# 实验七：朴素贝叶斯分类器

姓名：　　　　　　　　　　　　　　　　学号：

## 实验目的

理解和掌握朴素贝叶斯基本原理和方法，理解极大似然估计方法，理解先验概率分布和后验概率分布等概念，掌握朴素贝叶斯分类器训练方法。

## 实验要求

给定数据集，编程实现朴素贝叶斯分类算法，计算相应先验概率，条件概率，高斯分布均值和方差的估计值，并给出模型在测试集上的精度。

## 实验环境

python, numpy, scipy

## 实验代码

```
import pandas as pd

import numpy as np


# 读取数据

train_data = np.loadtxt('experiment_07_training_set.csv',
                        usecols=[1, 2, 3, 4],
                        delimiter=',',
                        skiprows=1)

train_tag = np.loadtxt('experiment_07_training_set.csv',
                       usecols=[-1],
                       delimiter=',',
                       skiprows=1,
                       dtype=str)
```

```
test_data = np.loadtxt('experiment_07_testing_set.csv',
                       usecols=[1, 2, 3, 4],
                       delimiter=',',
                       skiprows=1)
test_tag = np.loadtxt('experiment_07_testing_set.csv',
                      usecols=[-1],
                      delimiter=',',
                      skiprows=1,
                      dtype=str)

tag1_data = train_data[train_tag == "Iris-setosa", :]
tag2_data = train_data[train_tag == "Iris-versicolor", :]
tag3_data = train_data[train_tag == "Iris-virginica", :]
p_tag1 = tag1_data.shape[0] / train_data.shape[0]
p_tag2 = tag2_data.shape[0] / train_data.shape[0]
p_tag3 = tag3_data.shape[0] / train_data.shape[0]
print("P(Y=setosa:)",p_tag1)
print("P(Y=versicolor:)",p_tag2)
print("P(Y=virginica:)",p_tag3)
tag1_mean = np.mean(tag1_data, axis=0)
tag2_mean = np.mean(tag2_data, axis=0)
tag3_mean = np.mean(tag3_data, axis=0)
print("均值:")
print(tag1_mean)

print(tag2_mean)

print(tag3_mean)

tag1_std = np.std(tag1_data, axis=0)

tag2_std = np.std(tag2_data, axis=0)

tag3_std = np.std(tag3_data, axis=0)

print("标准差:")

print(tag1_std)

print(tag2_std)

print(tag3_std)



def calculate_pdf(test_data, tag_mean, tag_std, p_tag):

    test_tag = np.zeros([test_data.shape[0], 4])

    for i in range(4):

        test_tag[:, i] = 1 / (np.sqrt(2 * np.pi) * tag_std[i]) * np.exp(

            -(test_data[:, i] - tag_mean[i]) ** 2 / (2 * tag_std[i] ** 2))

    p_test_tag = np.ones(test_tag.shape[0])

    p_test_tag *= p_tag

    for i in range(4):

        p_test_tag *= test_tag[:, i]

    return p_test_tag
```

```
p_test_tag1 = calculate_pdf(test_data, tag1_mean, tag1_std, p_tag1)
p_test_tag2 = calculate_pdf(test_data, tag2_mean, tag2_std, p_tag2)
p_test_tag3 = calculate_pdf(test_data, tag3_mean, tag3_std, p_tag3)

p_test_tag1 = p_test_tag1.reshape(1, -1)
p_test_tag2 = p_test_tag2.reshape(1, -1)
p_test_tag3 = p_test_tag3.reshape(1, -1)

concatenated_array_col = np.hstack((p_test_tag1.T, p_test_tag2.T, p_test_tag3.T))
# print(concatenated_array_col)
index_max = np.argmax(concatenated_array_col, axis=1)
print(index_max)
test_tag = pd.Series(test_tag)
test_tag = pd.factorize(test_tag)[0]
print(test_tag)

acc = np.sum(index_max == test_tag)
print('Test Accuracy', acc / test_tag.shape[0])
```

- 结果分析

（1）先验概率

| 类别 | 先验概率 |
|---|---|
| $P(Y = setosa)$ | 0. 4 |
| $P(Y = versicolor)$ | 0. 4 |
| $P(Y = virginica)$ | 0. 2 |

(2) 高斯分布参数估计

| 类别 | $X1 = SepalLength$ | | $X2 = SepalWidth$ | | $X3 = PetalLength$ | | $X4 = PetalWidth$ | |
|---|---|---|---|---|---|---|---|---|
| | 均值 | 标准差 | 均值 | 标准差 | 均值 | 标准差 | 均值 | 标准差 |
| $P(X\|Y = setosa)$ | 5.0375 | 0.35755244 | 3.44 | 0.35972211 | 1.4625 | 0.16983448 | 0.2325 | 0.09845684 |
| $P(X\|Y = versicolor)$ | 6.015 | 0.51261584 | 2.7875 | 0.32572036 | 4.32 | 0.44395946 | 1.35 | 0.20493902 |
| $P(X\|Y = virginica)$ | 6.56 | 0.71302174 | 2.92 | 0.37629775 | 5.655 | 0.62407932 | 2.045 | 0.26734809 |

(3) 模型精度：

```
[0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
 2 2 2 2 2 2 1 1 2 2 2 1 2]
[0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2]
Test Accuracy 0.92
```