B-6）単純ベイズでテキスト分類

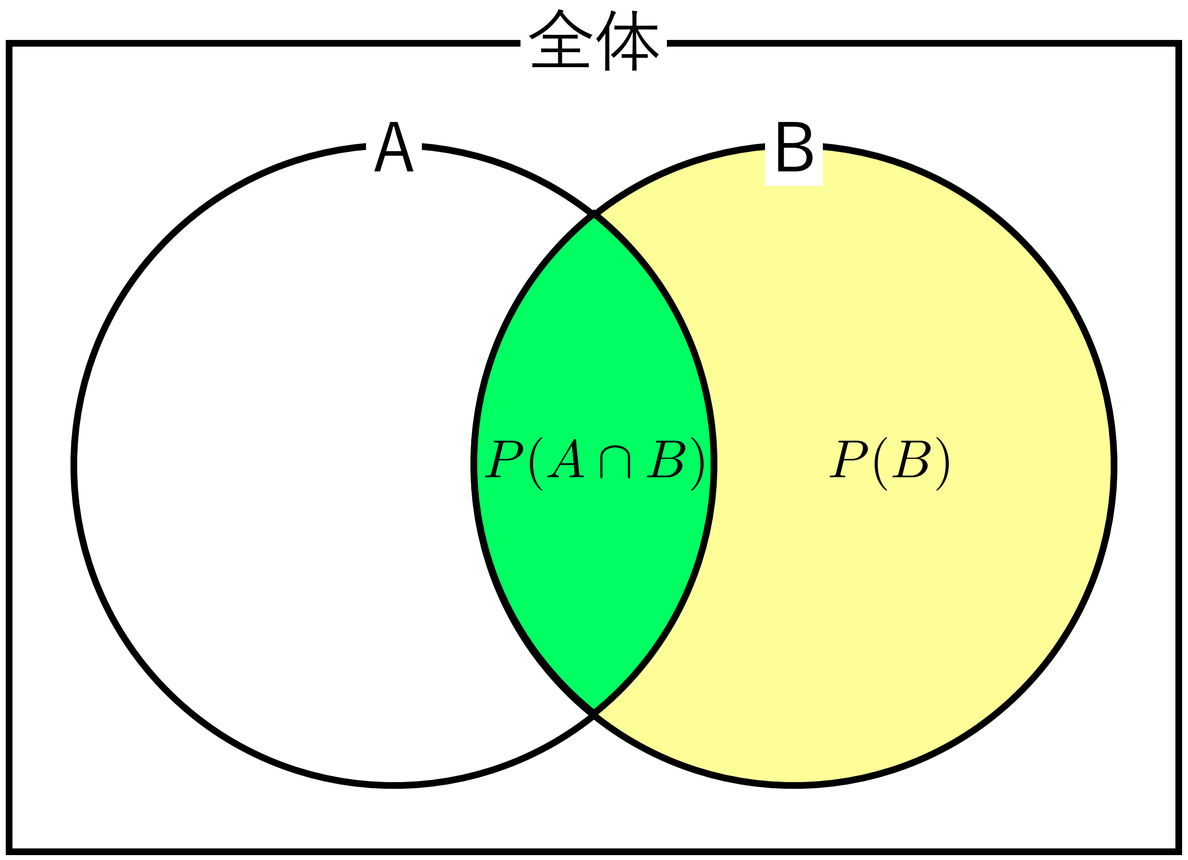
社会システム科学

単純ベイズ分類器

条件付き確率

- 事象Bが起こったという前提のもとで事象Aが起こる確率

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



$P(A \cap B)$

同時確率（事象Aと事象Bが同時に起こる確率）

※事象Aと事象Bが独立の場合は以下

$$P(A \cap B) = P(A) \cdot P(B)$$

条件付き確率

- 検査陽性になった人が実際に陽性である確率は？
  - 全体の1%がかかっている病気がある
  - 病気になっていない人でも10%は検査陽性（偽陽性）（
  - 病気になっている人の90%は検査陽性（10%は検査陰性（偽陰性））

ベイズの定理

- 条件付き確率の式から：

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$$

- これを変形して：

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

ベイズの定理

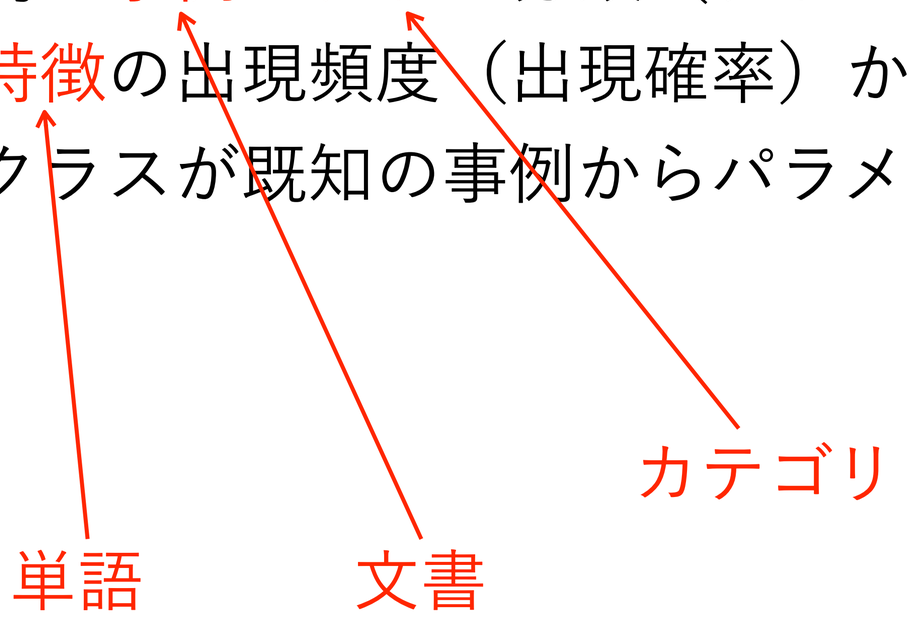
- 傘を持っている場合に雨が降っている確率は？
  - 雨が降ったら傘を持っている確率 0.8
  - 雨が降る確率 0.4
  - 雨でも晴れでもとにかく傘を持っている確率 0.5

単純ベイズ分類器（Naive Bayes Classifier）

- 目的：事例のクラス分類（クラスは既知）
  - 特徴の出現頻度（出現確率）からクラスへの分類確率を計算
  - クラスが既知の事例からパラメータを学習

単純ベイズ分類器：テキスト分類の場合

- 目的：事例のクラス分類（クラスは既知）
  - 特徴の出現頻度（出現確率）からクラスへの分類確率を計算
  - クラスが既知の事例からパラメータを学習



テキスト分類のための単純ベイズ分類器

- 目的
  - 文書に対して P(カテゴリ|文書) が最大となるカテゴリを求める
  - ベイズの定理から以下のように変形：

$$P(\text{カテゴリ} \mid \text{文書}) = \frac{P(\text{文書} \mid \text{カテゴリ}) \cdot P(\text{カテゴリ})}{P(\text{文書})}$$

  - 単一の文書に対して比較を行うならば**分子のみ**を比較すればOK

$$P(\text{カテゴリ} \mid \text{文書}) \propto P(\text{文書} \mid \text{カテゴリ}) \cdot P(\text{カテゴリ})$$

  - 単語の出現確率を互いに独立として以下のように変形：

$$P(\text{文書} \mid \text{カテゴリ}) = \Pi_i P(\text{単語}_i \mid \text{カテゴリ})$$

文書からこの確率分布を推定（学習）

単純ベイズ分類器の種類

- 単純ベイズ分類器の種類＝確率分布の種類
- 特徴によって複数の種類がある
  - 特徴が連続値 → ガウス分布（正規分布）
  - 特徴が二値　　→ ベルヌーイ分布
  - 特徴が離散値 → 多項分布

［演習］Google Colabで単純ベイズ分類器

［演習］単純ベイズ分類器を使ってみる

- ここからは Google Colaboratory で作業します。
- ノートブックの説明を見ながら解説します。
- ノートブックの指示に応じてこちらの資料に戻って参照します。

単純ベイズ分類器に必要なパッケージ

- Python用パッケージ
  - scikit-learn：機械学習用パッケージ

[参考]

- scikit-learnに含まれる機械学習の種類と利用法について：

  `http://scikit-learn.org/stable/tutorial/machine_learning_map`

学習セットの用意

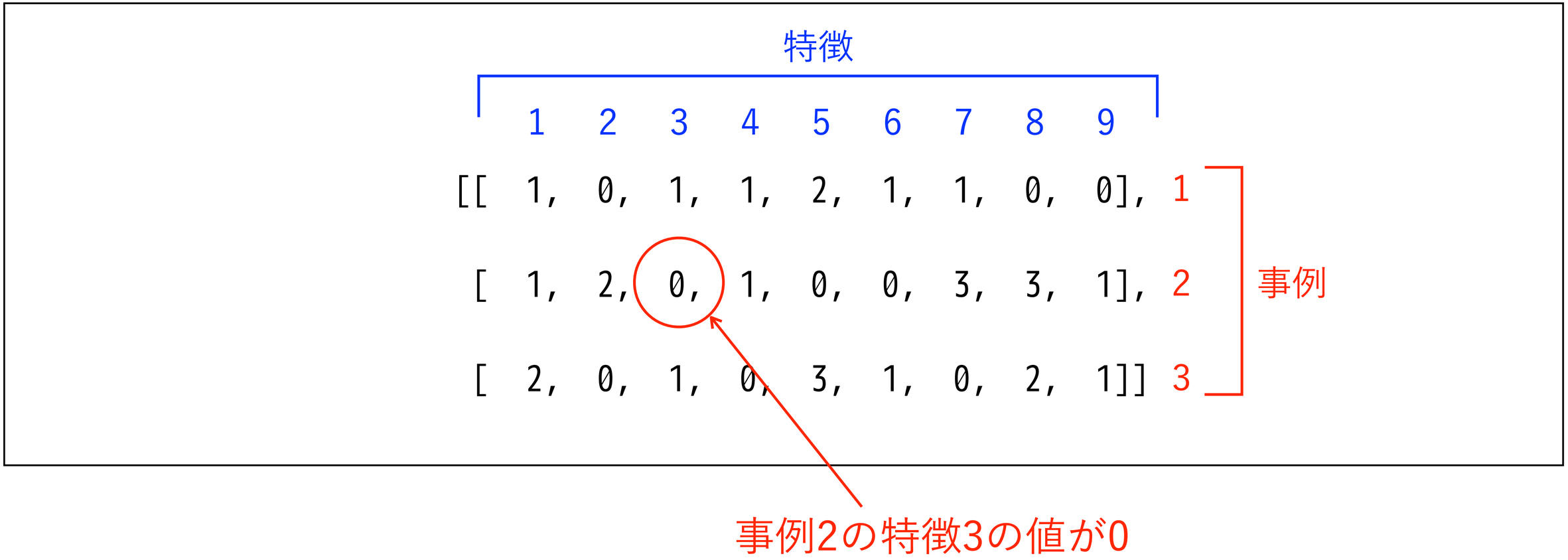
学習セット＝事例（特徴）＋クラス

```
X = np.array([[1, 0, 1, 1, 2, 1, 1, 0, 0],
              [1, 2, 0, 1, 0, 0, 3, 3, 1],
              [2, 0, 1, 0, 3, 1, 0, 2, 1]])
y = np.array([1, 2, 3])
```

特徴ベクトル

それぞれの事例が分類されるクラス

特徴ベクトルの形式

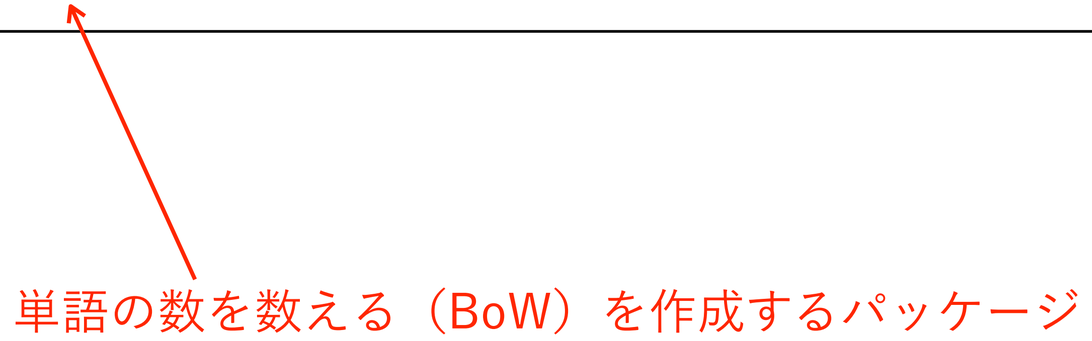

特徴
1 2 3 4 5 6 7 8 9
[[1, 0, 1, 1, 2, 1, 1, 0, 0], 1
[1, 2, 0, 1, 0, 0, 3, 3, 1], 2
[2, 0, 1, 0, 3, 1, 0, 2, 1]] 3
事例
事例2の特徴3の値が0

テキストの分類

パッケージを読み込む

1. 必要なパッケージを読み込む

```
import numpy as np
from sklearn.naive_bayes import GaussianNB
from sklearn.feature_extraction.text import CountVectorizer
```



もっと長いテキスト文書の分類

テキストファイルの準備

1. 前回の資料からテキストファイルをダウンロード
   - 34ro_content.txt（夏目漱石「三四郎」）
   - bot_content.txt（夏目漱石「坊ちゃん」）
   - sore_content.txt（夏目漱石「それから」）
   - jigokuhen_content.txt（芥川竜之介「地獄変」）
   - kappa_content.txt（芥川竜之介「河童」）
   - ningen_content.txt（太宰治「人間失格」）
   - shayou_content.txt（太宰治「斜陽」）
2. セッションストレージ（左側のファイルのところ）にアップロード