实验一 贝叶斯判别方法

一、实验目的

掌握统计判别的基本思想。通过对贝叶斯分类方法的学习，掌握贝叶斯判别原则、最小风险判别原则、最大似然函数判别原则，正态分布的贝叶斯分类器，贝叶斯分类器的错误概率，聂曼-皮尔逊准则，均值向量与协方差矩阵的估计。

二、实验内容

本实验通过贝叶斯判别原理对 Iris 鸢尾花数据集中的三类样本进行两两分类。数据集包含 150 个数据样本，分为 3 类，每类 50 个数据，每个数据包含 4 个属性。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度 4 个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。模型假设服从多维正态分布。

三、实验原理

1、贝叶斯判别原理

对于二分类问题，用ω_i表示样本所属类别，假设先验概率$P(\omega_1), P(\omega_2)$已知。这个假设是合理的，因为如果先验概率未知，可以从训练特征向量中估算出来。如果 N 是训练样本的总数，其中有N_1, N_2个样本分别属于ω_1, ω_2，则相应的先验概率为$P(\omega_1) = N_1/N,\ P(\omega_2) = N_2/N$。此外，假设类条件概率密度函数$P(x|\omega_i)$是已知的参数，用来描述每一类特征向量的分布情况。如果类条件概率密度函数是未知的，则可以从训练数据集中估算出来。

(1) 分类所使用的特征为 n 维特征向量$x = [x_1\ x_2\ x_3 \dots\ x_n]^T$

(2) 两类先验概率值分别为$P(\omega_1), P(\omega_2)$

(3) 两类条件概率密度函数分别为$P(x|\omega_1)$，$P(x|\omega_2)$

对于两类别分类问题，已知先验概率$\mathrm{P}(\omega_1), \mathrm{P}(\omega_2)$及条件概率密度函数$P(x|\omega_1), P(x|\omega_2)$可以得出某样本属于各类别的概率，即后验概率：

$$P(\omega_i|x) = \frac{P(x|\omega_i)P(\omega_i)}{P(x)}, i = 1,2$$

贝叶斯的分类规则最大后验概率准可以描述为：

$$P(\omega_1|x) > P(\omega_2|x),\ 则 x \in \omega_1$$
$$P(\omega_1|x) < P(\omega_2|x),\ 则 x \in \omega_2$$

结合后验概率的计算公式，可得：

$$P(x|\omega_1)P(\omega_1) > P(x|\omega_2)P(\omega_2),\ 则 x \in \omega_1$$
$$P(x|\omega_1)P(\omega_1) < P(x|\omega_2)P(\omega_2),\ 则 x \in \omega_2$$

2、样本服从正态分布

多变量正态分布也称为多变量高斯分布。用特征向量$x = [x_1\ x_2\ x_3 \dots\ x_N]^T$来表示多个变量，假设本实验所使用的数据中各类数据服从正态分布，则类概率密度函数为：

$$P(x|\omega_i) = \frac{1}{(2\pi)^{N/2}|C_i|^{1/2}} exp\left[-\frac{1}{2}(x-\mu_i)^T {C_i}^{-1}(x-\mu_i)\right] \quad i = 1,2$$

其中特征向量$x = [x_1\ x_2\ x_3\ \dots\ x_N]^T$是 N 维列向量，$\mu_i = [\mu_1\ \mu_2\ \mu_3 \dots\ \mu_N]^T$是 N 维均值向量，$C_i$是N × N的协方差矩阵，$|C_i|$是其行列式。本实验的 IRIS 数据有 4 个特征值，即$N = 4$。此外，每类的均值向量和协方差矩阵都可由样本计算得到：

$$\mu_i = E_i[x] \qquad C_i = E_i\{(x-\mu)(x-\mu)_i^T\}$$

其中$E_i[x]$表示对类别属于ω_i的x作数学期望运算。

三、实验过程

1. 实验设定

（1）选取数据方式与选取样本数据量

数据集包含 150 个数据样本，分为 3 类，每类 50 个数据。对于两两分类问题，选取数据的方式是，首先将两类数据（100 个）进行 shuffle 随机打乱，由于类别数量是均衡的，可以指定一个 a 值（split），从打乱后的数据中取出 a%的数据作为测试集，其他数据作为训练集。后面会讨论不同 a 值情况下，分类准确率是如何变化的。

（2）先验概率与条件概率设定

对于 Iris 鸢尾花数据两两分类问题，设定ω_1, ω_2表示这两类。每一类的先验概率是未知的，但是可以从训练集中估算出来。如果N是训练样本的总数，其中有N_1, N_2个样本分别属于N_1, N_2，则相应的先验概率为$P(\omega_1) = N_1/N,\ P(\omega_2) = N_2/N$。此外，类条件概率密度函数$P(x|\omega_i)$也是未知的，但是由于模型假定了是服从多维高斯分布的，可采用训练数据进行概率建模，对训练集中的每一类样本，分别求得 4 维（4 个特征）高斯分布的均值矩阵和协方差矩阵。

2. 实验过程

实验数据：IRIS 数据集

实验假设：各类数据服从正态分布

实验方法：最小错误率贝叶斯决策

编程语言：python

(1) 重要函数

```
import numpy as np
from sklearn.datasets import load_iris
from collections import  Counter
from numpy import *

def load_data(Cla_1=1 ,Cla_2=2):
    data_size = 150
    data = load_iris()
    #鸢尾花的四个特征
    # 'sepal length (cm)'
    # 'sepal width (cm)'
    # 'petal length (cm)'
    # 'petal width (cm)'
    data_feature = data.feature_names
    print("Features:")
    print(data_feature)
    Iris_data = data.data
    print("Datas:")
    print(Iris_data)

    #鸢尾花的三个类别
    #'setosa' 1
    # 'versicolor' 2
    # 'virginica'  3
    target_names =data.target_names
    print("Target_names:")
    print(target_names)
    Iris_label=data.target
    print("Targets:")
    print(Iris_label)
    #
```

```
def get_train_test_split(data, label, split = 0.4):
    ind = np.random.permutation(len(data))
    split_ind = int(len(data)*split)
    data = data[ind]
    label = label[ind]
    train_data = data[:split_ind]
    train_label = label[:split_ind]
    test_data = data[split_ind:]
    test_label = label[split_ind:]
    return train_data, train_label, test_data, test_label


def class_pior_prob(train_label):

    nb_cla1 = sum(train_label == 0)
    nb_cla2 = sum(train_label == 1)
    return nb_cla1/(nb_cla1+nb_cla2), nb_cla2/(nb_cla1+nb_cla2)

def coVariance(X):  # 数据的每一行是一个样本，每一列是一个特征
    ro, cl = X.shape
    row_mean = np.mean(X, axis=0)
    X_Mean = np.zeros_like(X)
    X_Mean[:] = row_mean  # 把向量赋值给每一行
    X_Minus = X - X_Mean
    covarMatrix = np.zeros((cl, cl))
    for i in range(cl):
        for j in range(cl):
            covarMatrix[i, j] = (X_Minus[:, i].dot(X_Minus[:, j].T)) / (ro - 1)
    return covarMatrix
```

```
def gussion_paramters(train_data_cla):
    mean  =  np.mean(train_data_cla, axis=0)
    cov = coVariance(train_data_cla)
    return  mean, cov

def prob_gussion(x, mean, cov):
        cov = mat(cov)
        r = mat([x[0] - mean[0], x[1] - mean[1], x[2]-mean[2], x[3]-mean[3]])
        multi = r * cov.I * r.T
        multi = float(multi)  # 1乘1矩阵取内容
        k = exp(-multi / 2)  # .I求逆,.T转置
        k /= 2 * math.pi * linalg.det(cov) ** (1 / 2)  # linalg.det求行列式的值
        return k

def post_prob(x, mean1, cov1, mean2, cov2, pior_prob_1, pior_prob_2):
    k1 = prob_gussion(x, mean1, cov1) * pior_prob_1
    k2 = prob_gussion(x, mean2, cov2) * pior_prob_2
    return k1 / (k1 + k2), k2 / (k1 + k2)
```

(2)　加载数据

这里选取'setosa'和 'versicolor'两类数据。

```
if __name__ == '__main__':
    Cla1, Cla2 = load_data(1, 2)
    data = np.concatenate((Cla1,Cla2), axis=0)
    label = np.zeros((100,))
    label[50:100] = 1
```

```
Features:
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
Datas:
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]
 [5.4 3.7 1.5 0.2]
 [4.8 3.4 1.6 0.2]
 [4.8 3.  1.4 0.1]
 [4.3 3.  1.1 0.1]
 [5.8 4.  1.2 0.2]
 [5.7 4.4 1.5 0.4]
```

(3) 划分数据

设置 split=0.8，取 80%的数据作为训练集，20%作为测试集（训练集大小为 80，测试集大小为 20），并计算先验概率，如下分别是 0.5625 和 0.4375。

```
train_data, train_label, test_data, test_label = get_train_test_split(data, label, split=0.8)
pior_prob_1, pior_prob_2 = class_pior_prob(train_label)
print(train_data.shape, train_label.shape, test_data.shape, test_label.shape)
print(pior_prob_1, pior_prob_2)
```

```
(80, 4) (80,) (20, 4) (20,)
0.5625 0.4375
```

(4) 按类别计算 4 维正态分布的参数

```
train_data_cla1 = []
train_data_cla2 = []
for i in range(len(train_data)):
    if train_label[i] == 0:
        train_data_cla1.append(train_data[i])
    else:
        train_data_cla2.append(train_data[i])
train_data_cla1 = np.array(train_data_cla1)
train_data_cla2 = np.array(train_data_cla2)
```

```
mu1, cov1 = gussion_paramters(train_data_cla1)
mu2, cov2 = gussion_paramters(train_data_cla2)
print(mu1)
print(cov1)
print(mu2)
print(cov2 )
```

```
[4.99111111 3.42444444 1.45777778 0.24       ]
[[0.1290101  0.10522222 0.01620707 0.00877273]
 [0.10522222 0.15007071 0.01082828 0.01036364]
 [0.01620707 0.01082828 0.03158586 0.00536364]
 [0.00877273 0.01036364 0.00536364 0.01154545]]
[5.96        2.76        4.28        1.32571429]
[[0.32188235 0.12511765 0.20476471 0.07047059]
 [0.12511765 0.11364706 0.108      0.054      ]
 [0.20476471 0.108      0.21929412 0.07847059]
 [0.07047059 0.054      0.07847059 0.04078992]]
```

(5) 在测试集上进行测试

对于测试集中的 20 个样本全部正确分类，正确率为 1。

```
    count = 0
    for i in range(len(test_data)):
        post_prob_1, post_prob_2 = post_prob(test_data[i], mu1, cov1, mu2, cov2, pior_prob_1, pior_prob_2)
        if post_prob_1 > post_prob_2 :
            pred = 0
        else:
            pred = 1
        if pred == test_label[i]:
            count += 1
    acc = count / len(test_label)
    print(count)
    print(acc)
```

```
20
1.0
```

四、实验结果与分析

1. 三类数据两两分类

设置不同的 split 值以及不同类别进行两类分类，由于训练样本是随机抽取的，每次实验结果都不相同，所以每次实验都进行 500 次，最后求得平均正确率，实验结果如下：

类别 / split	(ω_1,ω_2)	(ω_2,ω_3)	(ω_1,ω_3)
0.8	1.0	0.9614	1.0
0.6	1.0	0.9578	1.0
0.4	0.9998	0.94826	1.0
0.2	0.993675	0.890125	0.99125

分析：

（1）从实验中可以看出随着 split 值的增大，也就是样本数的增加，模型的性能也会有相应的提升，训练样本数量较少时模型的分类正确率较低，但是总的正确率都在一个很高的水平线上。

（2）在不同划分的情况下，(ω_1,ω_2)和(ω_1,ω_3)的分类效果比(ω_2,ω_3)更好，错误分类在(ω_2,ω_3)中更多，这可以得到解释：通过对这三类样本的均值及协方差的分析可以发现ω_1类的均值距离ω_2类和ω_3类的均值比较远，而ω_2类和ω_3类的均值是比较接近的，同时从ω_1类的协方差矩阵中可以看出ω_1类样本方差是比较小的，说明数据分布比较集中，所以即使训练过程抽取的样本比较少，第一类仍然可以和其它两类分开。

2. 最小风险贝叶斯决策

在以上代码和实验的基础上，继续进行最小风险贝叶斯决策实验。构建以下的决策表，按最小风险贝叶斯决策进行分类。

决策	状态	
	ω_1	ω_2
a_1	λ_{11}	λ_{12}
a_2	λ_{21}	λ_{22}

在上面的决策表中，指定$\lambda_{11}=\lambda_{22}=0$，观察$\lambda_{12}$和$\lambda_{21}$在不同取值情况下，模型分类正确率、$\omega_1$类的错误率和$\omega_2$类的错误率如何变化，同样为了减少随机误差，每次实验都进行 500 次，最后求得各平均值，实验结果如下：

指标 / $(\lambda_{21},\lambda_{12})$	正确率	错误率 1	错误率 2
（1，1）	0.88855	0.13368	0.08234
（1，10）	0.877975	0.17754	0.06077
（1，100）	0.8379	0.28034	0.03623
（10，1）	0.875575	0.09583	0.14605
（100，1）	0.856425	0.07549	0.20412

分析：

（1）本实验的参数定为 *split=0.2*，(ω_2,ω_3)，是因为由上一个实验结论可知，*split=0.2* 时，模型的分类能力较低，而且(ω_2,ω_3)两类相对来说更不易区分，这样的设置有利于观察在最小风险贝叶斯决策中，决策损失值对最后分类结果造成的影响。

（2）从实验结果中可以看出，当$(\lambda_{21},\lambda_{12})=(1,1)$时，最小风险贝叶斯决策就是最小错误率贝叶斯决策，可以看到此时的正确率和第一个表中的结果基本一致；

（3）随着λ_{12}值的增大，可以看到，模型的分类正确率在逐渐下降，这是因为λ_{12}值的增大表示 2 类(ω_3)被误分为 1 类(ω_2)时的损失值或风险增大，模型会倾向于把本属于 1 类(ω_2)的样本误判为 2 类(ω_3)，于是造成 1 类的错误率逐渐增大，2 类的错误率逐渐减小，同时正确率逐渐减小。

（4）随着λ_{21}值的增大，同样也可以观察到相同的现象，即 2 类的错误率逐渐增大，1 类的错误率逐渐减小，同时正确率逐渐减小。

五、实验结论

本次实验主要采用 Iris 鸢尾花数据集进行了最小错误率和最小风险贝叶斯决策的实验，实验内容主要包括：贝叶斯判别原理、数据选取方式、先验概率与类条件概率的设定、主要实验代码、实验结果与分析。通过此次实验，进一步加深了我对贝叶斯判别原则、最小风险判别原则以及正态分布中均值向量与协方差矩阵的估计的理解，也提高了自己动手编程做实验并对实验结果进行分析的能力。