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

一、实验环境

1.操作系统：macOS

2.编程工具：jupyter notebook

二、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。

三、实验具体内容

1.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
```

2.训练模型

1）分离真伪音频文件

2）提取音频 mfcc 特征并整合

3）训练出两个模型

4）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")
```

3.判别 eval 中音频

1）读取音频

2）提取 mfcc 特征

3）代入两个模型中判别

4）比较两个模型 score 函数，得出最终判别结果

5）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")
```

4.dev 验证准确率

1）对 dev 文件进行判别

2）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) +
"%")
```