**智能语音计算实验报告**

大连理工大学 沈佳眉

1. 读取数据，计算特征值，写入.csv文档中保存
2. 利用soundfile第三方库读出flac
3. 利用spafe第三方库中mfcc函数计算每一个音频的特征
4. 读取train.txt文件中的内容，获取该音频的真伪标签值，和上一步中的特征拼合成一条数据
5. 循环1-3步的操作，直到将每一条音频的信息都存成一条数据
6. 将所有数据转换成一个dataframe，写入.csv文件中

（此过程代码参考了老师发布的pdf文档）

关键代码如下：

|  |
| --- |
| for file\_name in file\_names:  # 把文件名和路径拼合，用soundfile库读出flac  path = config.Paths.data\_path\_train + "/" + file\_name  sig, fs = sf.read(path)  # 计算特征  mfccs = mfcc(sig=sig, fs=fs, num\_ceps=config.mfcc\_vars.num\_ceps, nfilts=config.mfcc\_vars.nfilts,  nfft=config.mfcc\_vars.nfft, low\_freq=config.mfcc\_vars.low\_freq,  high\_freq=config.mfcc\_vars.high\_freq, dct\_type=config.mfcc\_vars.dct\_type,  use\_energy=config.mfcc\_vars.use\_energy, lifter=config.mfcc\_vars.lifter,  normalize=config.mfcc\_vars.normalize)  # 从包含文件名和真伪的dataframe txt\_data中查到此音频的真伪标签，保存在label中  fname = file\_name[:-5]  label\_index = txt\_data[(txt\_data.fname == fname)].index.tolist()  label = txt\_data.loc[label\_index[0], "label"]  # print(file\_name,fname,label)  # 把每一个mfccs中的每一个mfcc拆出来，和label并成一条数据  #print(mfccs)  mfccs = np.array(mfccs)  for m in mfccs:  record = dict(mfcc=m,label=label)  data.append(record) |

1. 用train文件夹中的数据训练高斯混合模型，并保存模型
2. 利用sklearn.mixture中GaussianMixture第三方库，新建gmm\_spoof和gmm\_bonafide两个模型
3. 读取步骤一中保存好的.csv文件，根据每一条数据的标签分成“spoof”和“bonafide”两类
4. 调用库中fit函数，分别用两组数据从训练数据中拟合出对应的两个高斯混合模型
5. 利用joblib第三方库保存训练过后的两个模型

（此过程代码参考了老师发布的pdf文档）

关键代码如下：

|  |
| --- |
| features = pd.read\_csv("features\_dataset.csv") data\_spoof = features.loc[features['label'] == "spoof"] data\_bonafide = features.loc[features['label'] == "bonafide"] features\_spoof = np.array(data\_spoof["mfcc"]) features\_bonafide = np.array(data\_bonafide["mfcc"]) fbonafide = [] for f in features\_spoof:  f = f[1:-1]  f = f.split()  f = list(float(fi) for fi in f)  fspoof.append(f) for f in features\_bonafide:  f = f[1:-1]  f = f.split()  f = list(float(fi) for fi in f)  fbonafide.append(f)  gmm\_spoof.fit(fspoof) gmm\_bonafide.fit(fbonafide) joblib.dump(gmm\_spoof, "gmm\_spoof\_model.plk") joblib.dump(gmm\_bonafide, "gmm\_bonafide\_model.plk") |

1. 使用上一步的模型对eval数据进行评估，完成识别
2. 利用joblib第三方库加载步骤二中的两个模型
3. 利用soundfile第三方库读出需要识别的flac文件
4. 利用spafe第三方库中mfcc函数计算每一个音频的特征
5. 分别用两个模型对上一步中提取出的特征进行打分
6. 选取得分高的作为此条音频的识别结果
7. 重复2-5，汇总所有数据
8. 写出数据至.txt文件中，任务完成

（此部分代码参考对象是sklearn官方文档）

关键代码如下：

|  |
| --- |
| # 从文件夹中读出所有.flac文件的名字 file\_names = os.listdir(config.Paths.data\_path\_eval2) gmm\_spoof = joblib.load("gmm\_spoof\_model.plk") gmm\_bonafide = joblib.load("gmm\_bonafide\_model.plk") predict\_result = [] for file\_name in file\_names:  # 把文件名和路径拼合，用soundfile库读出flac  path = config.Paths.data\_path\_eval2 + "/" + file\_name  sig, fs = sf.read(path)  mfccs = mfcc(sig=sig, fs=fs, num\_ceps=config.mfcc\_vars.num\_ceps, nfilts=config.mfcc\_vars.nfilts,  nfft=config.mfcc\_vars.nfft, low\_freq=config.mfcc\_vars.low\_freq,  high\_freq=config.mfcc\_vars.high\_freq, dct\_type=config.mfcc\_vars.dct\_type,  use\_energy=config.mfcc\_vars.use\_energy, lifter=config.mfcc\_vars.lifter,  normalize=config.mfcc\_vars.normalize)  score\_spoof = gmm\_spoof.score(mfccs)  score\_bonafide = gmm\_bonafide.score(mfccs)  # print("score\_spoof", score\_spoof, ";", "score\_bonafide", score\_bonafide)  if score\_spoof > score\_bonafide:  label = "spoof"  else:  label = "bonafide"  record = dict(name=file\_name[:-5], label=label)  predict\_result.append(record)  with open("predict\_eval2\_1.0.txt", "w") as f:  for r in predict\_result:  f.writelines(r["name"] + " " + r["label"] + "\n") |