華中科技大學

课程报告

题目：基于朴素贝叶斯分类器的语音性别识别

专　　业：　计算机科学与技术

课　　程：　机器学习

班　　级：　CS1706 班

学　　号：　U201714762

姓　　名：　梁一飞

指导教师：　李玉华

2020 年 7 月 25 日

摘 要

语音性别识别是机器学习在生活中的一个新应用，语音辨别通过机器学习可以达到判断语音的机器判断，性别识别这是对于声音处理的基础应用。朴素贝叶斯分类器是机器学习基于贝叶斯公式的重要理论。本实验基于kaggle上的一个数据集，采用朴素贝叶斯分类器，实现了通过语音识别说话人性别的功能。本次以python为实验平台编写代码，研究朴素贝叶斯在语音性别识别的应用，最终基本完成了声音的性别识别并进行粗略的研究，主要内容包括：

朴素贝叶斯分类器的实现，根据贝叶斯公式和朴素贝叶斯的原理设计朴素贝叶斯分类器，最终执行程序。

将分类器应用与性别声音识别，采用了kaggle数据集，最终然后对数据结果进行分析，对朴素贝叶斯有了更深的理解和研究。

目录

1 绪论

1.1 研究目的及意义

语音性别识别是机器学习在生活中的一个新应用，语音辨别通过机器学习可以达到判断语音的机器判断，性别识别这是对于声音处理的基础应用。由此可以引申出更广泛的领域，用于犯罪侦查，身份识别，情感辨析等等。研究语音性别识别是最基础的部分，做好语音性别识别可以往上继续升华，实现更多对于声音信息的利用。

1.2 国内外研究现状

国内外基于朴素贝叶斯分类器已经有了很多研究，其中包括基于朴素贝叶斯分类器算法的康复训练行为识别，基于朴素贝叶斯分类器算法的爬虫研究，基于朴素贝叶斯分类器的棉花受虫危害等级分析等等，研究面很广，而且研究的比较深入，基于朴素贝叶斯分类器的语音识别也有很多人进行了尝试而且也有不错的效果。

1.3 实验环境与平台

1）实验环境条件：Intel Core i5-8750H CPU @2.20GHz，16GB

2）pycharm 平台编译运行

3）调用 numpy，math，csv 库

2 实验设计

2.1 模型与方法

2.1.1 模型选用

本次实验选用的是朴素贝叶斯分类器模型

朴素贝叶斯最核心的部分是贝叶斯定理，贝叶斯定理如下:

$$p(c_i \mid x,y)=\frac{p(x,y \mid c_i)p(c_i)}{p(x,y)}$$

图 2-1 贝叶斯公式

贝叶斯公式由三个部分组成，先验概率，后验概率，似然估计。朴素贝叶斯的基本思想是：对于给出的待分类项，求解出现的条件下各个类别出现的概率，哪个概率最大，那就判定他数与哪一类。

具体流程图如下图所示:

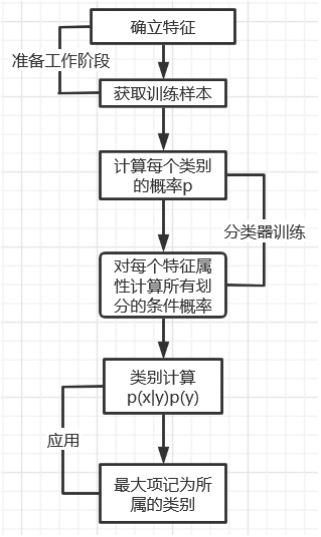

图 2-2 朴素贝叶斯流程图

通过朴素贝叶斯来进行语音性别识别实现的基础模型。

2.1.2 数据选用和处理

我使用的是Kaggle上下载的数据集，Kaggle是一个数据建模和数据分析竞赛平台。这个数据集是基于对男女语音段进行合理的声音预处理而得到的语音特征(并不包含原始语音段)。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，我则用 1 表示男性、0 表示女性以方便后续处理，这当然并无大碍。

3.2 算法设计与分析

算法流程图如下图所示:

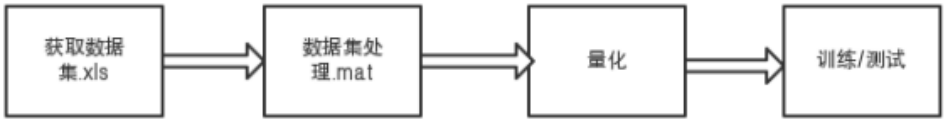

图 3-1 流程图

（1）数据预处理

下载的文件是 csv 文件，读入的都是 3168*21 的矩阵，首先要进行的是数据的量化，贝叶斯分类器是基于条件概率而进行分类的，数据处理不能改变分布律，因此一定要进行线性量化。这里使用 numpy 库，进行数据矩阵处理。

源代码：

```
def loaddata(file_name)
   datamat = []
   with open(file_name) as file_obj:#open the file
          listclass = []
          voice_reader = csv.DictReader(file_obj)
     #读取 csv
      num = len(label_name) - 1
      for l in voice_reader:
         if l[-1] == 'male' ： gender=1
             else： gender=0
       list_class.append(gender)
      # 将连续性数据离散化
      diff_vector = (data_mat.min(axis=0)- data_mat.max(axis=0))/9
      for i in range(len(data_mat)):
```

```
                new_data_set.append(line)
            line = np.array((data_mat[i] - min_v) / diff_v).astype(int)



        # 按照 7：3 划分训练集和测试集
        test_set = list(range(len(new_data_set)))
        train_set = []
        _train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3)
        # 训练数据集


        for i in train_set:
            train_mat.append(new_data_set[i])
            train_c.append(list_class[i])
        # 测试数据集
        for i in test_set:
            test_mat.append(new_data_set[i])
            testc.append(list_class[i])
    return trainmat, train_c, tesmat, tesclass, label
# trainmat：离散化训练数据集 trainclasses： 练数据的分类 label：特征的名称
tesmat：离散化的测试数据集 tesclass：测试数据集的分类#
(2)分类和训练
模型采用的是朴素贝叶斯模型。模型可见实验二实验报告
#test
file_name = 'voice.csv'
    train_mat,train_classes,test_mat,test_classes, label_name = load_data_set(file_name)


    count = 0.0#总计数，男人计数，女人计数
```

```
count1= 0.0
count0= 0.0
correct_count = 0.0#总正确，男人正确，女人正确
correct_count0 = 0.0
correct_count1 = 0.0

for I in list(range(len(test_mat))):
    test_v = test_mat[I]
    result = classify(test_vector, p_feature, p_1_feature, p_1_class)
    if result == test_classes[I]:
        correct_count += 1
        print(" the num is {}: reality:{} theory:{}".format(count,result,test_classes[i]))
        if result==1 :
            correct_count1+=1
        else :correct_count0+=1
    count += 1
    if test_classes[i]==1:
        count1 +=1
    else :count0+=1
print(" the total is {}: malerate:{} female:{}".format(correct_count / count,correct_count1 / count1,correct_count0 / count0))


test_bayes()#classify 分类：
for I in list(range(len(test_vector))):
        sum += feature1[i][test_v [I]]
        sum -= feature0[i][test_v [I]]
    p1 = sum + np.log(pclass)
      if p1 > 0.5:
```

```
            return 1
        else:
            return 0
    # 所有分类下的各特征的概率

    train_matrix = np.matrix(train_matrix)
    for L in list(range(num_feature)):
        feature_values = np.array(train_matrix[L])
        feature_values = feature_values.tolist() + list(range(n)
        count = len(feature_values)
        for value in set(feature_values):
            p[value] = np.log(feature_values.count(value) / float(count))
        p_feature[L] = p

    return feature0, feature1, pclass
```

#pclass 任一样本分类为 1 的概率 feature0, feature1 分别为给定类别的情况下所以特征所有取值的概率

3 实验结果与结论

3.1 实验结果

数据集按照 7：3 划分训练集和测试集大概训练集为 2000 个，总共数据集为 3168 个。

测试结果如下

（1）部分测试集运行结果如图所示：

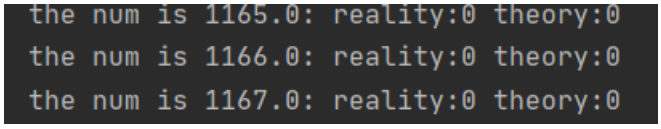

图 3-1 部分测试集运行结果

（2）总结果如图所示：

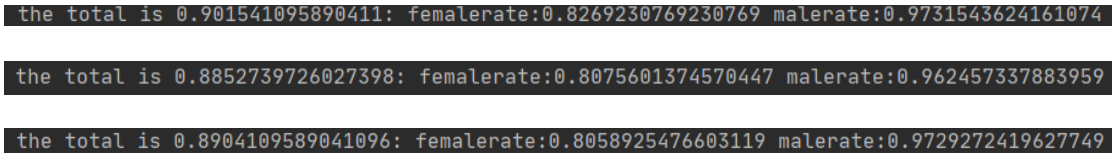

图 3-2 三次实验结果图

实验次数	female	male	total
1	82.69	97.31	90.15
2	80.75	96.24	88.52
3	80.58	97.29	89.04
Average	81.34	96.94	89.23

表 3-1 实验结果汇总

3.2 实验结论

通过本次实验，可以得知朴素贝叶斯能够较高正确率的辨别男女的声音，实现男女声音的辨别，且在其中，男声比女声更容易辨别。

3.3 实验改进

还可以使用 kmeans+Knn 实现提高辨识的准确度：

KNN 的流程如下：

1-准备已标记好类别的合适数据，构成特征空间；

2-选定 K 值；

3-当出现一个新数据时，计算该新数据与当前特征空间中所有已知类别的数据的距离，并按照从小到大的顺序排列；

3-取排列好的数据列表的前 K 条，找出这 K 条数据中条数最多的类别；

4-将该新数据归入此类别

直接调用库进行实验

```
for i=1:test_num
    temp = repmat(Test_m(i,:),2*k,1);
    dists(:,1) = sum((temp-C(:,1:20)).^2,2);
    dists(:,2) = [zeros(k,1);ones(k,1)];
    [B,ind] = sort(dists(:,1));
    ind = ind(1:K,1);
    for j=1:K
        if ind(j,1)<=k
            P_M = P_M+1;
        end
    end
    if P_M>=(K+1)/2          % K需要是奇数
        N_M2M = N_M2M+1;
    else
        N_M2F = N_M2F+1;
    end
    P_M = 0;
```

图 3-3Knn 部分代码

实验结果发现结果受 K 影响已经数据是否准确量化

```
K=10,k=5the total is 0.8998287671232876: femalerate:0.8212389380530973 malerate:0.9734660033167496
```

图 3-4knn 运行结果

K	M/f/t	k=1			k=3			k=5			k=7		
10		96.9	92.7	94.8	96.6	82.6	89.6	97.9	77.2	87.63	96.7	69.2	82.9
20		97.2	93.1	95.1	97.9	86.5	92.2	97.6	82.03	89.81	97.9	77.7	87.8
30		97.8	95.5	96.7	98.2	89.9	94.04	98	84.4	91.25	98.5	79.1	88.8

表 3-2kmeans+knn 结果汇总

由表可知：当 k=1 时效果较好

4 心得体会和总结

4.1 遇到的问题

（1）一开始的时候数据集不是很会处理。

（2）对于 bayes 算法不是很熟练，编写时有很多问题。

（3）数据读入处理不了解，空白数据处理不明白。

4.2 心得体会

这是机器学习的结课项目，我也是第一次做这种和实际生活相关的数据分析，感觉很有趣味，机器学习的目的最终都是数据分析并最后将数据应用于生活，最后也算是磕磕绊绊的把实验做完了，还有很多的不足，希望以后还能继续学习知识，我还想把机器学习继续深入学习，研究声音的别的应用，不只是辨别性别这一点。

参考文献

[1]https://www.kaggle.com/primaryobjects/voicegender
[2]Peter Harrington 机器学习实战,1998,22(4):220-232