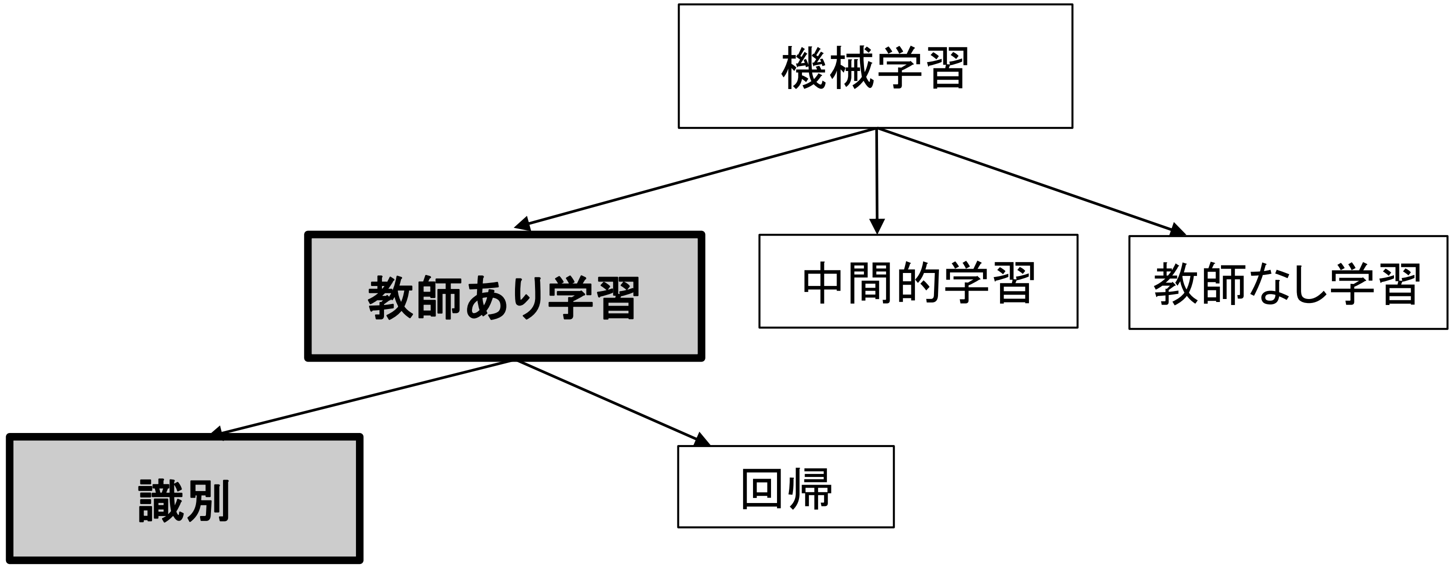
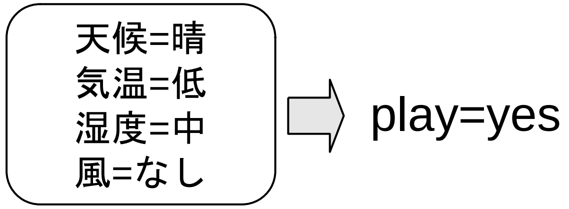
4. 識別　ー統計的手法ー



- カテゴリ特徴



- 第3章（決定木）：正解を表現する概念を得る　説明性
- 第4章（統計）　：識別結果の確率を得る　意思決定

weatherデータ

表 3.3　weather.nominal.arff（カテゴリ特徴）

No.	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

- outlook（天候）
  - sunny, overcast, rainy
- temperature（気温）
  - hot, mild, cool
- humidity（湿度）
  - high, normal
- windy（風）
  - TRUE, FALSE
- play（=クラス）
  - yes, no

4.1 統計的識別とは

- 特徴ベクトル $\boldsymbol{x}$ が観測されていないとき
  - 事前確率 $P(\mathrm{yes})$, $P(\mathrm{no})$ だけから判断するしかない
- 特徴ベクトル $\boldsymbol{x}$ 観測後
  - 事後確率 $P(\mathrm{yes} \mid \boldsymbol{x})$, $P(\mathrm{no} \mid \boldsymbol{x})$ の大きい方に判定
- 多クラスに一般化
  - 最大事後確率則による識別（ベイズ識別）

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

$\boldsymbol{x}$ ：特徴ベクトル

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

4.1 統計的識別とは

- 事後確率の求め方
  - 単純な方法としては、特徴ベクトルが完全に一致する事例を大量に集めて、その正解の頻度を求める

    例）$\boldsymbol{x}$ =(晴, 高, 中, True) 100事例中 yes:70, no:30
  - 上記の推定が行えるようなデータセットが得られることはほとんどない
  - 事後確率に対して、式変形・近似を行って、現実の規模のデータセットから確率を推定できるようにする

4.1 統計的識別とは

- ベイズの定理

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

証明 $P(A, B) = P(A|B)P(B)$

同時確率

$= P(B|A)P(A)$

- 事後確率の式の変形

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

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

$$= \arg\max_i \underbrace{P(\boldsymbol{x}|\omega_i)}_{\text{尤度}}\underbrace{P(\omega_i)}_{\text{事前確率}}$$

ベイズの定理

上式の分母は
判定に寄与しない

- 尤度：特定のクラスから、ある特徴ベクトルが出現する尤もらしさ

4.1 統計的識別とは

- ベイズ統計とは
  - 結果から原因を求める
    - 通常の統計学は原因から結果を予測する
  - ベイズ識別
    - 観測結果 $\boldsymbol{x}$ から、それが生じた原因 ω_i を求める（事後確率）
    - 通常、確率が与えられるのは原因→結果（尤度）
    - ベイズ識別では、事前分布 $P(\omega_i)$ が、観測 $\boldsymbol{x}$ によって事後分布 $P(\omega_i \mid \boldsymbol{x})$ に変化したと考えることができる

4.2 カテゴリ特徴に対するベイズ識別

- 事前確率　$P(\omega_i)$
  - 特徴ベクトルを観測する前の各クラスの起きやすさ
- 事前確率の最尤推定

$$P(\omega_i) = \frac{n_i}{N}$$

N: 全データ数、n_i :クラスω_iのデータ数

4.2.1 学習データの対数尤度

- 尤度の導出
  - 特徴ベクトル $\boldsymbol{x}$ を生成する（各クラスごとの）モデルを考え、そのモデルがパラメータ $\boldsymbol{\theta}$ に従ってデータを生成していると仮定

$$P(\boldsymbol{x}|\omega, \boldsymbol{\theta})$$

以後、1クラス分のデータを全データとみなす
⇒ ωを省略

  - 全データ D は、各データが同じ分布から独立に生成されていると仮定
    - i.i.d (independent and identically distributed)

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

4.2.1 学習データの対数尤度

- 対数尤度
  - 確率の積のアンダーフローを避けるため、対数尤度で計算

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

- 尤度関数の仮定
  - 特徴ベクトルが1次元、値0 or 1で、ベルヌーイ分布に従うと仮定
    - ベルヌーイ分布：確率θで値1、確率1-θで値0をとる分布

$$\begin{aligned}\mathcal{L}(D) &= \sum_{i=1}^{N} \log \theta^{x_i} (1-\theta)^{1-x_i} \\ &= \sum_{i=1}^{N} x_i \log \theta + (N - \sum_{i=1}^{N} x_i) \log(1-\theta)\end{aligned}$$

4.2.1 学習データの対数尤度

- 対数尤度を最大にするパラメータ $\hat{\theta}$
  - $\dfrac{\partial \mathcal{L}(D)}{\partial \theta} = 0$ の解を求める

$$\frac{\partial \mathcal{L}(D)}{\partial \theta} = \sum_{i=1}^{N} x_i \frac{1}{\theta} - (N - \sum_{i=1}^{N} x_i) \frac{1}{1-\theta}$$

$$= \frac{1}{\theta(1-\theta)} \{(1-\theta)\sum_{i=1}^{N} x_i - \theta(N - \sum_{i=1}^{N} x_i)\} = 0$$

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

値 x_i をとる回数を全データ数Nで割ったもの
⇒　最尤推定法

4.2.2 ナイーブベイス識別

- 多次元ベクトルの尤度関数を求める
  - 特徴値のすべての組合せが、データセット中に何度も出てくる必要があるが、これも非現実的
- ナイーブベイズの近似
  - すべての特徴が独立であると仮定

$$
\begin{aligned}
P(\boldsymbol{x}|\omega_i) &= P(x_1, \ldots, x_d|\omega_i) \\
&\approx \prod_{j=1}^{d} P(x_j|\omega_i)
\end{aligned}
$$

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

4.2.2 ナイーブベイス識別

- 尤度の最尤推定

$$P(x_j \mid \omega_i) = \frac{n_j}{n_i}$$

n_i : クラスω_iのデータ
n_j : クラスω_iのデータのうち、j次元目の値がx_j の個数

ゼロ頻度問題： n_j が0の場合、確率の推定値も0

- スムージング
  - 事前にその値がα回生じていたと仮定する

$$P(x_j \mid \omega_i) = \frac{n_j + \alpha}{n_i + \alpha m}$$

m: j次元目の値の種類数

  - ラプラス推定：α=1のとき

scikit-learnのナイーブベイズ識別

- カテゴリ特徴は OrdinalEncoder で整数値に置き換える
  - 変換情報

```
[array(['overcast', 'rainy', 'sunny'], dtype=object),
 array(['cool', 'hot', 'mild'], dtype=object),
 array(['high', 'normal'], dtype=object),
 array([False, True], dtype=object)]
```

```
['sunny', 'hot', ‘high’, False]

[   2,      1,      0,      0   ]
```

- 正解のラベルは LabelEncoder で整数値に置き換える

  yes → 1,　no → 0

scikit-learnのナイーブベイズ識別

- カテゴリ特徴に対するナイーブベイズ識別は CategoricalNB を用いる
  - 識別器のパラメータ
    - arpha：事前に仮定するサンプル数。教科書の mp に対応
    - fit_prior：事前確率を学習の対象とするかどうか
    - class_prior：事前確率を別途与えるときに用いる

scikit-learnのナイーブベイズ識別

- 典型的なコード

```
clf = CategoricalNB()
clf.fit(X, y)
clf.predict_proba(X_test[1])
```

インスタンスの作成

学習

識別

まとめ

- カテゴリ特徴の識別問題に対する統計的識別
    - ベイズ識別
        - 事後確率 $P(\omega_i \mid \boldsymbol{x})$ を最大とするクラス ω_i を求める
        - 事後確率をデータから推定するのは難しいので、ベイズの定理を用いて尤度 $P(\boldsymbol{x} \mid \omega_i)$ と事前確率 $P(\omega_i)$ の積に分解
    - ナイーブベイズ法
        - 特徴のすべての次元が独立であると仮定して、尤度をそれぞれの次元の確率の積に分解
        - 確率が0となることを避けるためにスムージングを行う