浙江大学软件学院-真伪语音鉴别实验报告

南京林业大学-张刘杰

1. 实验环境
2. 操作系统：macOS
3. 编程工具：jupyter notebook
4. 总体思路

参照github项目，该项目内容为对男女两性声音进行鉴别，与真伪语音鉴别在本质上都为二分类鉴别项目。仿照该项目，分离出训练集中的真伪语音，分别训练出两个对应的模型。在判别过程中，分别用两个模型进行判别，比较两者的score函数结果，取score和较大的一方作为最终判别结果。

1. 实验具体内容
2. 读取音频文件并提取特征值

1)未使用spafe库，而是使用librosa读取音频文件

2)利用pdf文档中给取的参数进行mfcc特征提取，使用python-speech-featrue库

3)对提取的特征进行处理修正

相关核心代码：

#读取音频 提取MFCC特征并进行处理 函数

def get\_Mfcc(path):

# 读取音频

sig, fs = librosa.load(path, sr=None)

mfcc\_feature = mfcc(sig=sig,

fs=fs,

num\_ceps= 13,

nfilts=24,

nfft=512,

low\_freq=0,

high\_freq=2000,

dct\_type=2,

use\_energy=False,

lifter=5,

normalize=False)

mfcc\_feature = preprocessing.scale(mfcc\_feature) #对特征进行预处理 标准化

deltas = delta(mfcc\_feature, 2)

double\_deltas = delta(deltas, 2)

combined = np.hstack((mfcc\_feature, deltas, double\_deltas))

return combined

1. 训练模型
2. 分离真伪音频文件
3. 提取音频mfcc特征并整合
4. 训练出两个模型
5. 存储为模型文件待用

相关核心代码：

#读取train.txt 从中分别提取出自然语音文件的文件名 和合成语音的文件名

fpath = 'train.txt'

spoofs = []

bonafides = []

with open(fpath, 'r') as f:

for line in f.readlines():

x = line.strip().split(' ', 1)

if line.strip().endswith('spoof'):

#print(line.strip())

#print(x)

#print(x[0])

spoofs.append(x[0])

else:

bonafides.append(x[0])

#分别对自然语音和合成语音进行特征提取和整合

features\_spoof=np.asarray(())

for f in spoofs:

# compute features

vector = get\_Mfcc(source + f + '.flac')

if features\_spoof.size == 0:

features\_spoof = vector

else:

features\_spoof = np.vstack((features\_spoof, vector))

#自然声

features\_bonafide=np.asarray(())

for f in bonafides:

vector = get\_Mfcc(source + f + '.flac')

if features\_bonafide.size == 0:

features\_bonafide = vector

else:

features\_bonafide = np.vstack((features\_bonafide, vector))

#利用特征训练处自然声音和环境声的模型 并存储为文件

# generate gaussian mixture models

spoof\_gmm = GMM(n\_components = 8, max\_iter = 200, covariance\_type='diag', n\_init = 3)

bonafides\_gmm = GMM(n\_components = 8, max\_iter = 200, covariance\_type='diag', n\_init = 3)

# fit features to models 训练模型

spoof\_gmm.fit(features\_spoof)

bonafides\_gmm.fit(features\_bonafide)

# save models

save\_gmm(spoof\_gmm, "spoof")

save\_gmm(spoof\_gmm, "bonafide")

1. 判别eval中音频
2. 读取音频
3. 提取mfcc特征
4. 代入两个模型中判别
5. 比较两个模型score函数，得出最终判别结果
6. 将结果写入文件中

相关核心代码：

#######################eval1判别阶段############################

#根据模型 对eval 1中音频进行判断并输出结果

eval1\_path = "eval1/flac/"

eval1\_files = [os.path.join(eval1\_path, f) for f in os.listdir(eval1\_path)]

eval1\_out = open("eval1.txt", "w")

for file in eval1\_files:

vector = get\_Mfcc(file)

predict = identify(vector)

eval1\_out.write(file[11:23] + " " + predict + "\n") #输出结果到文件中

#######################eval1判别阶段############################

#根据模型 对eval 2中音频进行判断并输出结果

eval2\_path = "eval2/flac/"

eval2\_files = [os.path.join(eval2\_path, f) for f in os.listdir(eval2\_path)]

eval2\_out = open("eval2.txt", "w")

for file in eval2\_files:

vector = get\_Mfcc(file)

predict = identify(vector)

eval2\_out.write(file[11:23] + " " + predict + "\n")

1. dev验证准确率
2. 对dev文件进行判别
3. 与dev.txt进行比较 得出准确率（82.6%左右）

相关核心代码：

######################dev测试阶段##########################

#利用dev 测试准确率

#利用dict 键值对及对应期望答案 便于后续验证准确率

files={}

for line in open("dev.txt"):

value = line.split()

files[value[0]] = value[1]

dev\_path="dev/flac/"

sum = 0

right = 0

# read the test directory and get the list of test audio files

for file in files:

sum += 1

print("%10s %8s %1s" % ("--> TESTING", ":", dev\_path + file + ".flac"))

vector = get\_Mfcc(dev\_path + file + '.flac')

winner = identify(vector)

expected\_result = files[file]

print("%10s %6s %1s" % ("+ EXPECTATION",":", expected\_result))

print("%10s %3s %1s" % ("+ IDENTIFICATION", ":", winner))

if winner == expected\_result:

right += 1

print("----------------------------------------------------")

print("sum: " + str(sum))

print("right: " + str(right))

print("accuracy: " + str(float(right) / float(sum) \* 100) + "%")