とプロトタイプ$\boldsymbol{p}_i$のユークリッド距離

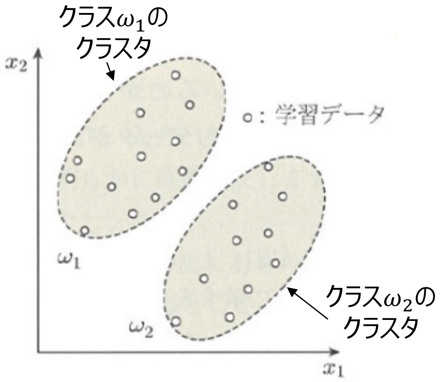
- $D(\boldsymbol{x}, \boldsymbol{p}_i) = \sqrt{(x_1 - p_{i1})^2 + (x_2 - p_{i2})^2 + \cdots + (x_d - p_{id})^2}$

最も近いプロトタイプp_kが属するクラスω_kの求め方

- $\boldsymbol{k = \arg\min_{i} D(x, p_i)}$ $(i = 1, \dots, c)$

プロトタイプと識別面の関係

- 同じクラスに属する学習データは、特徴空間上で**クラスタ（かたまり）**になっていることが多い
  - 例：2次元の特徴空間における2クラスの識別問題



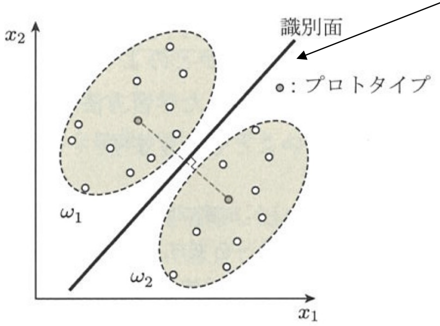
2つのクラスを分離（ぶんり）する**境界線（識別面）**をどのようにして決めるのか？

プロトタイプと識別面の関係

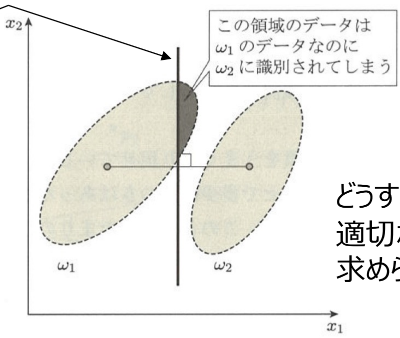
- 最近傍決定則における識別面
  - **特徴ベクトル$\boldsymbol{x}$がプロトタイプから等距離にある線**
    - つまり、**垂直二等分線**（すいちょくにとうぶんせん）（※ d次元では垂直二等分$d-1$次元超平面（ちょうへいめん））
  - プロトタイプの位置によって、正しい/間違（まちが）った識別面が設定されることに注意（ちゅうい）

プロトタイプを結んだ線に対する垂直二等分線

ちなみに…
左図のように
線形の識別面でクラスを
分離できる場合を
線形分離可能と呼ぶ



正しく設定された識別面



間違った識別面

どうすれば、
適切な識別面を
求められるのか？

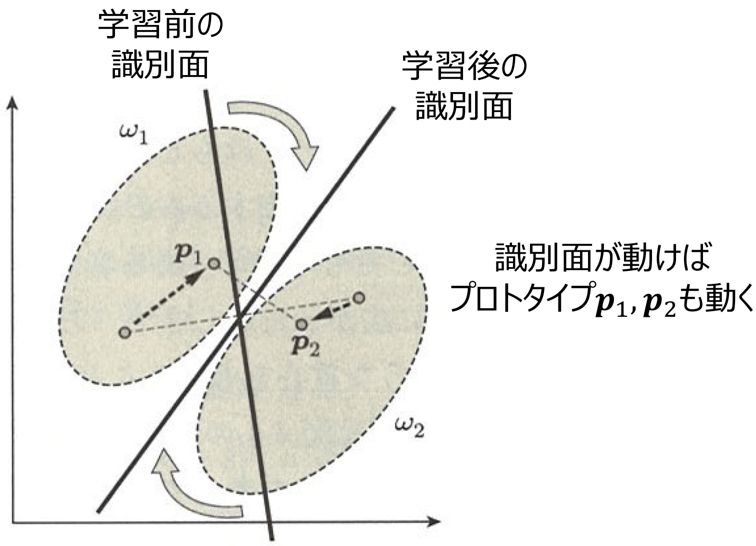
演習問題3-1（10分間）

- 以下の2つのクラスのプロトタイプが与えられたとき、入力 $\boldsymbol{x} = (1,6)$ は最近傍決定則を用いるとどちらのクラスに識別されるか？
  - クラスω_1のプロトタイプ：$\boldsymbol{p}_1 = (2,8)$
  - クラスω_2のプロトタイプ：$\boldsymbol{p}_2 = (8,4)$

- このときの識別面の式を求めよ。

パーセプトロンの学習規則

- パーセプトロンの学習規則
  - 学習データから識別面を求めるアルゴリズムの1つ
  - 誤識別が無くなるように識別面を動かす（学習する）



パーセプトロンの学習規則

- 識別関数の設定
  - 入力ベクトル$\boldsymbol{x}$とクラスω_iのプロトタイプ$\boldsymbol{p}_i$の距離

$$D(\boldsymbol{x},\boldsymbol{p}_i)=\sqrt{(x_1-p_{i1})^2+(x_2-p_{i2})^2+\cdots+(x_d-p_{id})^2}$$

$$=\|\boldsymbol{x}-\boldsymbol{p}_i\|$$ ← これを**最小**にする$\boldsymbol{p}_i$を求めるのが目的

う へ ん　　　　　　　　　　　　　　じょう

  - 右辺を最小にする$\boldsymbol{p}_i$は、右辺を2乗しても変わらないので

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

$$=\|\boldsymbol{x}\|^2-2\left(\boldsymbol{p}_i^T\boldsymbol{x}-\frac{1}{2}\|\boldsymbol{p}_i\|^2\right)$$

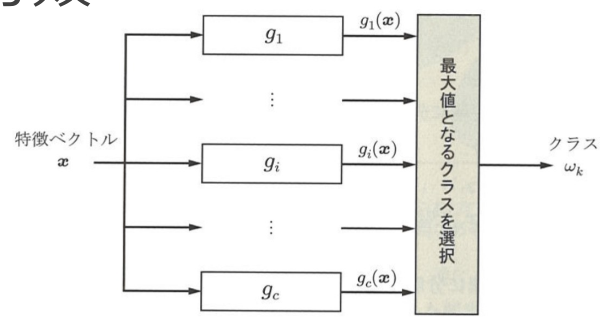
$\boldsymbol{p}_i$に関係なく
常に同じ値なので無視する
つ ね　　　　　　　　む　し

これを**識別関数** $g_i(\boldsymbol{x})$とする
識別関数の値が**大きい**と、距離が**小さく**なる

パーセプトロンの学習規則

- 入力ベクトル$\boldsymbol{x}$との距離が最小のプロトタイプ$\boldsymbol{p}_i$
  - 識別関数 $g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_c(\boldsymbol{x})$ の中から
    最大の$g_i(\boldsymbol{x})$を求めれば良い

- 最近傍決定則によるクラス分類
  - 特徴ベクトル$\boldsymbol{x}$を入力し、
    各クラスω_iに対応する
    識別関数$g_i(\boldsymbol{x})$を計算して、
    最大値をとるクラスω_kを選択



最近傍決定則によるクラス分類

パーセプトロンの学習規則

□ 識別関数 $g_i(\boldsymbol{x})$

- ■ 特徴ベクトルの各次元の値に対して係数(けいすう)をかけたものの和(第1項(こう))を求め、それに定数(ていすう)(第2項)を足した形

$$g_i(\boldsymbol{x}) = \boldsymbol{p}_i^T \boldsymbol{x} - \frac{1}{2}\|\boldsymbol{p}_i\|^2$$

- ■ $\boldsymbol{x}$の係数と定数項を以下のように置(お)き換(か)える
  - $\boldsymbol{x}$の係数：$p_{ij} = w_{ij}\ (j = 1, \dots, d)$、定数項：$-\frac{1}{2}\|\boldsymbol{p}_i\|^2 = w_{i0}$

$$g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij} x_j$$

  - 「最近傍決定則におけるプロトタイプの位置を調整する問題」が「識別関数の**係数を調整する問題**」に置き換わる

パーセプトロンの学習規則

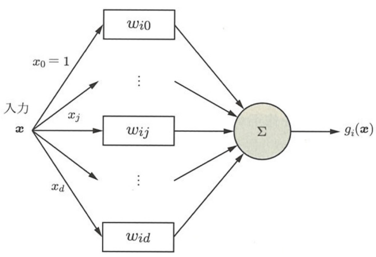
□ 識別関数 $g_i(\boldsymbol{x})$ （つづき）

- ■ 表記を簡単にするために、
  $x_0 = 1$とした $\boldsymbol{d+1}$**次元**の特徴ベクトルを$\boldsymbol{x}$とすると

$$g_i(\boldsymbol{x}) = w_{i0} + \sum_{j=1}^{d} w_{ij} x_j = \sum_{j=0}^{d} w_{ij} x_j = (w_{i0}, \dots, w_{id})(x_0, \dots, x_d)^T = \boldsymbol{w}_i^T \boldsymbol{x}$$

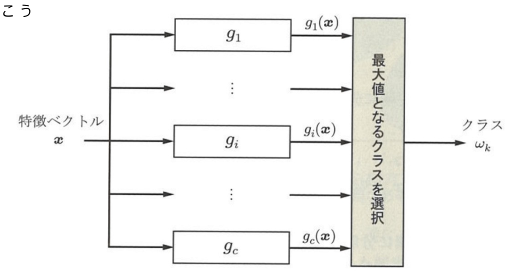
※ **重みベクトル**： $\boldsymbol{w}_i = (w_{i0}, \dots, w_{id})^T$

識別関数は入力の**重み付(つ)き線形和**



識別関数$g_i(\boldsymbol{x})$の計算メカニズム

「重み付き線形和」と「最大値選択」を組み合わせた計算機構(きこう)を**パーセプトロン**と呼ぶ



【再掲】最近傍決定則によるクラス分類

パーセプトロンの学習規則

- 識別関数の学習
  - 線形分離可能な学習データに対して
    識別関数の**重み**w_{ij}を正しく決定できれば
    最近傍決定則による誤(あやま)りのない識別ができる。
  - χ_i $(i = 1, \dots, c)$ をクラスω_iに属するデータ集合としたとき
    χ_iに属する全ての$\boldsymbol{x}$に対して、以下の式が成り立つように
    重みを調整すれば良い。

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

クラスω_iに属する**全てのデータ$\boldsymbol{x}$**に対して
クラスω_iの識別関数の値 $\boldsymbol{g_i}(\boldsymbol{x})$が、他(ほか)の識別関数の値 $\boldsymbol{g_j}(\boldsymbol{x})$を**上回(うわまわ)れば良い**

パーセプトロンの学習規則

- ここから、2クラスの識別関数の学習を考える
  - 2クラスの場合、入力に対する2つの識別関数の値の大小(だいしょう)を比較した結果から識別結果が得られる
  - 2つの識別関数の差を、新(あら)たな識別関数$g(\boldsymbol{x})$とする

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

$$= \boldsymbol{w}_1^T\boldsymbol{x} - \boldsymbol{w}_2^T\boldsymbol{x} = \left(\boldsymbol{w}_1^T - \boldsymbol{w}_2^T\right)\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{x}$$

識別関数$g(\boldsymbol{x})$の値の意味(いみ)
$$\begin{cases} g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} > \boldsymbol{0} \rightarrow \boldsymbol{x} \text{ は}\textbf{クラス}\boldsymbol{\omega_1}\textbf{に属する} \\ g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} = 0 \rightarrow \boldsymbol{x} \text{ は}\textbf{識別面上}\text{にある} \\ g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} < 0 \rightarrow \boldsymbol{x} \text{ は}\textbf{クラス}\boldsymbol{\omega_2}\textbf{に属する} \end{cases}$$

パーセプトロンの学習規則

□ 以下を満(み)たすように重み$\boldsymbol{w}$を調整する

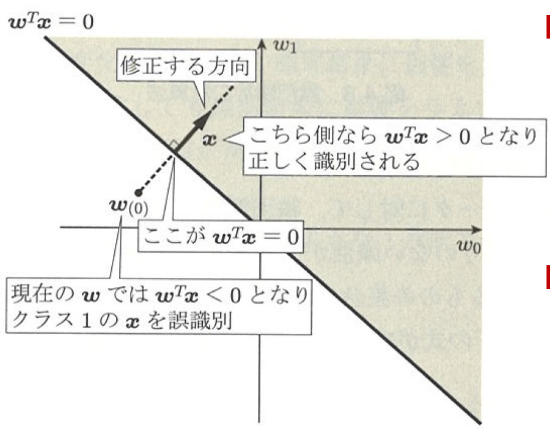
$$\begin{cases} g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} > 0 \ (\boldsymbol{x} \in \chi_1) \\ g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} < 0 \ (\boldsymbol{x} \in \chi_2) \end{cases}$$

- 適当な重み（初期値(しょきち)）からはじめて、
  ある学習データに対して上式と異なる結果が出たとき
  重みを修正するアルゴリズムを考える

- ここまで特徴ベクトル$\boldsymbol{x}$が存在する特徴空間を考えてきたが
  ここからは重みベクトル$\boldsymbol{w}$を要素とする重み空間を考える
  - 特徴空間がd次元ならば、**重み空間**は$\boldsymbol{d+1}$**次元**となる

パーセプトロンの学習規則

□ 重み$\boldsymbol{w}$の修正方針(しゅうせいほうしん)



重み空間での重みの修正

- $\boldsymbol{w}_{(0)}$を重みの初期値とする

- 重み空間で超平面$\boldsymbol{w}^T\boldsymbol{x} = 0$を考える
  - 重み空間では、$\boldsymbol{x}$**が定数**、$\boldsymbol{w}$**が変数(へんすう)**である
  - 個々の学習データ$\boldsymbol{x}$に
    1つの超平面$\boldsymbol{w}^T\boldsymbol{x} = 0$が対応する
  - 超平面を境(さかい)に$\boldsymbol{w}^T\boldsymbol{x}$の正負が反転する

- 誤識別が生(しょう)じたときのみ重みを修正
  - 例：クラスω_1のデ(こ)ータ$\boldsymbol{x}$に対して $\boldsymbol{g(x) = w_{(0)}^T x < 0}$
  - $\boldsymbol{w}_{(0)}$が超平面を超えるように重みを移動させる
  - **超平面に垂直な方向へ移動させる**のが最も近道(ちかみち)

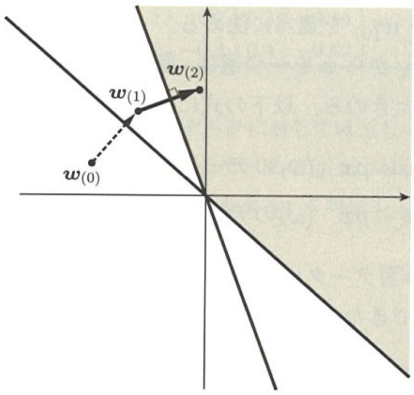
↓

> **$\boldsymbol{w}$を$\boldsymbol{x}$の方向へ移動させれば良い**
> ※ 理由(りゆう)：$\boldsymbol{x}$は超平面$\boldsymbol{w}^T\boldsymbol{x} = 0$の法線(ほうせん)ベクトルだから

パーセプトロンの学習規則

- 重みwの修正方針（つづき）
  - 別の学習データに対しても、その時点(じてん)での重みで識別する
    - 誤識別が起これば、重みを修正するという手順を繰り返す
    - データが変(か)われば、前のデータとは異なる超平面となる

データが変わった後の超平面



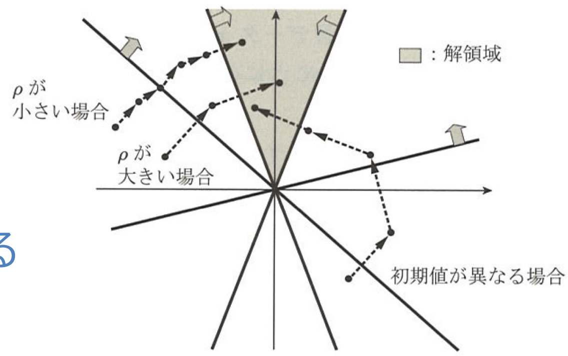
データが変わる前の超平面

別の学習データに対する重みの修正

パーセプトロンの学習規則

- 解領域(かいりょういき)
  - 重み空間上で、全ての学習データに対して正しい識別結果を出力する領域
  - 今回説明(せつめい)した手順は、**適当な初期値**から出発して、**修正を繰り返し**ながら解領域を探すプロセスである

- 学習係数 ρ
  - 重みの**修正幅(はば)**
    - 小さ過(す)ぎると収束(しゅうそく)に時間がかかる
    - 大き過ぎると解領域付近(ふきん)で振動して収束しない可能性あり



解領域への重みの修正プロセス

パーセプトロンの学習規則

- パーセプトロンの学習規則のアルゴリズム
  1. 重みの初期値 $\boldsymbol{w} = \boldsymbol{w}_{(0)}$ を適当に決定する
  2. 学習データの集合χから $\boldsymbol{x}$ を1つ選び、
     識別関数の値 $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$ を計算する
  3. 誤識別が起こったときのみ、以下の式に従い $\boldsymbol{w}$ を修正する
     1. $\boldsymbol{w}' = \boldsymbol{w} + \rho\boldsymbol{x}$ （ω_1のデータをω_2と誤ったとき）
     2. $\boldsymbol{w}' = \boldsymbol{w} - \rho\boldsymbol{x}$ （ω_2のデータをω_1と誤ったとき）
  4. 2. と 3. をすべての学習データについて繰り返す
  5. 全て正しく識別できたら終了。そうでなければ、2. へ戻(もど)る
- 学習データが線形分離可能ならば、
  この学習規則は有限回(ゆうげんかい)の繰り返しで終了する。

演習問題3-2（10分間）

- 前スライドのパーセプトロンの学習規則を用いて
  以下の表のデータから識別関数$g(\boldsymbol{x})$を求めよ。
  - 重みの初期値：$w_0 = 0.2, w_1 = 0.3$
  - 学習係数：$\rho = 0.5$

クラス	x
ω_1	1.0
ω_1	0.5
ω_2	-0.2
ω_2	-1.3