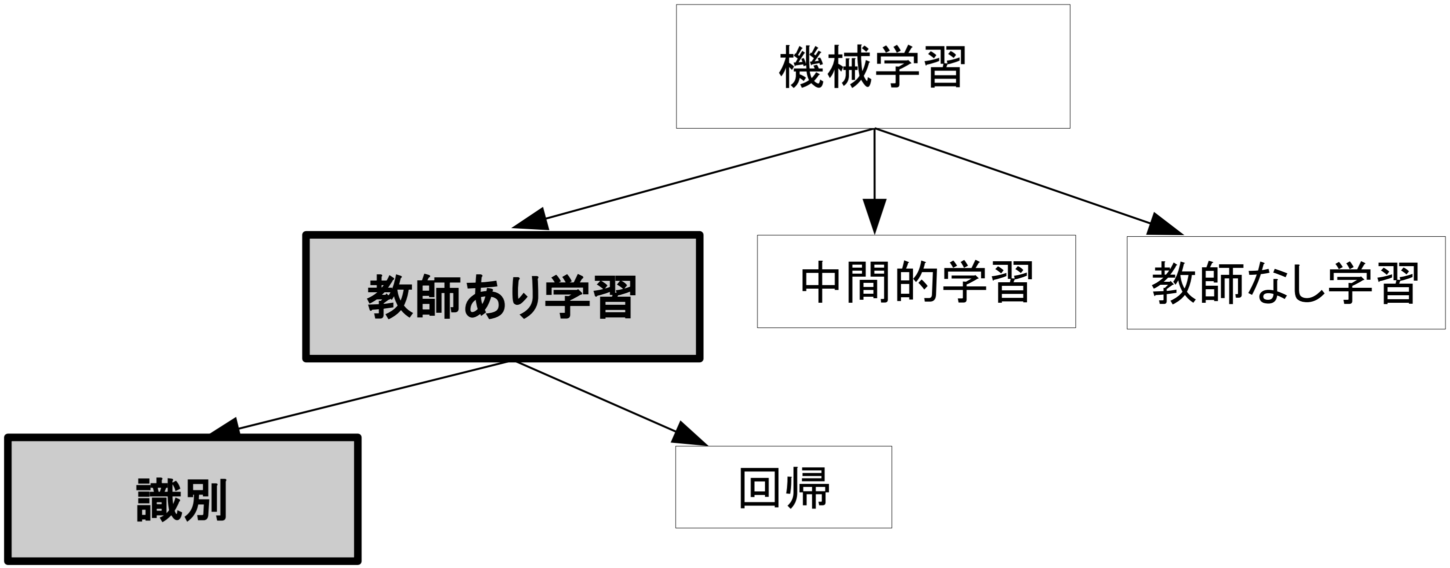
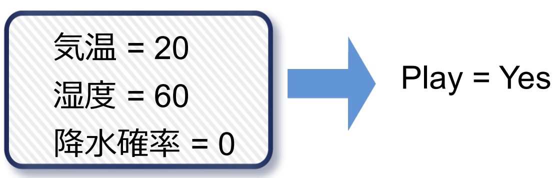
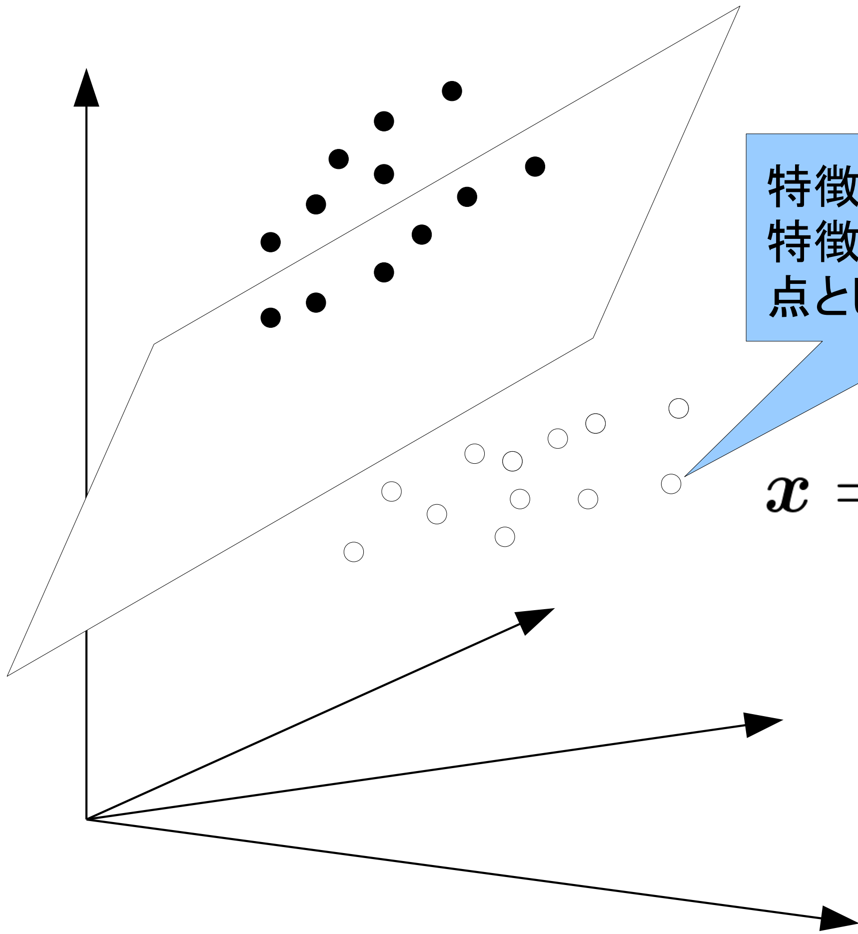
# 5. 識別　ー生成モデルと識別モデルー




- 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




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

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**

# 統計的識別とは（復習）

- 最大事後確率則による識別

$$C_{MAP} = \arg\max_i P(\omega_i|\boldsymbol{x})$$

$\boldsymbol{x}$ ：特徴ベクトル

$\omega_i \quad (1 \leq i \leq c)$ ：クラス

- ベイズの定理 $P(A|B) = \dfrac{P(B|A)P(A)}{P(B)}$

$$\begin{aligned} C_{MAP} &= \arg\max_i P(\omega_i|\boldsymbol{x}) \\ &= \arg\max_i \frac{P(\boldsymbol{x}|\omega_i)P(\omega_i)}{P(\boldsymbol{x})} \\ &= \arg\max_i P(\boldsymbol{x}|\omega_i)P(\omega_i) \end{aligned}$$

尤度

事前確率 $P(\omega_i) = \dfrac{n_i}{N}$

# 尤度 $P(\boldsymbol{x} \mid \omega_i)$ の推定 （復習）

- カテゴリ特徴の場合
  - $P(\text{sunny},\text{hot},\text{high},\text{FALSE} \mid \text{yes})$ などの値をすべての特徴・クラスの組合せについて求めるのは難しい
  - ナイーブベイズの近似

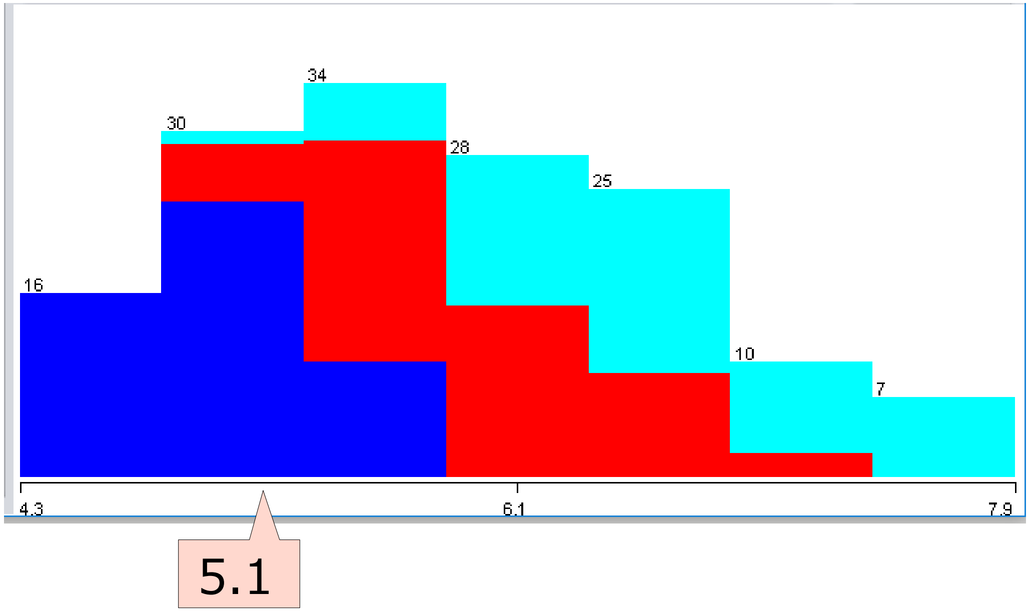
$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$

$$\approx \prod_{j=1}^{d} P(x_j|\omega_i)$$

$P(\text{sunny} \mid \text{yes})$ などを求めて掛け合わせる

# 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

- 数値特徴の場合
  - $P(5.1, 3.5, 1.4, 0.2 \mid \text{iris-setosa})$ などの値をすべての特徴・クラスの組合せについて求めるのは**不可能**
  - ナイーブベイズの近似
    - $P(5.1 \mid \text{iris-setosa})$ の値をどうやって求める？



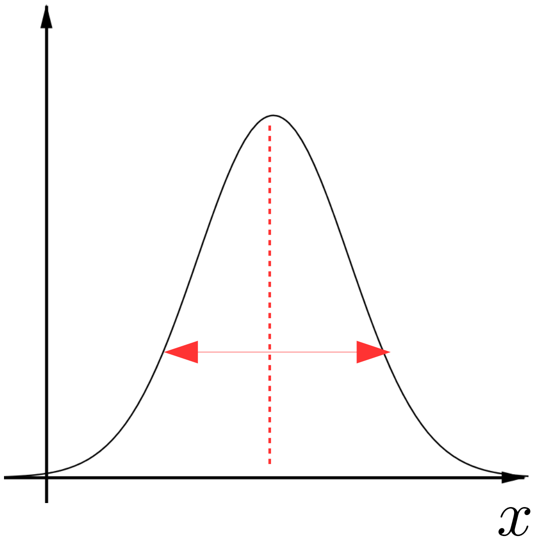
# 5.2.1 数値特徴に対するナイーブベイズ識別

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

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

  - データから平均 $\mu$ と分散 $\sigma^2$ を最尤推定

$P$(sepallength | iris-setosa)
などが正規分布すると仮定する




データの平均と分散を
正規分布のパラメータとする

# iris データのナイーブベイズ識別

```
                         Class
Attribute          Iris-setosa Iris-versicolor  Iris-virginica
                        (0.33)          (0.33)          (0.33)
===============================================================
sepallength
  mean                  4.9913          5.9379          6.5795
  std. dev.              0.355          0.5042          0.6353
  weight sum                50              50              50
  precision             0.1059          0.1059          0.1059

sepalwidth
  mean                  3.4015          2.7687          2.9629
  std. dev.             0.3925          0.3038          0.3088
  weight sum                50              50              50
  precision             0.1091          0.1091          0.1091
```

・・・

# 正規分布とは

- 離散型二項分布の例
  - n 枚のコインを投げた時の、表の枚数の度数
    - n=1　　1 1
    - n=2　　1 2 1
    - n=3　　1 3 3 1
    - n=4　　1 4 6 4 1
    - n=5　　1 5 10 10 5 1

    …
  - n→∞　の時の分布が正規分布

# 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

$$P(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i)P(\omega_i)}{p(\boldsymbol{x})}$$

$$= \frac{p(\omega_i, \boldsymbol{x})}{p(\boldsymbol{x})}$$

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？

# 5.3.3 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

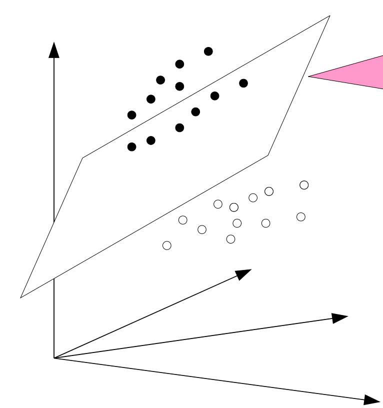
$$\boldsymbol{x} = (x_1, \quad x_2, \quad x_3, \quad \ldots, \quad x_d)$$

$$\Bigg\downarrow \begin{array}{l} w_1 \\ \text{正の} \\ \text{大きな値} \end{array} \qquad \Bigg\downarrow \begin{array}{l} w_2 \\ \text{負の} \\ \text{大きな値} \end{array} \qquad \Bigg\downarrow \begin{array}{l} w_3 \\ \text{小さな値} \end{array}$$

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 $\boldsymbol{w}$ を学習する



この平面を求めていることになる

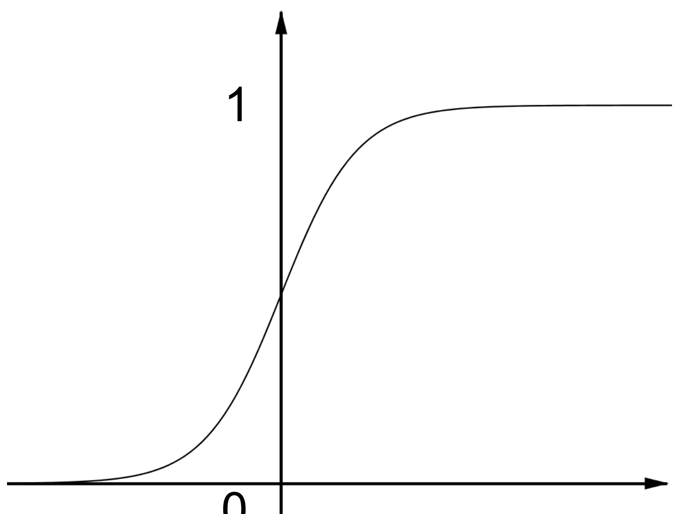
**確率と対応づけるには？**

# 5.3.4 ロジスティック識別

- 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

# 5.3.4 ロジスティック識別

- 最適化対象＝モデルが学習データを生成する確率

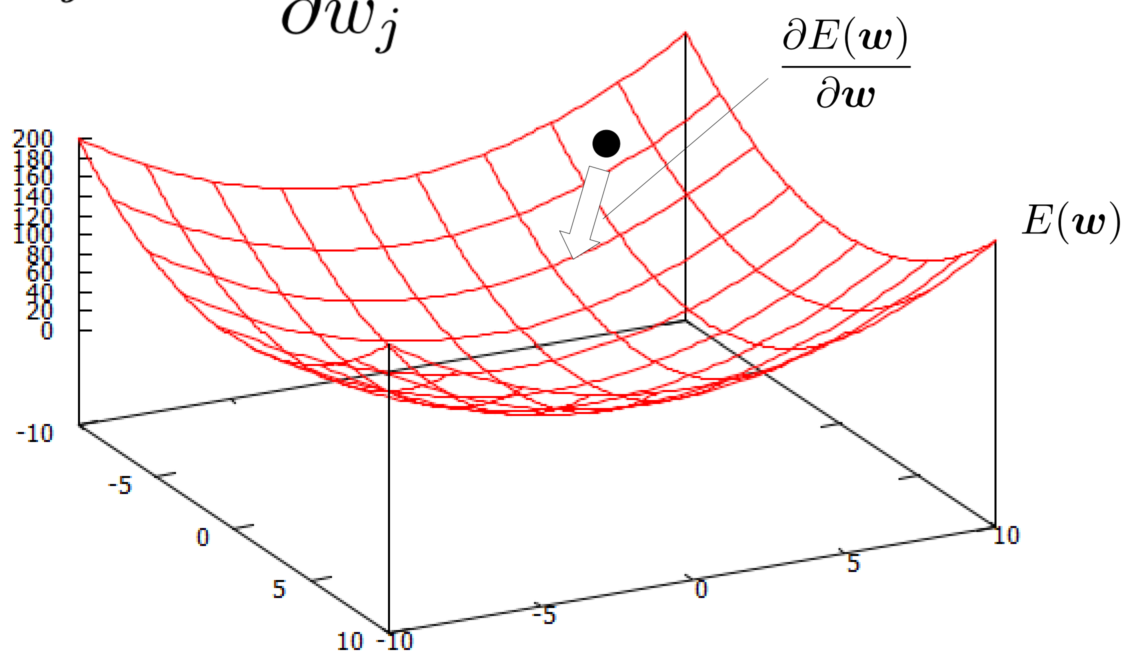
$$E(\boldsymbol{w}) = -\log P(D|\boldsymbol{w}) = -\log \prod_{\boldsymbol{x}_i \in D} o_i^{y_i}(1-o_i)^{(1-y_i)}$$

  - $E(\boldsymbol{w})$ を最急降下法で最小化

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

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

正解ラベル



# 5.3.4 ロジスティック識別

- 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

# まとめ

- Weka デモ
  - iris データ、 glass データ
  - NaiveBayes, SimpleLogistic
- ナイーブベイズ
  - 特徴の各次元に対して、 1 次元正規分布のパラメータ（平均・分散）を最尤推定
- ロジスティック識別
  - 特徴の重み付き和で得られる「そのクラスらしさ」を確率値に変換

# 次回までの勉強

- Weka
  - インストールして、各章の例題・演習問題を実行
- Python
  - Google Colaboratory で各章の例題・演習問題を実行
    - Google アカウントが必要
- 余裕があれば
  - Chainer チュートリアル（1 ～ 6）

    https://tutorials.chainer.org/ja/tutorial.html