

**项目开发报告**

**题目： 基于朴素贝叶斯分类器的语音性别识别**

**课程名称： 机器学习**

**专业班级： CS1703**

**学 号： U201714600**

**姓 名： 李泽贤**

**指导教师： 李玉华**

**报告日期： 2020.6.27**

**计算机科学与技术学院**

# 项目开发报告

## 1.1项目目的

（1）掌握机器学习的一般方法；

（2）理解朴素贝叶斯分类器的构造过程及应用；

（3）加深对python numpy matplotlib等工具的熟练度；

（4）解决实际应用中声音性别识别的问题；

## 1.2问题分析

**具体题目参考**

<https://www.kaggle.com/primaryobjects/voicegender>

**数据集概述**

集合中共有 3168 条数据，男女各 1584 条，每条数据可视

作一个长度为 21 的一维数组。其中前 20 个数值是这条语音的 20 个特征值，这些特征值包括了语音信号的长度、基频、标准差、频带中值点/一分位频率/三分位频率等；最后一个数值是性别标记。元数据集中直接以字符串,即 male 和 female 进行标注。

**问题分析**

使用 7：3 划分数据集。通过朴素贝叶斯方法，可以先对所有特征值做统计，并且通过连续性参数估计（高斯分布）方法得到参数。之后使用预测函数预测测试集。

打开voice.csv文件，20个特征值分类为F1，F2 ,F3 …,F20。

朴素贝叶斯方法，对于某测试数据，Fi’代表其在Fi特征的值

则P(男|F1’×F2’×F3’....×F20’)=P(男)\*P(F1’|男)\*P(F2’|男).....\*P(F20’|男)/

[P(F1)\*P(F2’).....\*P(F20’)]

则可以通过比较P(男)\*P(F1’|男)\*P(F2’|男).....\*P(F20’|男)和P(女)\*P(F1’|女)\*P(F2’|女).....\*P(F20’|女)的大小给出性别。

方法本质上是概率的叠乘，20个特征，多个特征的概率相乘，乘积的结果将会非常小，从而影响实验结果。所以这里统一取对数。

考虑到数据缺失的问题，发现mode 和 dfrange 等列都有为0 的值，说明数据有缺失，若采用高斯分布，则其概率可能接近0，导致总概率非常低，所以对缺失的值，采取该特征平均值代替。结合高斯分布，同时避免了某个概率接近0，则总的概率接近0的情况。

原数据集中直接以字符串male和female进行标注，本次实验用1表示男性、0表示女性。

原数据中男女并不是随机分布，而是前半部分男后半部分女，所以要先打乱或者随机抽取训练数据。

## 1.3设计与分析

设计流程图如图1.1

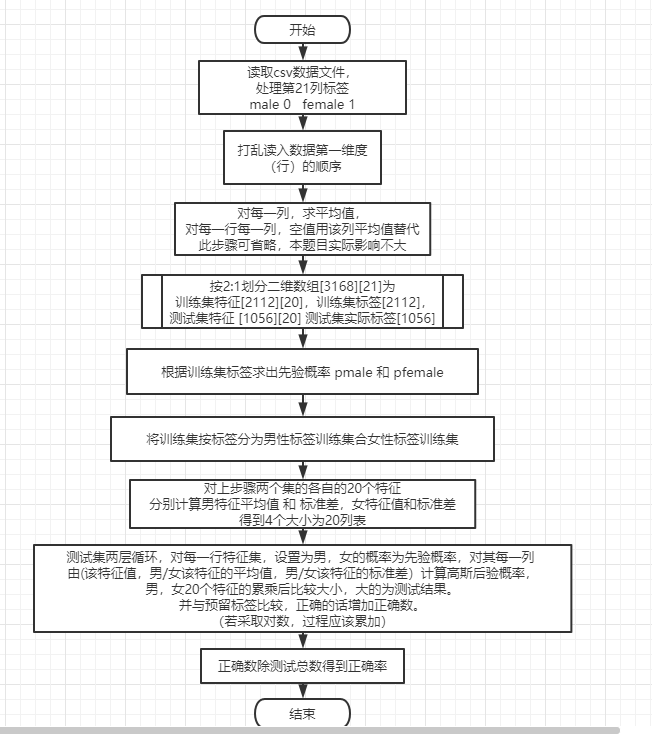


图1.1

## 1.4结果分析（包括实际运行结果截图，性能结果图、表等）

由于每次随机划分测试与训练数据，结果不同，取四次运行结果，如图

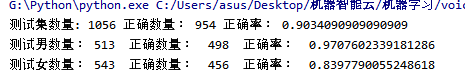


图1.2

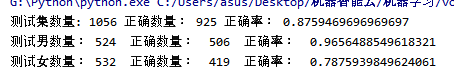


图1.3

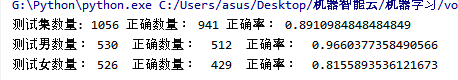


图1.4

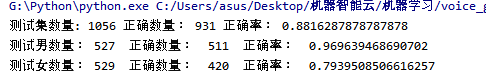


图1.5

综合以上情况得到平均情况如下表1.1

表1.1

|  |  |
| --- | --- |
| 综合正确率 0.89 | 综合错误率 0.11 |
| 男生正确率 0.97 | 男生错误率 0.03 |
| 女生正确率 0.80 | 女生错误率 0.20 |

其它：实验测试过程中发现取对数与否，处理值元数据中值为0的数据与否 对结果影响不大。

## 1.5思考与总结

朴素贝叶斯与那里比较简单，实验难度在于对numpy掌握不熟练。

朴素贝叶斯假设特征独立，但实验实际中有些数据关联比较大，如三分频和中间值。

实验结果表明对男生，女生的测试准确度有很大差别，原因可能是女声的声调（pitch) 变化范围比男性的大，准确说是女性语音的基频更高，而训练数据集不够大而不能很好的覆盖这些变化。

实验给我留下了一个问题，如何将一段人声处理成这20个特征，解决这个问题就能实现一个有实际作用的应用。

## 1.6代码附录

**import** numpy **as** np  
*#csv 文件格式***import** csv  
**import** matplotlib.pyplot **as** plt  
**import** math  
**from** scipy.stats **import** norm  
  
  
*#处理数据得到训练和测试的特征和标签 返回四个列表***def** LoadFileToSet(file\_name):  
 **with** open(file\_name) **as** f:  
 rawCharacteristic = [] *# 特征* trainCharacteristic = []  
 testCharacteristic = []  
 trainLable = []  
 testLable = []  
  
 f\_csv=csv.DictReader(f)  
 lableName =list(f\_csv.fieldnames)  
 *#print(lableName)  
 #* **for** line **in** f\_csv.reader:  
 rawCharacteristic.append(line)  
 *# 转换标签 male 0 female 1* **for** i **in** range(len(rawCharacteristic)):  
 rawCharacteristic[i][20]=0 **if** rawCharacteristic[i][20]==**'male' else** 1  
 *#打乱* np.random.shuffle(rawCharacteristic)  
 *# for line in rawCharacteristic:  
 # print(line)  
 #特征列数 20  
  
 # 求每一个特征的平均值* data\_mat = np.array(rawCharacteristic).astype(float)  
 count\_vector = np.count\_nonzero(data\_mat, axis=0)  
 sum\_vector = np.sum(data\_mat, axis=0)  
 mean\_vector = sum\_vector / count\_vector  
 *## 数据缺失的地方 用 平均值填充 其实去掉对结果没啥影响* **for** row **in** range(len(data\_mat)):  
 **for** col **in** range (0,20):  
 **if** data\_mat[row][col] == 0.0:  
 data\_mat[row][col] = mean\_vector[col]  
 *#划分训练集 测试集 特征和标签* num1 = int(len(data\_mat) / 3 \* 2) *#2112* **for** i **in** range(num1):  
 trainCharacteristic.append(data\_mat[i][:20])  
 trainLable.append(data\_mat[i][20])  
 **for** i **in** range(num1,len(data\_mat)):  
 testCharacteristic.append(data\_mat[i][:20])  
 testLable.append(data\_mat[i][20])  
 *#不能data\_mat[：num][：20] shape 20,21 ???* trainLable=np.array(trainLable).astype(int)  
 testLable=np.array(testLable).astype(int)  
 **return** trainCharacteristic,testCharacteristic,trainLable,testLable  
  
  
*#num1 是2/3数量 2112*trainCharacteristic,testCharacteristic,trainLable,testLable=LoadFileToSet(**"./voice.csv"**)  
*#测试结果标签*TestResultlable=[]  
*# 训练集性别概率 先验概率*pmale = trainLable.sum(axis=0) / len(trainLable)  
pfamle = 1 - pmale  
*# 训练集性别前提下的特征*trainMaleCharacteristic=[]  
trainFemaleCharacteristic=[]  
**for** i **in** range(len(trainLable)):  
 **if** trainLable[i]==0: trainMaleCharacteristic.append(trainCharacteristic[i])  
 **else**: trainFemaleCharacteristic.append(trainCharacteristic[i])  
*#MaleMeanVector=np.sum(trainMaleCharacteristic,axis=0)/len(trainMaleCharacteristic)  
#训练男女各个特征的均值和标准差 var方差 2\*20*MaleMeanVector=np.mean(trainMaleCharacteristic,axis=0)  
MaleDeviationVector=np.sqrt(np.var(trainMaleCharacteristic,axis=0))  
FemaleMeanVector=np.mean(trainFemaleCharacteristic,axis=0)  
FemaleDeviationVector=np.sqrt(np.var(trainFemaleCharacteristic,axis=0))  
  
*#计算概率 过程取对数*BeMaleRrobablityVector=[]  
BeFemaleRrobablityVector=[]  
AccurateResualtNum =0  
MaleAccurateResualtNum =0  
FemaleAccurateResualtNum =0  
**for** i **in** range(len(testLable)):  
 BeMaleRrobablity = math.log2(pmale)  
 BeFemaleRrobablity = math.log2(pfamle)  
 **for** j **in** range(0,20):  
 BeMaleRrobablity+=math.log2(norm.cdf(testCharacteristic[i][j],MaleMeanVector[j],MaleDeviationVector[j]))  
 BeFemaleRrobablity+=math.log2(norm.cdf(testCharacteristic[i][j],FemaleMeanVector[j],FemaleDeviationVector[j]))  
  
 *# #不取对数  
 # BeMaleRrobablity = (pmale)  
 # BeFemaleRrobablity = (pfamle)  
 # for j in range(0,20):  
 # BeMaleRrobablity\*=norm.cdf(testCharacteristic[i][j],MaleMeanVector[j],MaleDeviationVector[j])  
 # BeFemaleRrobablity\*=(norm.cdf(testCharacteristic[i][j],FemaleMeanVector[j],FemaleDeviationVector[j]))  
  
 #print(BeMaleRrobablity,BeFemaleRrobablity)  
 #BeMaleRrobablityVector.append(BeMaleRrobablity)  
 #BeFemaleRrobablityVector.append(BeFemaleRrobablity)* lable1=0 **if** BeMaleRrobablity>BeFemaleRrobablity **else** 1  
 TestResultlable.append(lable1)  
  
 **if** lable1==testLable[i]:  
 AccurateResualtNum +=1  
 **if** lable1==0: MaleAccurateResualtNum+=1  
 **else**: FemaleAccurateResualtNum+=1  
  
  
  
print(**"测试集数量: "**+ str(len(testLable))+**" 正确数量： "**+ str(AccurateResualtNum)+ **" 正确率： "**+str(AccurateResualtNum/len(testLable)))  
print(**"测试男数量："**,len(testLable)-np.count\_nonzero(testLable),**" 正确数量： "**,MaleAccurateResualtNum,**" 正确率： "**,MaleAccurateResualtNum/(len(testLable)-np.count\_nonzero(testLable)))  
print(**"测试女数量："**,np.count\_nonzero(testLable),**" 正确数量： "**,FemaleAccurateResualtNum,**" 正确率： "**,FemaleAccurateResualtNum/np.count\_nonzero(testLable))