统计分析第五章作业

16337183 孟衍璋

实验要求

汽车评价数据库一共有6个变量，即**buying**, **maint**, **doors**, **persons**, **lug boot**, **safety** (在本次作业中，不考虑第6个变量safety,否则协方差求逆有bug,我们简化之，即只要前五个变量).每个变量经过了等级划分。数据的类别为4类，即**unacc**, **acc**, **good**, **vgood**。问：

1. 假设4个类别总体方差都相等，请根据训练数据(train.txt)，用马氏距离预测出测试集的类别(test.txt)。
2. 假设4个类别总体方差不一致，请根据训练数据(train.txt)，用马氏距离预测出测试集的类别(test.txt)。

相关数据

变量等级：

$$buying : vhigh, high, med, low.$$

$$maint : vhigh, high, med, low.$$

$$doors : 2, 3, 4, 5more.$$

$$persons : 2, 4, more.$$

$$lug\_boot : small, med, big.$$

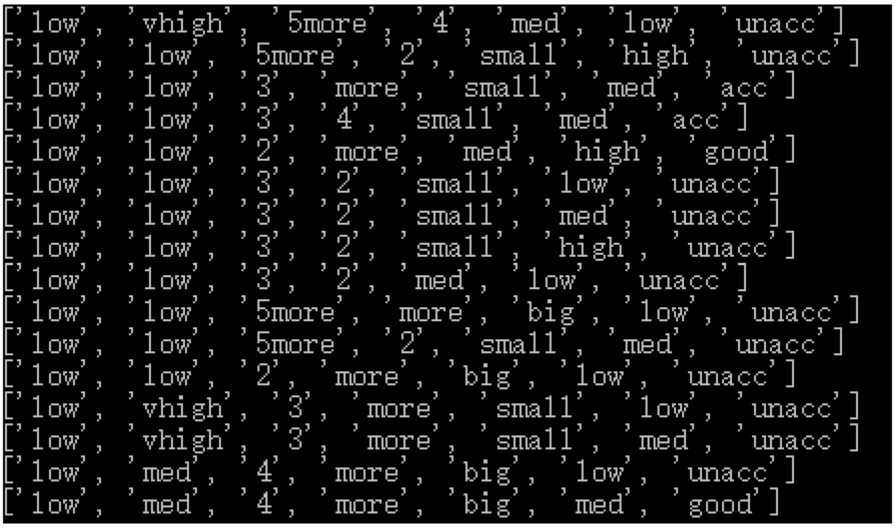
$$safety : low, med, high.$$

实验步骤

读取训练数据文件

```
with open('train.txt','r') as f:
    data = f.readlines()
    # 将数据保存在index中
    index = []
    for line in data:
        index.append(line.replace('\n', '').split(','))
```

将所有指标都存入列表**index**中，其中一部分显示如下：



```
['low', 'vhigh', '5more', '4', 'med', 'low', 'unacc']
['low', 'low', '5more', '2', 'small', 'high', 'unacc']
['low', 'low', '3', 'more', 'small', 'med', 'acc']
['low', 'low', '3', '4', 'small', 'med', 'acc']
['low', 'low', '2', 'more', 'med', 'high', 'good']
['low', 'low', '3', '2', 'small', 'low', 'unacc']
['low', 'low', '3', '2', 'small', 'med', 'unacc']
['low', 'low', '3', '2', 'small', 'high', 'unacc']
['low', 'low', '3', '2', 'med', 'low', 'unacc']
['low', 'low', '5more', 'more', 'big', 'low', 'unacc']
['low', 'low', '5more', '2', 'small', 'med', 'unacc']
['low', 'low', '2', 'more', 'big', 'low', 'unacc']
['low', 'vhigh', '3', 'more', 'small', 'low', 'unacc']
['low', 'vhigh', '3', 'more', 'small', 'med', 'unacc']
['low', 'med', '4', 'more', 'big', 'low', 'unacc']
['low', 'med', '4', 'more', 'big', 'med', 'good']
```

读取测试数据文件

```
# 读取test.txt
with open('test.txt','r') as f:
    data = f.readlines()
    # 将数据保存在test中
    test = []
    for line in data:
        test.append(line.replace('\n', '').split(','))
# print(test)
```

将指标按等级用不同数字表示

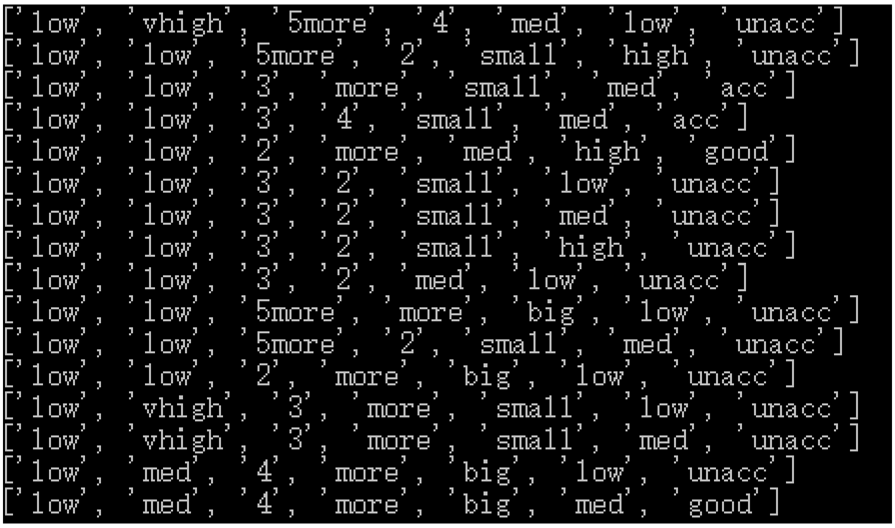
```
# 将指标类别用数字表示
```

```
buying = {'vhigh':0, 'high':1, 'med':2, 'low':3}
maint = {'vhigh':0, 'high':1, 'med':2, 'low':3}
doors = {'2':0, '3':1, '4':2, '5more':3}
persons = {'2':0, '4':1, 'more':2}
lug_boot = {'small':0, 'med':1, 'big':2}
level = {'unacc':0, 'acc':1, 'good':2, 'vgood':3}
i = 0  # 用于计数，判断是哪个指标
for j in range(len(index)):
    for k in range(len(index[0])):
        if i % 7 == 0:
            index[j][k] = buying[index[j][k]]
        elif i % 7 == 1:
            index[j][k] = maint[index[j][k]]
        elif i % 7 == 2:
            index[j][k] = doors[index[j][k]]
        elif i % 7 == 3:
            index[j][k] = persons[index[j][k]]
        elif i % 7 == 4:
            index[j][k] = lug_boot[index[j][k]]
        elif i % 7 == 6:
            index[j][k] = level[index[j][k]]
        i += 1

# 将指标类别用数字表示
i = 0  # 用于计数，判断是哪个指标
for j in range(len(test)):
    for k in range(len(test[0])):
        if i % 6 == 0:
            test[j][k] = buying[test[j][k]]
        elif i % 6 == 1:
            test[j][k] = maint[test[j][k]]
        elif i % 6 == 2:
            test[j][k] = doors[test[j][k]]
        elif i % 6 == 3:
            test[j][k] = persons[test[j][k]]
        elif i % 6 == 4:
            test[j][k] = lug_boot[test[j][k]]
        i += 1
# print(test)

# 转化为array
for i in range(len(test)):
    test[i] = test[i][:5]
test = np.array(test, dtype = float)
# print(test)
```

处理后一部分数据显示如下：



```
['low', 'vhigh', '5more', '4', 'med', 'low', 'unacc']
['low', 'low', '5more', '2', 'small', 'high', 'unacc']
['low', 'low', '3', 'more', 'small', 'med', 'acc']
['low', 'low', '3', '4', 'small', 'med', 'acc']
['low', 'low', '2', 'more', 'med', 'high', 'good']
['low', 'low', '3', '2', 'small', 'low', 'unacc']
['low', 'low', '3', '2', 'small', 'med', 'unacc']
['low', 'low', '3', '2', 'small', 'high', 'unacc']
['low', 'low', '3', '2', 'med', 'low', 'unacc']
['low', 'low', '5more', 'more', 'big', 'low', 'unacc']
['low', 'low', '5more', '2', 'small', 'med', 'unacc']
['low', 'low', '2', 'more', 'big', 'low', 'unacc']
['low', 'vhigh', '3', 'more', 'small', 'low', 'unacc']
['low', 'vhigh', '3', 'more', 'small', 'med', 'unacc']
['low', 'med', '4', 'more', 'big', 'low', 'unacc']
['low', 'med', '4', 'more', 'big', 'med', 'good']
```

将样本按照unacc、acc、good、vgood分类

```
# 将样本按照unacc、acc、good、vgood分类
index_unacc = []
index_acc = []
index_good = []
index_vgood = []
for i in range(len(index)):
    if index[i][6] == 0:
        index_unacc.append(index[i][:5])
    elif index[i][6] == 1:
        index_acc.append(index[i][:5])
    elif index[i][6] == 2:
        index_good.append(index[i][:5])
    elif index[i][6] == 3:
        index_vgood.append(index[i][:5])

# 转化为array
index_unacc = np.array(index_unacc)
index_acc = np.array(index_acc)
index_good = np.array(index_good)
```

```
index_vgood = np.array(index_vgood)
```

计算每个总体的均值

```
# 计算每个总体的均值
mean_unacc = np.mean(index_unacc, axis=0)
mean_acc = np.mean(index_acc, axis=0)
mean_good = np.mean(index_good, axis=0)
mean_vgood = np.mean(index_vgood, axis=0)
mean = np.vstack((mean_unacc,mean_acc,mean_good,mean_vgood))
# print(mean)
```

接下来按照题目要求，需要分成两种不同的情况，**协方差相同**和**协方差不同**的情况：

协方差相同

计算协方差

```
# 计算协方差(协方差相同的情况)
# A_unacc = np.cov(index_unacc.T)
A_unacc = np.dot((index_unacc - mean_unacc).T, index_unacc -
mean_unacc)
A_unacc_inv = np.linalg.inv(A_unacc)
# A_acc = np.cov(index_acc.T)
A_acc = np.dot((index_acc - mean_acc).T, index_acc - mean_acc)
A_acc_inv = np.linalg.inv(A_acc)
# A_good = np.cov(index_good.T)
A_good = np.dot((index_good - mean_good).T, index_good - mean_good)
A_good_inv = np.linalg.inv(A_good)
# A_vgood = np.cov(index_vgood.T)
A_vgood = np.dot((index_vgood - mean_vgood).T, index_vgood -
mean_vgood)
A_vgood_inv = np.linalg.inv(A_vgood)
covariance = 1 / (len(index) - 4) * (A_unacc + A_acc + A_good +
A_vgood)
covariance_inv = np.linalg.inv(covariance)
# print(covariance_inv)
```

使用判别函数进行判断

```
# 判别函数(协方差相同)
def mashi_distance_1(x):
    W = np.zeros((4,4),dtype = float)
    c = 99
    for i in range(4):
        for j in range(4):
            if i != j:
                temp = np.dot(x - (mean[i] + mean[j]) / 2,
covariance_inv)
                W[i][j] = np.dot(temp, (mean[i] - mean[j]).T)
    for i in range(4):
        for j in range(4):
            if W[i][j] < 0:
                valid = 0
                break
            else:
                valid = 1
        if valid:
            c = i
    # print(W)
    return c
```

判断类别之后写入txt文件

```
c = []
for i in range(len(test)):
    c.append(mashi_distance_1(test[i]))
    print('class', i, ':', c[i])

# 将结果写入txt文件
LEVEL = {0:'unacc', 1:'acc', 2:'good', 3:'vgood'}
with open('test.txt','r') as f:
    data = f.readlines()
    result = []
    result_diff = []
    for i in range(len(data)):
        result.append(data[i].replace('\n', '') + ',' + LEVEL[c[i]])
        result_diff.append(data[i].replace('\n', '') + ',' +
LEVEL[c_diff[i]])
    # print(result)
with open('test_result1.txt', 'w') as f:
    for line in result:
        f.write(line + '\n')
```

协方差不同

计算协方差

```
# 计算协方差(协方差不同的情况)
covariance_unacc = 1 / (len(index_unacc) - 1) * A_unacc
covariance_acc = 1 / (len(index_acc) - 1) * A_acc
covariance_good = 1 / (len(index_good) - 1) * A_good
covariance_vgood = 1 / (len(index_vgood) - 1) * A_vgood
covariance_inv_diff = []
covariance_inv_diff.append(np.linalg.inv(covariance_unacc))
covariance_inv_diff.append(np.linalg.inv(covariance_acc))
covariance_inv_diff.append(np.linalg.inv(covariance_good))
covariance_inv_diff.append(np.linalg.inv(covariance_vgood))
print(covariance_inv_diff[1])
```

使用判别函数进行判断

```
# 判别函数(协方差不同)
def mashi_distance_2(x):
    V = np.zeros((4,4),dtype = float)
    c = 99
    for i in range(4):
        for j in range(4):
            if i != j:
                V[i][j] = np.dot(np.dot(x -
mean[i],covariance_inv_diff[i]), (x - mean[i]).T) - np.dot(np.dot(x -
mean[j],covariance_inv_diff[j]), (x - mean[j]).T)
    for i in range(4):
        for j in range(4):
            if v[i][j] < 0:
                valid = 0
                break
            else:
                valid = 1
        if valid:
            c = i
    # print(V)
    return c
```

判断类别之后写入txt文件

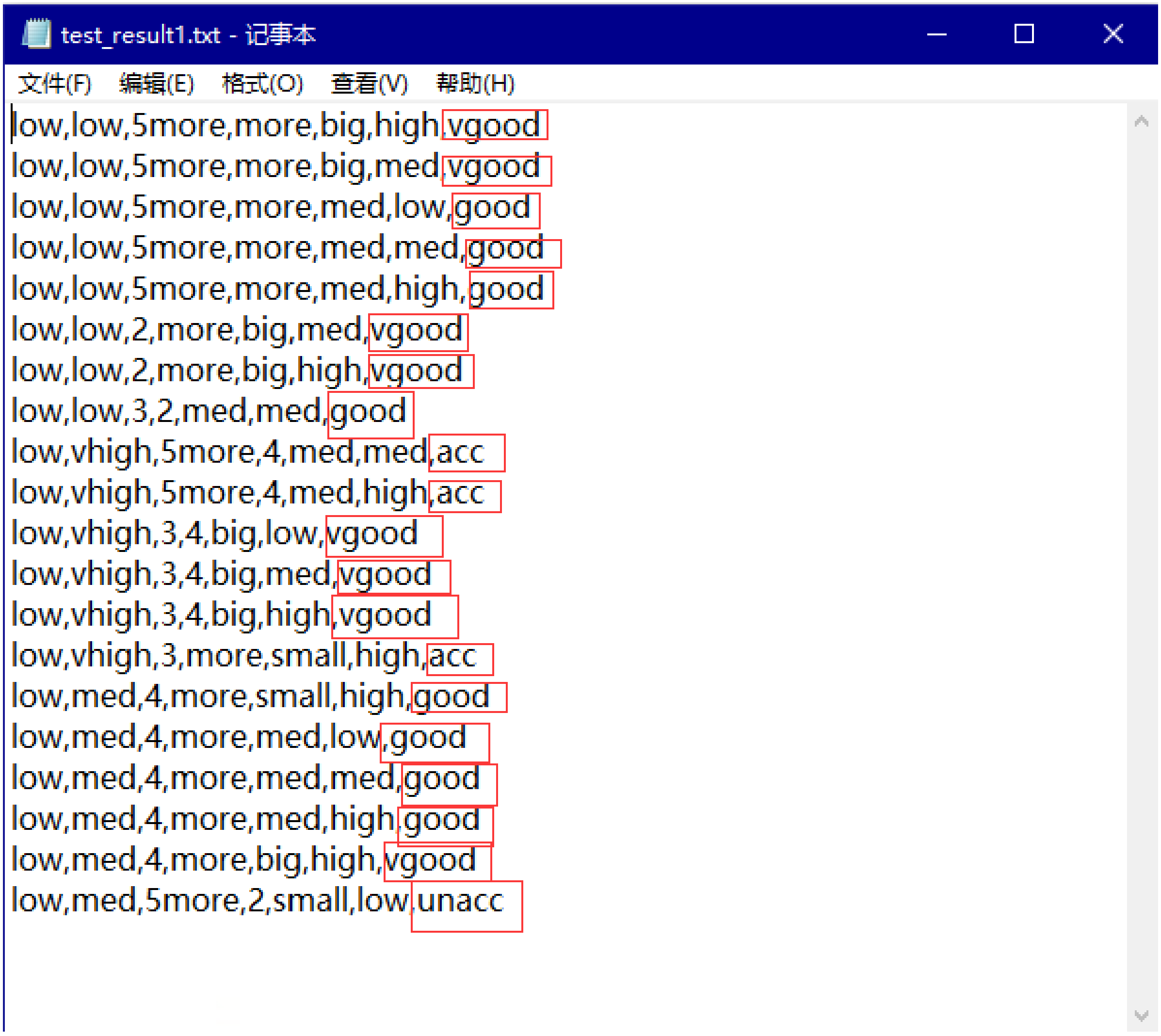
```
c_diff = []
for i in range(len(test)):
    c_diff.append(mashi_distance_2(test[i]))
    print('class', i, ':', c_diff[i])

LEVEL = {0:'unacc', 1:'acc', 2:'good', 3:'vgood'}
with open('test.txt','r') as f:
    data = f.readlines()
    result = []
    result_diff = []
    for i in range(len(data)):
        result.append(data[i].replace('\n', '') + ',' + LEVEL[c[i]])
        result_diff.append(data[i].replace('\n', '') + ',' + LEVEL[c_diff[i]])
    # print(result)
with open('test_result2.txt', 'w') as f:
    for line in result_diff:
        f.write(line + '\n')
```

实验结果

得到的测试集判断结果如下：

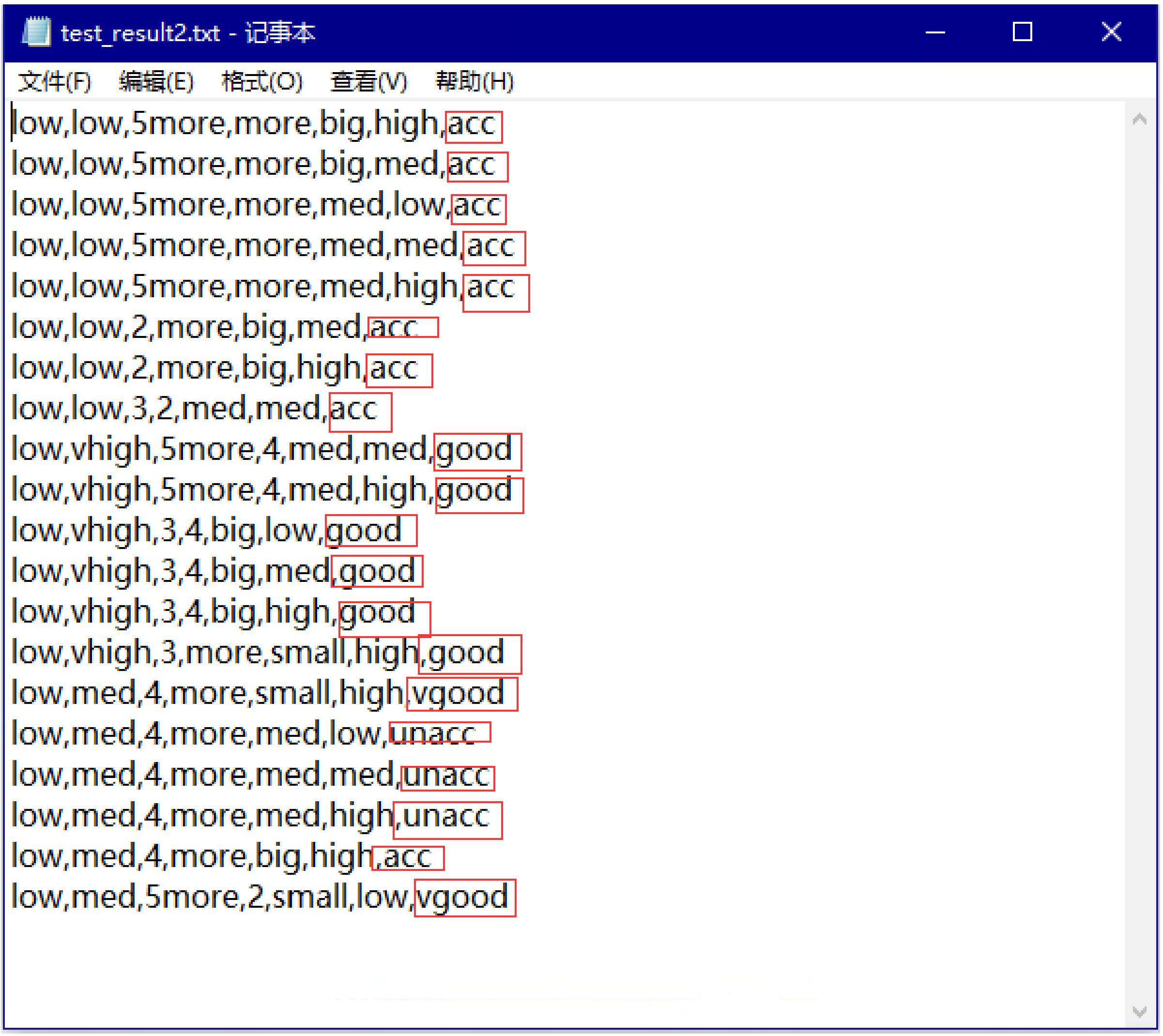
1. 方法1（协方差相同，红框部分为预测结果）



test_result1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
low,low,5more,more,big,high,vgood
low,low,5more,more,big,med,vgood
low,low,5more,more,med,low,good
low,low,5more,more,med,med,good
low,low,5more,more,med,high,good
low,low,2,more,big,med,vgood
low,low,2,more,big,high,vgood
low,low,3,2,med,med,good
low,vhigh,5more,4,med,med,acc
low,vhigh,5more,4,med,high,acc
low,vhigh,3,4,big,low,vgood
low,vhigh,3,4,big,med,vgood
low,vhigh,3,4,big,high,vgood
low,vhigh,3,more,small,high,acc
low,med,4,more,small,high,good
low,med,4,more,med,low,good
low,med,4,more,med,med,good
low,med,4,more,med,high,good
low,med,4,more,big,high,vgood
low,med,5more,2,small,low,unacc
```

2. 方法2（协方差不相同，红框部分为预测结果）


test_result2.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
low,low,5more,more,big,high,acc
low,low,5more,more,big,med,acc
low,low,5more,more,med,low,acc
low,low,5more,more,med,med,acc
low,low,5more,more,med,high,acc
low,low,2,more,big,med,acc
low,low,2,more,big,high,acc
low,low,3,2,med,med,acc
low,vhigh,5more,4,med,med,good
low,vhigh,5more,4,med,high,good
low,vhigh,3,4,big,low,good
low,vhigh,3,4,big,med,good
low,vhigh,3,4,big,high,good
low,vhigh,3,more,small,high,good
low,med,4,more,small,high,vgood
low,med,4,more,med,low,unacc
low,med,4,more,med,med,unacc
low,med,4,more,med,high,unacc
low,med,4,more,big,high,acc
low,med,5more,2,small,low,vgood
```

结果分别存储于文件`test_result1`和`test_result2`中。