

特征生成

相关的主要函数写在了featureUtils.py文件中。

特征生成主要使用的是dataGeneration函数。

```
# 参数说明：
#   featureName: 特征的名称(可以自己取)，特征生成完毕后会保存在同一目录下的
#               features/{featureName}下
#   snr: 信噪比
#   featureFun: 特征的提取方法(需要自己写)，下面有详细介绍
#   exParas: 特征提取方法除输入信号之外需要的其他参数
#   nPoints: 生成正负样例各多少组数据
def dataGeneration(featureName, snr, featureFun, exParas, nPoints = 200):
    ...
```

其中，featureFun特征提取方法需要我们自己设计，其函数输入参数定义为输入信号，以及其他参数（使用元组包装起来一并送入，在函数中解包即可），函数输出为从输入信号中提取的特征，下面是一个提取信号均值的特征函数写法，以及其使用方法。

下面是一个使用案例：

```
from featureUtils import *

def getMeanFeature(sig, exParas):
    fs, f0 = exParas #额外参数的解包，与主函数中的(FS, F0)对应，求平均值没有使用到这两个参数，仅作为演示
    return np.mean(sig)

if __name__ == "__main__":
    snr = -55
    featureName = 'Mean'
    nPoint = 50
    dataGeneration(featureName, snr, getMeanFeature, (FS, F0), nPoint)
```

上述代码完成了在-55db时提取正负样例各50组平均值特征数据的工作，其中传入了采样频率FS以及信号频率F0的额外参数，生成的特征会保存在features/Mean文件夹下，如下图所示。同时，如果生成的特征数据文件已经存在，则新生成的特征数据会追加到旧的特征数据中，不会覆盖掉旧数据，因此可多次调用该脚本，增量式地生成数据。



名称	修改日期	类型	大小
-55.pkl	2019/11/28 16:39	PKL 文件	2 KB

对于特征提取方法，featureUtils.py中包含了单频点时域波形的特征提取方法，其需要输入的额外参数有采样频率以及提取单频点频率。

```
def getSiglePointFeature(sig, exParas):
    fs, f0 = exParas
    ...
```

特征测试

特征测试主要使用的是testAcc函数。

```
# 参数说明：
# model:分类器模型，目前仅支持sklearn库中的各种分类器模型
# featureName:特征名称，对应特征生成时自己取得特征名称
# snr:信噪比
# fold: 测试集占总数据的几分之一
def testAcc(model,featureName,snr,fold = 5):
    ...
```

下面是一个使用案例：

```
from featureUtils import *
from sklearn import svm

if __name__ == "__main__":
    snr = -55
    featureName = 'Mean'
    model = svm.SVC(kernel = 'linear')
    testAcc(model,featureName,snr)
```

上述代码的运行结果如下：

```
X.shape:(100, 1)
y.shape:(100,)
Acc:0.6000
```

程序会输出特征数据的数量以及特征的维度，并将model的分类准确度打印出来，由于均值特征区分性信号以及噪声的效果很差，其分类结果也很差。