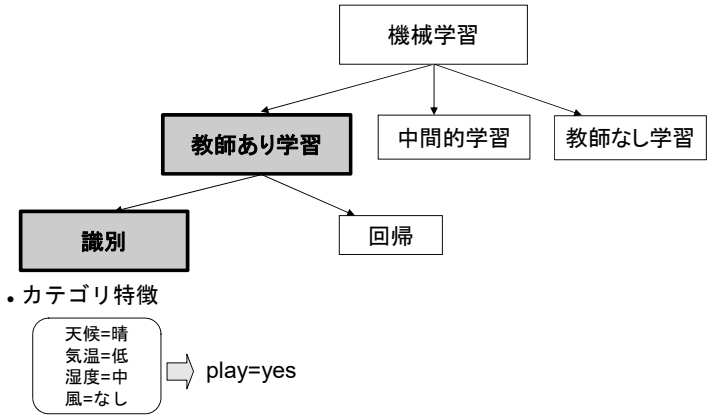
4. 識別　ー統計的手法ー



- 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

この章では前章に引き続き，教師あり学習における識別問題で，特徴ベクトルの要素がすべてカテゴリの場合から説明をはじめます．前章と異なるのは，識別に統計的手法を用いることによって，結果に確信度を付与することができる点です．

4.1 統計的識別とは

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

例に用いるweather.nominalデータは．別名Golfデータともよばれ，ある人がゴルフをするか否かを気象条件を特徴として決定するという識別問題用のデータです．
クラス特徴はplayの値でyesかnoです．
特徴は天気，気温，湿度，風です．

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

事象 yes が起きる確率

- 特徴ベクトル $\boldsymbol{x}$ 観測後
  - 事後確率 $P(\mathrm{yes}\,|\,\boldsymbol{x})$, $P(\mathrm{no}\,|\,\boldsymbol{x})$ の大きい方に判定

条件付き確率
事象 $\boldsymbol{x}$ が起きたもとで
事象 yes が起きる確率

- 多クラスに一般化
  - 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

まず，特徴ベクトル x が観測されていないときにどうやって判定するかを考えてみます．
この場合は，事前確率 P(yes), P(no) だけから判断するしかないでしょう．
次に，特徴ベクトル x が観測された後で判定することを考えます．
これが通常のタスクです．
この場合は，条件付き確率の形式で表現されている事後確率 P(yes| x), P(no| x) の大きい方に判定します．
これを一般化すると，最大事後確率則による識別を行うことになります．
argmaxは引数の値が最大となるときのパラメータを返すものです．

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値を推定できるようにする

次に，事後確率の求め方を考えます．
通常は，特徴ベクトルが完全に一致する事例を大量に集めて頻度を求めます．
しかし，このような状況はあまり考えられないので，データから直接的にこの確率を求めるのは難しいといえます．
そこで事後確率に対して、式変形・近似を行って、現実の規模のデータセットから値を推定できるようにします．

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

そこでベイズの定理を用いて，事後確率の式を変形します．
証明は2つの事象A,Bの同時確率が、単独の事象の確率と、そのもとでの条件付き確率の積で表現できることから導けます．

4.1 統計的識別とは

- 事後確率の式の変形

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

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

ベイズの定理

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

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

次に，事後確率に対してベイズの定理を用いて，最大事後確率則の式を変形します．
変形後の式の分母は、クラス番号iを含まないので、全クラスについて共通の値となり、最大値の判定に寄与しないので削除します．
最後に得られた式の第一項を尤度とよびます．
クラスωiから特徴ベクトルxが出てくる尤もらしさという意味です．
最終的に、尤度と事前確率の積が最大となるクラスを求めればよいということになります．

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 $P(\omega_i|\boldsymbol{x})$ に変化したと考えることができる

この方法がベイズ識別とよばれるのは、このような理由からです.

4.1 統計的識別とは

- 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

事前確率は、特徴ベクトルを観測する前の、各クラスの起こりやすさです。
これは単純に、データセット中の各クラスのデータ数を全データ数で割ることで推定できます.

4.2 カテゴリ特徴に対するベイズ識別
4.2.1 学習データの対数尤度

- 最尤推定法の導出
  - 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

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

以後、1クラス分のデータを全データとみなす
⇒ ω_i を省略

  - 全データ D は、それぞれ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)

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

それでは尤度を求める方法を考えてゆきましょう．
それぞれのクラスは，特徴ベクトルxを生成する何らかの確率分布Pをもっていて，その確率分布はパラメータθで表現できるものとします．
学習データはその同一の確率分布から，各事例独立に生成されたものと仮定します．
この仮定をiidとよびます．
iidを仮定すると，各データxの生成は独立事象なので，全データDの尤度は各データxの尤度の積となります．

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

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

確率は0以上1以下の値で，それらを一般に大きな数であるN回掛け合わせると、とても小さな数になり、正確な計算ができません.
この現象をアンダーフローとよびます.
アンダーフローを避けるため、対数尤度を使って計算します.
対数は単調増加関数なので、尤度を最大にするパラメータと対数尤度を最大にするパラメータは等しくなります.

4.2.1 学習データの対数尤度

- 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

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

尤度関数について、特徴ベクトルが1次元で、値は0または1をとるものとし、その値はベルヌーイ分布に従うと仮定します.
ベルヌーイ分布とは確率θで値1、確率1-θで値0をとる分布のことです.
この分布をPとすると、対数尤度はこのように書き換えられます.

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 $\hat{\theta}$
  - $\frac{\partial \mathcal{L}(D)}{\partial \theta} = 0$ 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

ここで、対数尤度の最大値を求めるために、変数θで偏微分した値が0となる極値を求めます.
式変形してθについて解くと、対数尤度を最大にするパラメータθ^は、値xiをとる回数を全データ数で割ったものとなります.
このように、データの頻度に基づいてパラメータを推定する方法を最尤推定法とよびます.

4.2.2 ナイーブベイズ識別

- 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

$$P(\boldsymbol{x}|\omega_i) = P(x_1, \ldots, x_d|\omega_i)$$
$$\approx \prod_{j=1}^{d} P(x_j|\omega_i)$$
$$C_{NB} = \arg\max_i P(\omega_i) \prod_{j=1}^{d} P(x_j|\omega_i)$$

次に、多次元ベクトルの尤度関数の求め方を考えます.
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のですが、これも現実のデータセットではほとんどありえません.
そこで、すべての特徴が独立であると仮定すると、多次元ベクトルの尤度は、1次元変数の尤度の積で近似することができます.
これをナイーブベイズの近似とよび、この方法による識別をナイーブベイズ識別または単純ベイズ識別とよびます.

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_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

- 確率のm推定

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

p: 事前に見積もった各特徴値の割合
m: 事前に用意する標本数

  - ラプラス推定
    - *m*:特徴値の種類数、*p*:等確率　とすると、*mp=1*

しかし，このように少ないデータでも学習が行えるように尤度計算の方法を単純にしても，学習データが少ないがゆえに生じる問題がまだあります．
あるクラスのある特徴に限定したとしても、特定の値が1度も観測されない、いわゆるゼロ頻度問題が生じます．
このようなゼロ頻度問題へ対処するには，確率のm推定という考え方を用います．
これは m 個の仮想的なデータがすでにあると考え，それらによって各特徴値の出現は事前にカバーされているという状況を設定します．
各特徴値の出現割合 p を事前に見積もり，事前に用意する標本数を m とすると，尤度は推定式のようになります．
この工夫によって，ゼロ頻度問題が回避できることになります．

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  ]
```

scikit-learnのナイーブベイズ識別

- カテゴリで表された正解情報は 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