模式识别实验(一)实验报告

学生信息

- 姓名-学号-班级
- 李欢欢-2020904302-2020240402
- 刘易行-2020905896-2020240401

实验题目

2.参考上机练习1（b），并考虑将上面表格中的10个样本点进行分类的问题，假设分布是正态的。

（a）假设前面两个类别的先验概率相等($P(\omega_1) = P(\omega_2) = 1/2$，且$P(\omega_3) = 0$)，仅利用$x_1$特征值为这两类判别设计一个分类器。

（b）确定样本的经验训练误差，即误分点的百分比。

（c）利用Bhattacharyya界来界定对该分布所产生的新模式进行分类会产生的误差。

（d）现在利用两个特征值x_1和x_2，重复以上各步。

（e）利用所有3个特征值重复以上各步。

（f）讨论所得的结论。特别是，对于一有限的数据集，是否有可能在更高的数据维数下经验误差会增加？ 可能

问题回答

(a)分类器是如何设计的?

1. 数据的读取
   - 我们将书本上的样本点数据存入excel表格进行处理，并通过xlrd库进行读取
   - 表格设计为四列30行，从左至右列的含义依次为x1,x2,x3,所属类别wi(i=1,2,3)
   -

	A	B	C	D
1	-5.01	-8.12	-3.68	1
2	-0.91	-0.18	-0.05	2
3	5.35	2.26	8.13	3
4	-5.43	-3.48	-3.54	1
5	1.3	-2.06	-3.53	2
6	5.12	3.22	-2.66	3
7	1.08	-5.52	1.66	1
8	-7.75	-4.54	-0.95	2
9	-1.34	-5.31	-9.87	3
10	0.86	-3.78	-4.11	1
11	-5.47	0.5	3.92	2
12	4.48	3.42	5.19	3
13	-2.67	0.63	7.39	1
14	6.14	5.72	-4.85	2
15	7.11	2.39	9.21	3
16	4.94	3.29	2.08	1
17	3.6	1.26	4.36	2
18	7.17	4.33	-0.98	3
19	-2.51	2.09	-2.59	1
20	5.37	-4.63	-3.65	2
21	5.75	3.97	6.65	3
22	-2.25	-2.13	-6.94	1
23	7.18	1.46	-6.66	2
24	0.77	0.27	2.41	3
25	5.56	2.86	-2.26	1
26	-7.39	1.17	6.3	2
27	0.9	-0.43	-8.71	3
28	1.03	-3.33	4.33	1
29	-7.5	-6.32	-0.31	2

```
row value=[-5.01, -8.12, -3.68, 1.0]
row value=[-0.91, -0.18, -0.05, 2.0]
row value=[5.35, 2.26, 8.13, 3.0]
row value=[-5.43, -3.48, -3.54, 1.0]
row value=[1.3, -2.06, -3.53, 2.0]
row value=[5.12, 3.22, -2.66, 3.0]
row value=[1.08, -5.52, 1.66, 1.0]
row value=[-7.75, -4.54, -0.95, 2.0]
row value=[-1.34, -5.31, -9.87, 3.0]
row value=[0.86, -3.78, -4.11, 1.0]
row value=[-5.47, 0.5, 3.92, 2.0]
row value=[4.48, 3.42, 5.19, 3.0]
row value=[-2.67, 0.63, 7.39, 1.0]
row value=[6.14, 5.72, -4.85, 2.0]
row value=[7.11, 2.39, 9.21, 3.0]
row value=[4.94, 3.29, 2.08, 1.0]
row value=[3.6, 1.26, 4.36, 2.0]
row value=[7.17, 4.33, -0.98, 3.0]
row value=[-2.51, 2.09, -2.59, 1.0]
```

2. 分类器设计

- 由题目信息我们得知`p(w1)=p(w2)=0.5`且`p(w3)=0`，那么就说明了在考虑数据时无需考虑第三类别的数据，只需要考虑w1和w2对应的数据，那么在设计分类器时，根据`公式49`，依次使用numpy计算出公式的每一项，矩阵的均值，矩阵的协方差，然后带入求解比较判别函数的值即可解决二分类问题
- 例如在只考虑x1单维度特征值的情况下，我们依次计算出公式所需数据如下：

```
i=0
第1个类别对应的参数
xi=[[-5.01], [-5.43], [1.08], [0.86], [-2.67], [4.94], [-2.51], [-2.25], [5.56],
[1.03]]
ui=[-0.44]
sigmali=14.38051111111111
----------------
```

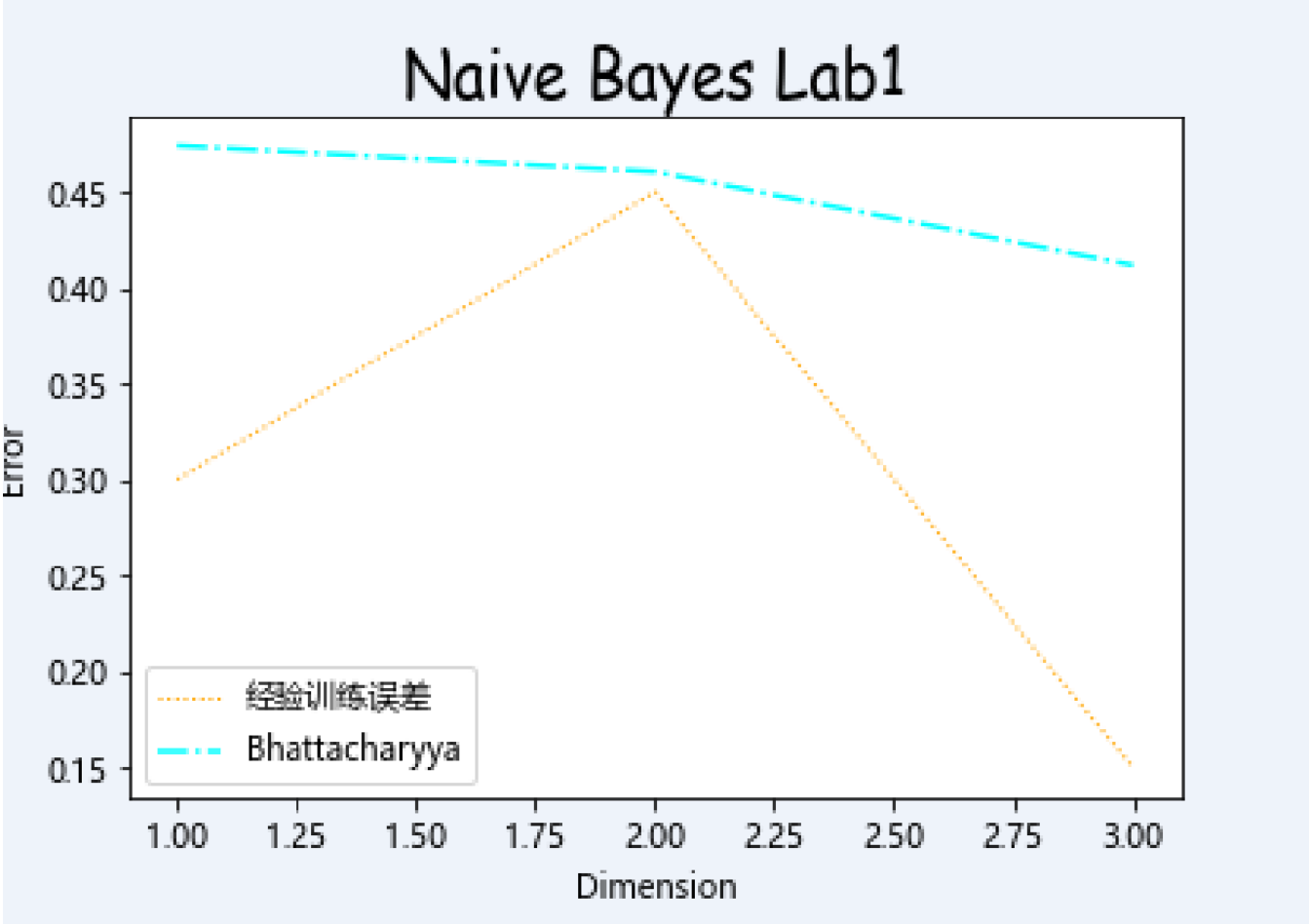
```
i=1
第2个类别对应的参数
xi=[[-0.91], [1.3], [-7.75], [-5.47], [6.14], [3.6], [5.37], [7.18], [-7.39], [-
7.5]]
ui=[-0.543]
sigmali=36.82933444444445
```

- 哪一类别的判别函数值大，那么则将其归为哪一类，并与表格第四列对应的标签比对，判定分类结果是否正确，并将其统计到`count_false`和`count_true`

```
第1个样本点
x =  [-5.01]
g1(x) =[[-3.67117557]] g2(x) =[[-3.68613244]]
w1 True
第1个样本点
x =  [-0.91]
g1(x) =[[-2.95270319]] g2(x) =[[-3.4170616]]
w1 False
第2个样本点
x =  [-5.43]
g1(x) =[[-3.81078118]] g2(x) =[[-3.73946873]]
w2 False
```

- $$g_i(x) = -\frac{1}{2}(x-\mu_i)^t \sum_i^{-1}(x-\mu_i) - \frac{d}{2}\ln 2\Pi - \frac{1}{2}\ln|\sum_i| + \ln p(w_i)$$

(b、c)分别选取一个（两个、三个）特征的情况下的经验训练误差以及Bhattacharyya界

- 经验训练误差与Bhattacharyya界
  - 一个特征
    - 经验训练误差：0.3
      Bhattacharyya=[[0.47399944]]
  - 两个特征
    - 经验训练误差：0.45
      Bhattacharyya=[[0.46046616]]
  - 三个特征
    - 经验训练误差：0.15
      Bhattacharyya=[[0.41192563]]
- 图像分析（放到了同一张图中）
  - 

(f)对比、分析、总结

- 对比
  - 通过实验结果我们发现 经验训练误差分别为`[0.3 , 0.45, 0.15]`,Bhattacharyya界为`[0.47399944, 0.46046616, 0.41192563]`
- 分析
  - 我们发现，对于一有限数据集，有可能在更高的数据维数下训练误差会增加，原因是否有可能是各个特征值之间彼此并不独立，存在着某种联系，导致着经验训练误差出现了波动
- 总结
  - 此次实验是本学期模式识别课程的第一次实验，在此前的课程中，受限于没有纸质书本以及线上课程的影响，对于模式识别这门课的理解和掌握并不算很到位，网课效果也不是很好，但是通过本次实验，我们小组在进一步巩固Python科学计算能力与编码水平的同时，对于模式识别的含义以及分类问题和贝叶斯决策论有了更深一步的认识，通过实践

操作去理解原理并掌握的感觉很棒，印象也很深，尽管刚一拿到本实验题目时我们两人都陷入了一筹莫展的窘境，但随着我们二人之间的大量研讨与学习，我们逐步对于实验问题以及要解决的问题有了清晰的认识，将大问题拆解为一个个的小模块，并开始策划分工，组织分模块编码与测试，最终圆满完成了本次实验，并于9月29日课程中向老师进行了初步的演示，在老师的指导下，证实了我们的思路的正确性以及学习成果，并于10月4日最终完成此次实验，收益颇丰。

实验分工

- 注：本次实验全过程为小组两人课余时间以及国庆假期内自习时一并研讨完成
- 李欢欢
  - 贝叶斯决策论原理学习、研讨
  - 实验整体思路敲定、模块划分、分类器的设计
  - 数据的读取方式以及读取数据相关代码、excel表格的设计以及后续的处理操作
  - numpy库的学习，编码实现了计算均值函数`get_u()`、计算协方差函数`get_sigmal(x)`、数据读取函数`get_data()`、分类器函数`ClassifyTest(di)`
  - Debug，解决了数据类型不统一的重大问题，将数据类型统一为`np.matrix`
  - 实验报告的撰写
- 刘易行
  - 贝叶斯决策论原理学习、研讨
  - 实验整体思路敲定、模块划分、分类器的设计
  - matplotlib库的学习使用，编码实现、绘图函数`draw()`
  - Bhattacharyya的计算函数`get_Bhattacharyya(u1,u2,sigmal1,sigmal2,pw1,pw2,d)`、、判别函数`get_gi()`、,分类器函数`ClassifyTest(di)`
  - Debug，优化代码，解决了多特征值计算时的np.shape()不统一的问题
  - 注释撰写
  - 实验报告的撰写

实验源代码

- 编码过程使用Jupyter编写
- 后统一为一个python文件

```
import numpy as np  # 计算数据
import matplotlib.pyplot as plt  # 绘制图线
import xlrd  # 读取数据


# 均值计算函数 计算出u
def get_u(x):
    u = np.mean(x, axis=0)  # 计算每一列的均值u
    return u


# 计算协方差矩阵 sigmal
def get_sigmal(x):
    sigmal = np.cov(np.mat(x).T)
    return sigmal


# 根据式(49)计算判别函数的值
def get_gi(x, u, sigmal, pw, d):
    # 将各个数据转换为np.array矩阵形式
    x = np.mat(x)
    u = np.mat(u)
    sigmal = np.mat(sigmal)
```

```
    pw = np.mat(pw)

    # 根据公式49计算判别函数的值
    g = -0.5 * (x - u) * sigmal.I * (x - u).T - d / 2 * np.log(2 * np.pi) - 0.5 * np.log(
        np.linalg.det(sigmal)) + np.log(pw)
    return g


def get_data():
    x = []  # 存放最终的数据
    data = xlrd.open_workbook("data.xls")  # 加载表格文件
    table = data.sheets()[0]  # 选取工作簿 sheet1
    rows = table.nrows  # 表格的总行数
    for i in range(0, rows):  # 遍历每一行读取数据
        row_value = table.row_values(i)
        print("row value={}".format(row_value))
        if row_value[3] != 3:  # x3不需要添加进去因为p(x3)=0
            x.append(row_value)
    print("----------------------")
    print("------数据读取完毕------")
    print("----------------------")
    print("")
    print("")
    return x


def get_Bhattacharyya(u1, u2, sigmal1, sigmal2, pw1, pw2, d):
    # 计算Bhattacharyya界的值

    # 数据类型归一为np.matrix
    sigmal1 = np.mat(sigmal1)
    sigmal2 = np.mat(sigmal2)
    u1 = np.mat(u1)
    u2 = np.mat(u2)

    # 数据验证 便于纠错
    print("-------Bhattacharyya参数验证--------")
    print("parameter and their type:")
    print("u1:", u1, "type:", type(u1))
    print("u2:", u2, "type:", type(u2))
    print("sigmal1:", sigmal1, "type:", type(sigmal1))
    print("sigmal2:", sigmal2, "type:", type(sigmal2))
    print("pw1:", pw1, "type:", type(pw1))
    print("pw2:", pw2, "type:", type(pw2))
    print("d:", d, "type:", type(d))
    if d == 1:
        k_half = (1 / 8) * (u2 - u1).T * ((sigmal1 + sigmal2) / 2).I * (u2 - u1) + (1 / 2) * np.log(
            np.linalg.det((sigmal1 + sigmal2) / 2) / np.sqrt(np.linalg.det(sigmal1) *
np.linalg.det(sigmal2)))
    else:
        k_half = (1 / 8) * (u2 - u1) * ((sigmal1 + sigmal2) / 2).I * (u2 - u1).T + (1 / 2) * np.log(
            np.linalg.det((sigmal1 + sigmal2) / 2) / np.sqrt(np.linalg.det(sigmal1) *
np.linalg.det(sigmal2)))

    # 由公式(76)算出Perror
    Perror = np.sqrt(np.array(pw1 * pw2)) * np.exp(-k_half)
    print("-------Bhattacharyya计算完毕--------")
    print("-----------------------------------")
    print("")
    print("")
```

```
    return Perror


def ClassifyTest(di):
    # 分类器函数 参数di为特征维度值
    d = di  # 特征维度
    g = [0, 0, 0]
    pw = (0.5, 0.5, 0)  # 先验概率
    u = [0, 0, 0]  # 各个维度的均值
    sigmal = [0, 0, 0]  # 各个维度的协方差矩阵
    count_true = 0
    count_false = 0
    num = 0
    data = get_data()  # 读取数据

    #     print("类型",type(np.mat(sigmal[0])))
    #     print(type(sigmal[0]))
    #     print("sigmal1",sigmal[0])
    for i in range(len(pw) - 1):
        xi = [x[:d] for x in filter(lambda x: x[3] == i + 1, data)]  # 获得每一个类别的对应特征
        u[i] = get_u(xi)
        sigmal[i] = get_sigmal(xi)
        print("i={}".format(i))
        print("第{}个特征对应的参数".format(i + 1))
        print("xi={}".format(xi))
        print("ui={}".format(u[i]))
        print("sigmali={}".format(sigmal[i]))
        print("----------------")
        print("")
        print("")

    Bbound = get_Bhattacharyya(u[0], u[1], sigmal[0], sigmal[1], pw[0], pw[1], d)
    for i in range(len(data)):
        if i % 2 == 0:
            num += 1
        print("第{}个样本点".format(num))
        x = data[i][:d]
        w = data[i][3]
        print("x = ", x)
        g[0] = get_gi(x, u[0], sigmal[0], pw[0], d)
        g[1] = get_gi(x, u[1], sigmal[1], pw[1], d)
        print("g1(x) ={}".format(g[0]), "g2(x) ={}".format(g[1]))
        if g[0] > g[1]:  # 根据判别函数的值去进行比对分类
            flag = w == 1
            print("w1", flag)
        else:
            flag = w == 2
            print("w2", flag)
        if flag:
            count_true += 1
        else:
            count_false += 1
    error_rate = (count_false) / (count_true + count_false)
    print("经验训练误差：{}".format(error_rate))
    print("Bhattacharyya={}".format(Bbound))
    print("")
    print("")
    return error_rate, Bbound
```

```
def draw(data1, data2):
    plt.rcParams['font.sans-serif'] = ['Microsoft YaHei']  # 显示中文
    plt.title("Naive Bayes Lab1", fontdict={'fontname': 'comic sans MS', 'fontsize': 20})
    plt.xlabel('Dimension')
    plt.ylabel('Error')
    x = range(1, 4)

    # plt.figure(figsize=(20, 8), dpi=80)

    plt.plot(x, data1, label='经验训练误差', color='orange', linestyle=":", linewidth=1)
    plt.plot(x, data2, label='Bhattacharyya', color='cyan', linestyle="-.")
    plt.legend()
    plt.savefig("Bayes.png")
    plt.show()


def main():
    print("一维度二分类问题：")
    oneData = ClassifyTest(1)
    print("二维度二分类问题：")
    twoData = ClassifyTest(2)
    print("三维度二分类问题：")
    threeData = ClassifyTest(3)

    errorVec = np.array([oneData[0], twoData[0], threeData[0]])
    BboundVec = np.array([float(oneData[1]), float(twoData[1]), float(threeData[1])])

    draw(errorVec, BboundVec)


main()
```

实验结果

数据读取

```
一维度二分类问题：
row value=[-5.01, -8.12, -3.68, 1.0]
row value=[-0.91, -0.18, -0.05, 2.0]
row value=[5.35, 2.26, 8.13, 3.0]
row value=[-5.43, -3.48, -3.54, 1.0]
row value=[1.3, -2.06, -3.53, 2.0]
row value=[5.12, 3.22, -2.66, 3.0]
row value=[1.08, -5.52, 1.66, 1.0]
row value=[-7.75, -4.54, -0.95, 2.0]
row value=[-1.34, -5.31, -9.87, 3.0]
row value=[0.86, -3.78, -4.11, 1.0]
row value=[-5.47, 0.5, 3.92, 2.0]
row value=[4.48, 3.42, 5.19, 3.0]
row value=[-2.67, 0.63, 7.39, 1.0]
row value=[6.14, 5.72, -4.85, 2.0]
row value=[7.11, 2.39, 9.21, 3.0]
row value=[4.94, 3.29, 2.08, 1.0]
row value=[3.6, 1.26, 4.36, 2.0]
row value=[7.17, 4.33, -0.98, 3.0]
row value=[-2.51, 2.09, -2.59, 1.0]
row value=[5.37, -4.63, -3.65, 2.0]
row value=[5.75, 3.97, 6.65, 3.0]
row value=[-2.25, -2.13, -6.94, 1.0]
row value=[7.18, 1.46, -6.66, 2.0]
row value=[0.77, 0.27, 2.41, 3.0]
row value=[5.56, 2.86, -2.26, 1.0]
row value=[-7.39, 1.17, 6.3, 2.0]
row value=[0.9, -0.43, -8.71, 3.0]
row value=[1.03, -3.33, 4.33, 1.0]
row value=[-7.5, -6.32, -0.31, 2.0]
row value=[3.52, -0.36, 6.43, 3.0]
————————————————————
————数据读取完毕————
————————————————————
```

一个特征维度

```
i=0
第1个特征对应的参数
xi=[[-5.01], [-5.43], [1.08], [0.86], [-2.67], [4.94], [-2.51], [-2.25], [5.56], [1.03]]
ui=[-0.44]
sigmali=14.38051111111111
————————————


i=1
第2个特征对应的参数
xi=[[-0.91], [1.3], [-7.75], [-5.47], [6.14], [3.6], [5.37], [7.18], [-7.39], [-7.5]]
ui=[-0.543]
sigmali=36.82933444444445
————————————


————Bhattacharyya参数验证————
parameter and their type:
u1: [[-0.44]] type: <class 'numpy.matrix'>
u2: [[-0.543]] type: <class 'numpy.matrix'>
sigma1: [[14.38051111]] type: <class 'numpy.matrix'>
sigma2: [[36.82933444]] type: <class 'numpy.matrix'>
pw1: 0.5 type: <class 'float'>
pw2: 0.5 type: <class 'float'>
d: 1 type: <class 'int'>
————Bhattacharyya计算完毕————
————————————————————————————
```

lab1 (1)

```
g1(x) =[[-3.51251292]] g2(x) =[[-3.64825991]]
w1 False
第7个样本点
x =  [-2.51]
g1(x) =[[-3.09400554]] g2(x) =[[-3.46776032]]
w1 True
第7个样本点
x =  [5.37]
g1(x) =[[-4.11869791]] g2(x) =[[-3.88990316]]
w2 True
第8个样本点
x =  [-2.25]
g1(x) =[[-3.0589303]] g2(x) =[[-3.45479185]]
w1 True
第8个样本点
x =  [7.18]
g1(x) =[[-4.96387997]] g2(x) =[[-4.22497791]]
w2 True
第9个样本点
x =  [5.56]
g1(x) =[[-4.1967167]] g2(x) =[[-3.92089802]]
w2 False
第9个样本点
x =  [-7.39]
g1(x) =[[-4.62446575]] g2(x) =[[-4.05170136]]
w2 True
第10个样本点
x =  [1.03]
g1(x) =[[-3.0201556]] g2(x) =[[-3.44882487]]
w1 True
第10个样本点
x =  [-7.5]
g1(x) =[[-4.67804869]] g2(x) =[[-4.07231591]]
w2 True
经验训练误差: 0.3
Bhattacharyya=[[0.47399944]]
```

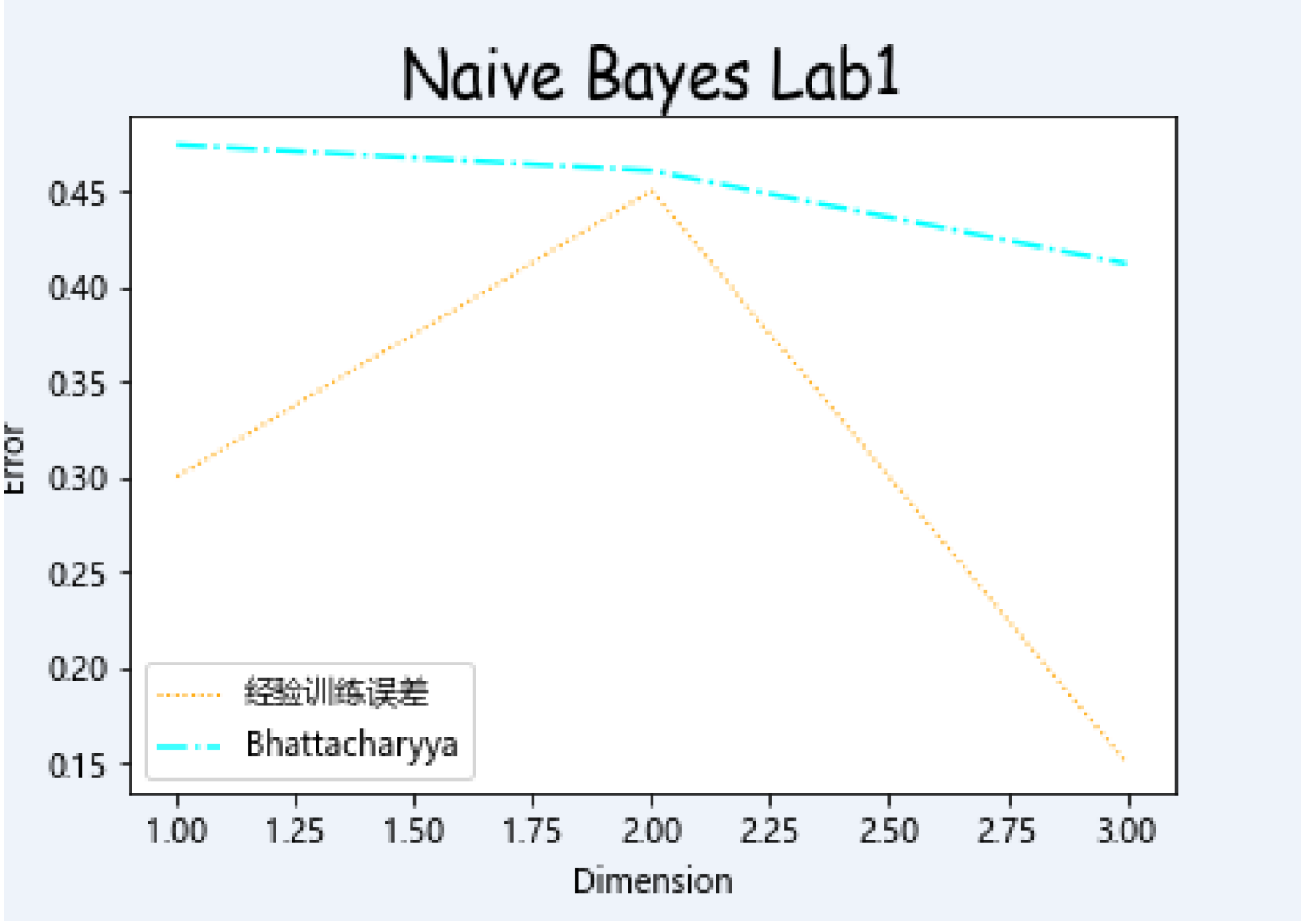
两个特征维度

```
lab1 (1)
w2 False
第6个样本点
x =  [3.6, 1.26]
g1(x) =[[-5.6415364]] g2(x) =[[-5.77979206]]
w1 False
第7个样本点
x =  [-2.51, 2.09]
g1(x) =[[-6.35354211]] g2(x) =[[-6.10819916]]
w2 False
第7个样本点
x =  [5.37, -4.63]
g1(x) =[[-7.92135535]] g2(x) =[[-7.41278384]]
w2 True
第8个样本点
x =  [-2.25, -2.13]
g1(x) =[[-5.17023291]] g2(x) =[[-5.58685586]]
w1 True
第8个样本点
x =  [7.18, 1.46]
g1(x) =[[-7.09466434]] g2(x) =[[-6.31866672]]
w2 True
第9个样本点
x =  [5.56, 2.86]
g1(x) =[[-6.38464133]] g2(x) =[[-6.19887214]]
w2 False
第9个样本点
x =  [-7.39, 1.17]
g1(x) =[[-8.81662903]] g2(x) =[[-6.83009421]]
w2 True
第10个样本点
x =  [1.03, -3.33]
g1(x) =[[-5.38183335]] g2(x) =[[-5.9701421]]
w1 True
第10个样本点
x =  [-7.5, -6.32]
g1(x) =[[-6.80285064]] g2(x) =[[-6.8097312]]
w1 False
经验训练误差: 0.45
Bhattacharyya=[[0.46046616]]
```

三个特征维度

```
lab1 (1)
w1 False
第7个样本点
x =  [-2.51, 2.09, -2.59]
g1(x) =[[-8.83800477]] g2(x) =[[-9.25207517]]
w1 True
第7个样本点
x =  [5.37, -4.63, -3.65]
g1(x) =[[-10.64358362]] g2(x) =[[-9.66155324]]
w2 True
第8个样本点
x =  [-2.25, -2.13, -6.94]
g1(x) =[[-8.44677649]] g2(x) =[[-10.14781384]]
w1 True
第8个样本点
x =  [7.18, 1.46, -6.66]
g1(x) =[[-11.1907397]] g2(x) =[[-8.75419976]]
w2 True
第9个样本点
x =  [5.56, 2.86, -2.26]
g1(x) =[[-9.08010492]] g2(x) =[[-8.21150281]]
w2 False
第9个样本点
x =  [-7.39, 1.17, 6.3]
g1(x) =[[-12.93266582]] g2(x) =[[-9.07730019]]
w2 True
第10个样本点
x =  [1.03, -3.33, 4.33]
g1(x) =[[-8.45356356]] g2(x) =[[-10.70022425]]
w1 True
第10个样本点
x =  [-7.5, -6.32, -0.31]
g1(x) =[[-9.3605496]] g2(x) =[[-8.89743559]]
w2 True
经验训练误差：0.15
Bhattacharyya=[[0.41192563]]
```

实验图表分析


Naive Bayes Lab1
0.45
0.40
0.35
0.30
0.25
0.20
0.15
error
经验训练误差
Bhattacharyya
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Dimension

请老师批评指正！