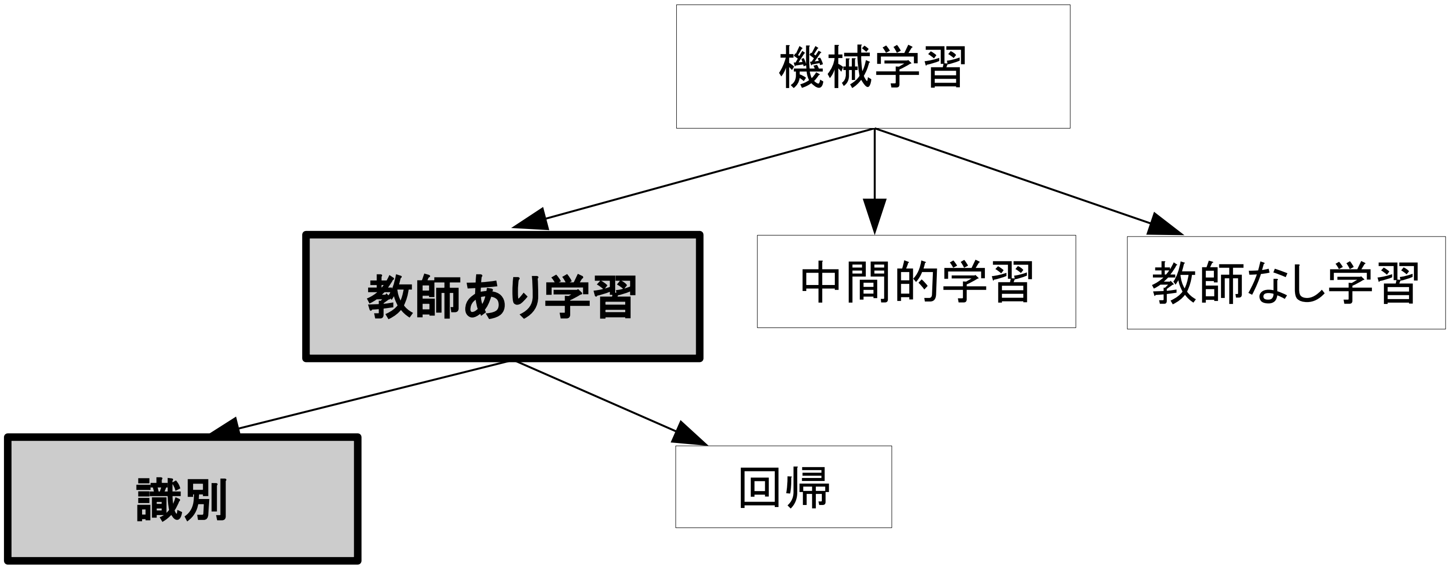
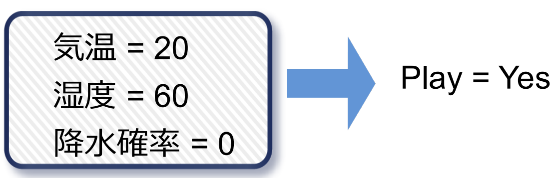
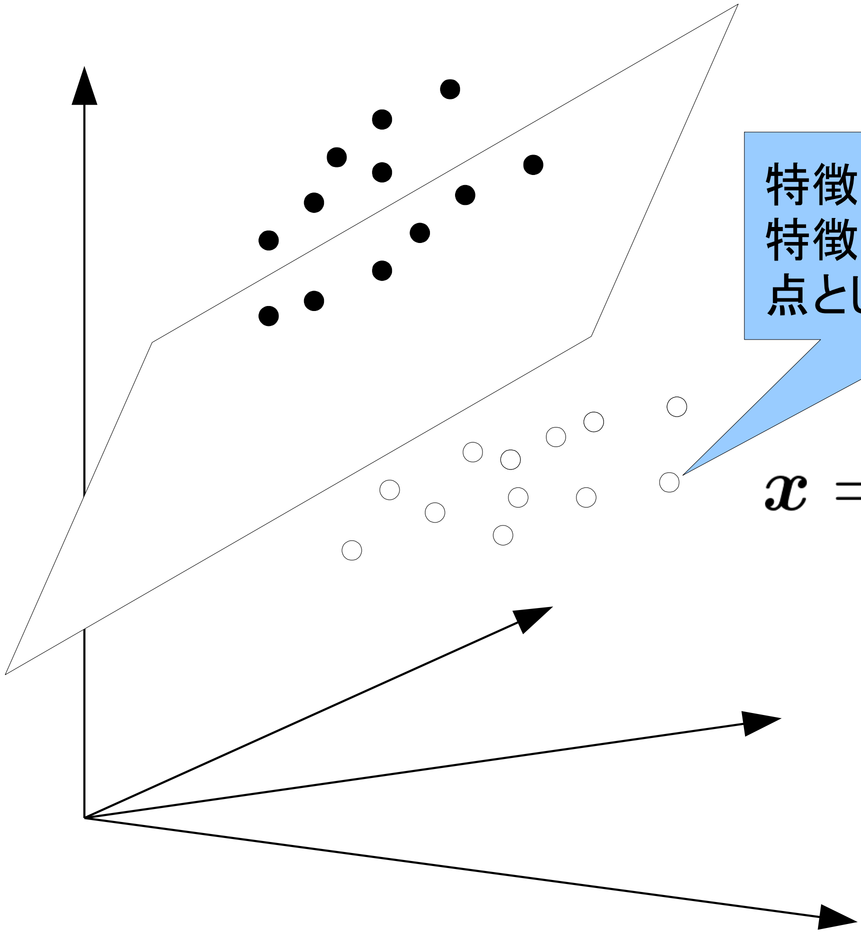
5. 識別　ー生成モデルと識別モデルー



- ラベル特徴
- 数値特徴



5.1 数値特徴に対する「教師あり・識別」問題の定義



特徴ベクトルは、
特徴空間上の
点として表現される

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

比較的クラスが分離している

→ 小数のパラメータで各クラスを表現可能

⇒ **統計的手法**

クラス境界が複雑

非線形識別面 ⇒ **ニューラルネット**

高次元へマッピング⇒ **SVM**

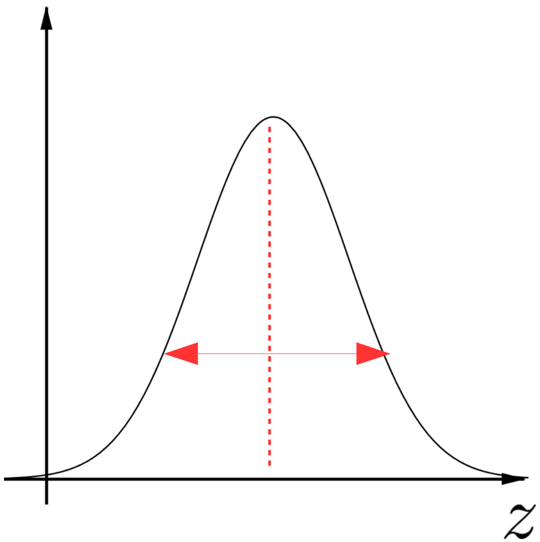
5.2 数値特徴に対するベイズ識別

5.2.1 数値特徴に対するナイーブベイズ識別

$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} p(x_j|\omega_i)$$

- 確率密度関数 $p(x_j|\omega_i)$ の推定
  - 正規分布を仮定

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(z-\mu)^2}{2\sigma^2})$$



  - 平均 μ と分散 σ を最尤推定
    - それぞれ、学習データの平均と分散になる
    - 計算法 https://mathtrain.jp/mle

ベイズ推定法

- 最尤推定の考え方
  - クラス分布関数 q のパラメータ $\boldsymbol{\theta}$ を尤度を最大にするひとつに定める

$$\mathcal{L}(D) = P(D|\boldsymbol{\theta}) = \prod_{i=1}^{N} q(\boldsymbol{x}_i|\boldsymbol{\theta})$$

- ベイズ推定法の考え方
  - パラメータ $\boldsymbol{\theta}$ を確率変数とみなす
  - モデル q をパラメータの事後確率に関して平均することでクラス分布関数を推定

ベイズ推定法

- 「パラメータ $\boldsymbol{\theta}$ を確率変数とみなす」とは

「サイコロを 3 回振って 3 回とも 1 が出た」

最尤推定法　θ は決定論的な変数

1 の確率： 1

他の確率： 0

ベイズ推定法　θ は確率変数

最も確率の高い θ の値

すべての確率： 1/6

観測

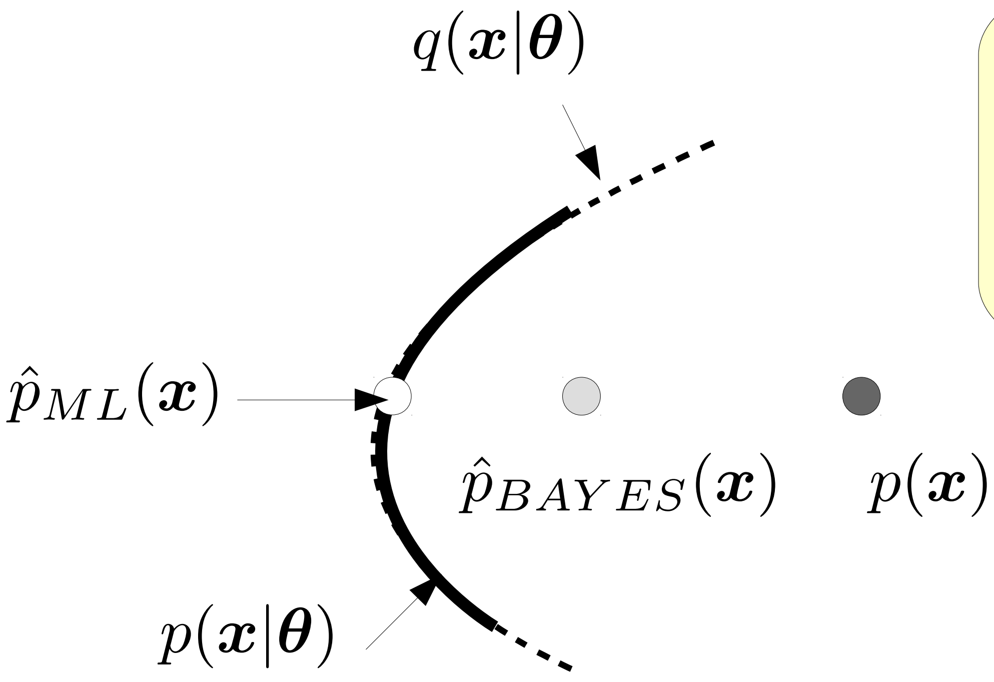
1 の確率： 0.44

他の確率： 0.11

ベイズ推定法

- 「モデル q をパラメータの事後確率に関して平均する」とは

$$\hat{p}_{BAYES}(\boldsymbol{x}) = \int q(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|D)d\boldsymbol{\theta}$$



・ q の分布の形を仮定することで、最尤推定は真の分布とずれている
・ベイズ推定は平均を取ることで、モデルの外に出ることができる

ベイズ推定法

- パラメータ $\boldsymbol{\theta}$ の事後確率

$$p(\boldsymbol{\theta}|D) = \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)} = \frac{\prod_{i=1}^{n} q(\boldsymbol{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int \prod_{i=1}^{n} q(\boldsymbol{x}_i|\boldsymbol{\theta}')p(\boldsymbol{\theta}')d\boldsymbol{\theta}'}$$

- ベイズ推定によって得られるクラス分布関数

$$\hat{p}_{BAYES}(\boldsymbol{x}) = \frac{\int q(\boldsymbol{x}|\boldsymbol{\theta}) \prod_{i=1}^{n} q(\boldsymbol{x}_i|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}{\int \prod_{i=1}^{n} q(\boldsymbol{x}_i|\boldsymbol{\theta}')p(\boldsymbol{\theta}')d\boldsymbol{\theta}'}$$

ベイズ推定法

- 積分の近似法

$$\int g(\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

  - モンテカルロ積分： $p(\boldsymbol{\theta})$ からの標本 $\boldsymbol{\theta}_i$ を用いる

$$\frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{\theta}_i)$$

5.2.2 生成モデルの考え方

- 事後確率を求めるにあたって、同時確率を求めている
  - データが生成される様子をモデル化していると見ることも出来る
    - 事前確率に基づいてクラスを選ぶ
    - そのもとで、特徴ベクトルを出力する

$$
\begin{aligned}
P(\omega_i|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})} \\
&= \frac{p(\omega_i, \boldsymbol{x})}{p(\boldsymbol{x})}
\end{aligned}
$$

事後確率を求めるより、難しい問題を解いているのではないか？

5.3.1 識別モデルの考え方

- 事後確率を直接求める

正例で大きな値

負例で大きな値

あまり関係ない

$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots, \quad x_d)$$

w_1 正の大きな値

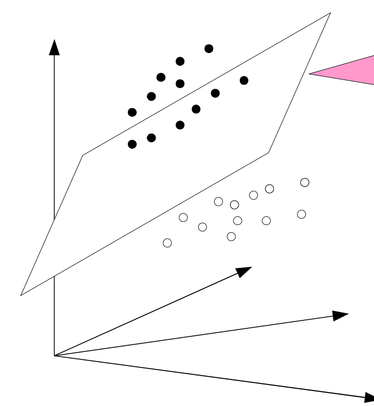
w_2 負の大きな値

w_3 小さな値

正例／負例の閾値

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_d x_d + w_0$$

この値が正なら正例、負なら負例となるように重み w を学習する



この平面を求めていることになる

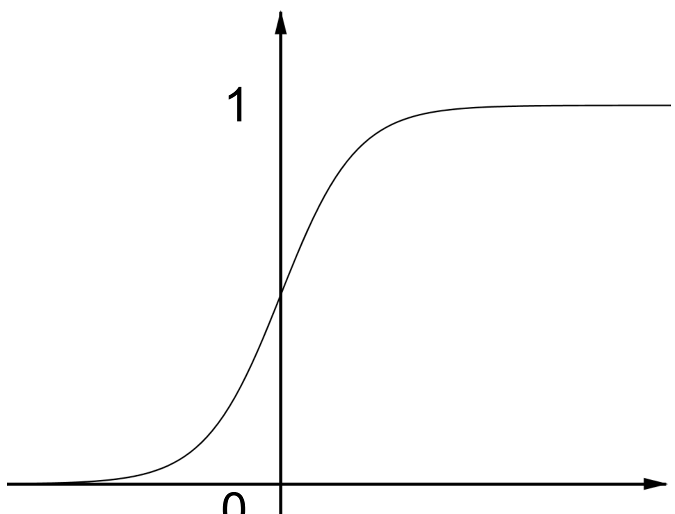
確率と対応づけるには？

5.3.1 識別モデルの考え方

- ロジスティック識別
  - 入力が正例である確率

$$P(\oplus|\boldsymbol{x}) = \frac{1}{1+\exp(-(\boldsymbol{w}\cdot\boldsymbol{x}+w_0))}$$

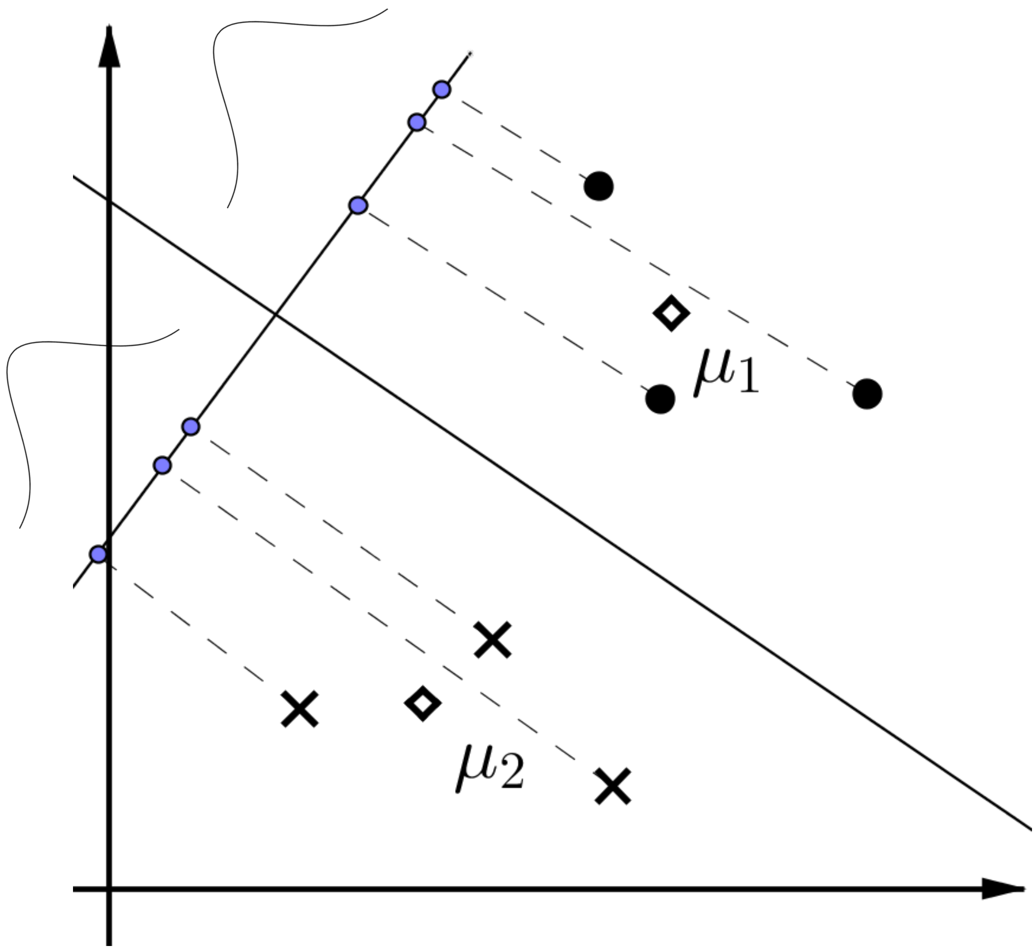
-∞ 〜 +∞ の値域を持つものを、順序を変えずに 0 〜 1 にマッピング



シグモイド関数

5.3.1 識別モデルの考え方

- ロジスティック識別の導出



- 識別面の存在を仮定
- データと識別面との距離が
  $$Dist(\boldsymbol{x}_i) = \boldsymbol{w} \cdot \boldsymbol{x}_i + w_0$$
  となるように重みを調整
- この距離が、クラスごとに正規分布すると仮定 $p(Dist(\boldsymbol{x}_i)|\oplus)$
- $Dist(\boldsymbol{x}_i)$ を $\boldsymbol{x}_i$ とみなす
- ベイズの定理で $p(\oplus|\boldsymbol{x}_i)$ を求めるとシグモイド関数が得られる）

5.3.2 ロジスティック識別器の学習

- 最適化対象＝モデルが学習データを生成する確率

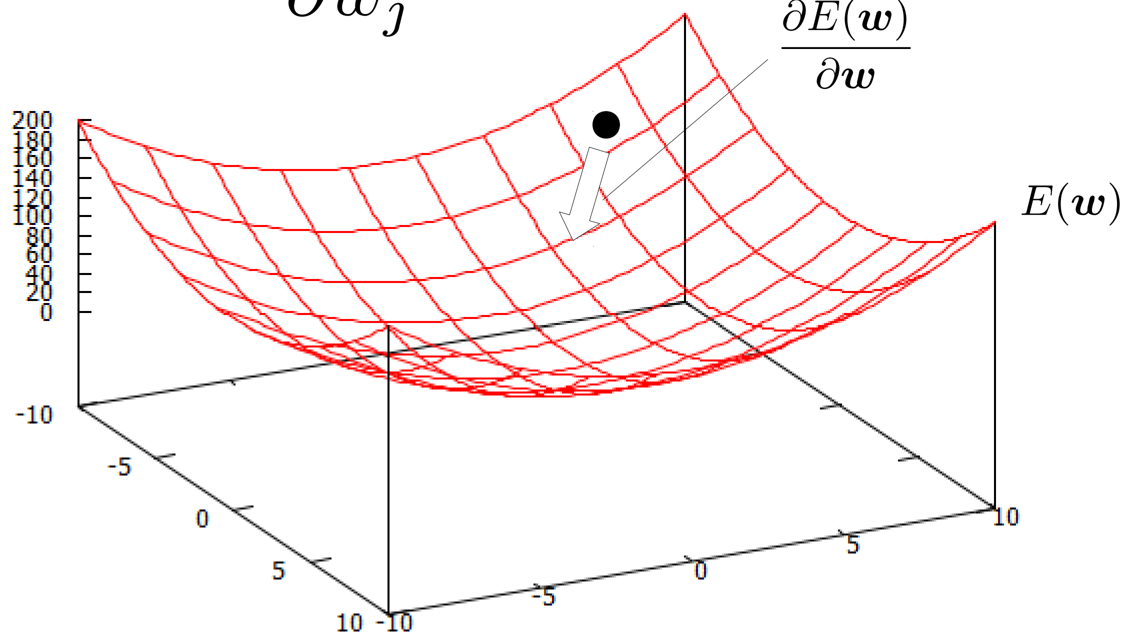
$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\log \prod_{\boldsymbol{x}_i \in D} o_i^{y_i}(1-o_i)^{(1-y_i)}$$

$$o = P(\oplus|\boldsymbol{x})$$
$$y = 0 \;\; or \;\; 1$$

正解ラベル

  - $E(\boldsymbol{w})$ を最急勾配法で最小化

$$w_j \leftarrow w_j - \eta \frac{\partial E(\boldsymbol{w})}{\partial w_j}$$



5.3.2 ロジスティック識別器の学習

- 重み更新量の計算

$$\frac{\partial E(\boldsymbol{w})}{\partial w_j} = \sum_{\boldsymbol{x}_i \in D} \left(\frac{y_i}{o_i} - \frac{1-y_i}{1-o_i}\right) o_i(1-o_i)x_{ij} = \sum_{\boldsymbol{x}_i \in D} (y_i - o_i)x_{ij}$$

- 重みの更新式

$$w_j \leftarrow w_j - \eta \sum_{\boldsymbol{x}_i \in D} (y_i - o_i)x_{ij}$$

この更新を全データ集合ではなく、個別のデータに対して行うのが、確率的最急降下法

多クラスのロジスティック識別

- ソフトマックス関数

$$P(\omega_i|\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_i \cdot \boldsymbol{x})}{\sum_{j=1}^{c} \exp(\boldsymbol{w}_j \cdot \boldsymbol{x})}$$

  - $0 < P(\omega_i \,|\, \boldsymbol{x}) < 1$
  - $\Sigma\, P(\omega_i \,|\, \boldsymbol{x}) = 1$
  - マックス関数（最大値の要素のみ 1 、他は 0 ）をソフトにしたもの

```
deff('y=sm(x)','y=exp(x) ./ sum(exp(x))')
sm([1 2 3 4 6])
```

  - 2 クラスの場合はシグモイド関数に一致

一般的な識別モデル

- 対数線形モデル

$$P(y|\boldsymbol{x}) = \frac{1}{Z}\exp(\boldsymbol{w}\cdot\boldsymbol{\phi}(\boldsymbol{x},y))$$

Z : 確率の和を 1 にするための正規化項

- 素性ベクトル
  - 素性：入力と出力から作った識別に役立つ情報

$$\boldsymbol{\phi}(\boldsymbol{x},y) = (\phi_1(\boldsymbol{x},y),\cdots,\phi_d(\boldsymbol{x},y))$$

- 生成モデルとの違い
  - 素性は、この出力ならばこの特徴というような組み合わせで作ることができる
  - あるクラスの確率が増えれば、残りのクラスの確率が減る