青蛙叫声聚类分析实验报告

# 一、实验环境

操作系统：Windows10 专业版 64位

处理器：Intel(R) Core(TM) i7-8700 @3.2GHZ

内存：16G

编程语言：Python 3.7

集成环境：Anaconda3

库文件： fastdtw 0.3.2 (pip install fastdtw)

# 二、实验设计

## 1. MFCC与距离度量方式

参考资料：

1. Mel-frequency cepstrum <https://en.wikipedia.org/wiki/Mel-frequency_cepstrum>
2. 音频特征(MFCC)提取 <https://blog.csdn.net/yunnangf/article/details/78965446>
3. 梅尔频率倒谱系数 <https://blog.csdn.net/zouxy09/article/details/9156785>

MFCC是一种音频特征的表现形式。声音的形状可以通过语音短时功率谱的包络中显示出来。MFCC是描述这个包络的一种特征。

提取MFCC特征的步骤为：

1. 对音频信号预加重、分帧和加窗
2. FFT(快速傅里叶变换)得到频谱
3. 频谱通过Mel滤波器组得到Mel频谱
4. 在Mel频谱上面进行倒谱分析得到MFCC特征

基于以上音频特征，在计算音频数据距离度量时分别使用以下两种方法：

## 1.1 余弦相似度

在比较两条MFCC特征数据时使用该方法。将MFCC特征数据理解为多维空间向量，两个向量夹角的余弦值作为衡量两个个体间差异的大小。余弦值越接近1，就表明夹角越接近0度，也就是两个向量越相似。

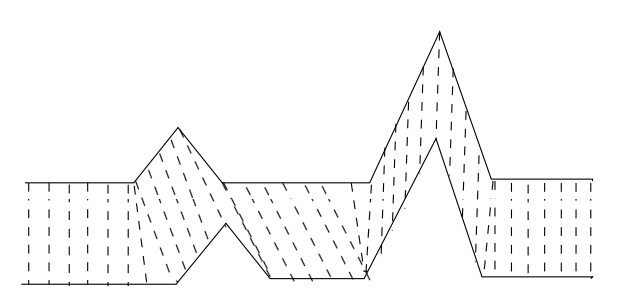
向量x和向量y的夹角的余弦计算公式为：

具体在项目中，按以下步骤计算MFCC特征向量x和y的距离：

1. 求
2. 求x与y范数的乘积
3. 计算余弦值
4. 计算距离

## 1.2 动态时间规整DTW

DTW常被用在语音识别领域，求解两模板匹配时累计距离最小所对应的规整函数，该算法基于动态规划的思想，解决了发音长短不一的模板匹配问题。MFCC特征值是从基于时间序列的音频信号中提取的特征，也是与时间相关的数据。DTW把短的序列线性放大到和长的序列一样的长度再比较，使用动态规划来实现该算法。



算法伪码如下所示：

|  |
| --- |
| DTW-DISTANCE (X, Y):  let D be a new array of size n \* m  for i=1 to n  D[i][0] = INFINITY  for i=1 to m  D[0][i] = INFNITY  D[0][0]=0  for i=1 to n  for j=1 to m  cost=|X[i]-Y[j]|  D[i][j]=cost + min(D[i-1][j], D[i][j-1], D[i-1][j-1])//插入、删除和匹配  return D[n][m] |

但是这样的方法还是有效率问题，fastdtw基于论文的研究成果，将复杂度控制到线性时间和空间复杂度，因此直接使用已有的方法。

论文：

Stan Salvador, and Philip Chan. "FastDTW: Toward accurate dynamic time warping in linear time and space." Intelligent Data Analysis 11.5 (2007): 561-580.

## 2. 降维和特征选择

由于原始数据的正确标注结果为字符串，所以需要将原始标注结果转换为数值型数据，由于本实验只针对科进行分类，因此定义：

FAMILY = ['Bufonidae','Dendrobatidae','Hylidae','Leptodactylidae']

使用pandas导入数据，并将对应的值进行修改：

data\_frame['Family']

= data\_frame['Family'].map(lambdax: FAMILY.index(x))

由于原始数据维度过多，因此考虑使用**PCA**进行降维。PCA通过计算数据矩阵的协方差矩阵，得到协方差矩阵的特征值和特征向量，选取特征值最大的N个特征所对应的特征向量组成的矩阵，可以将数据转到新的空间，实现数据特征降维。本实验实现了PCA算法，并封装为模块进行调用。

本算法的实现流程为：

1. 减去平均值
2. 计算协方差矩阵
3. 计算协方差矩阵的特征值和特征向量
4. 对特征值排序
5. 选取前N个最大的特征向量
6. 转换数据到前N个特征向量构建的空间

## 3. 算法1：k-means + agnes

算法的思路是使用k-means划分多个类，再使用agnes层次聚类算法聚类到FAMILY指定的4类。

k-means是一种划分聚类方法，使用最简单的k-means算法：

1. 随机选择K个随机的点
2. 对与数据集中的每个数据点，按照距离K个中心点的距离，将其与距离最近的中心点关联起来，与同一中心点关联的所有点聚成一类；
3. 计算每一组的均值，将该组所关联的中心点移动到平均值的位置
4. 重复执行（2）-（3）步，直至中心点不再变化

在k-means执行过程中，使用**余弦相似度**来衡量MFCC特征值之间的距离。

agnes是一种层次聚类方法，agnes使用自底向上的策略进行簇之间的聚类。在前一步中，数据集已经按照k-means算法划分为多个簇，使用agnes计算簇之间的距离，由下而上合并距离最小的两个簇，直到最终只剩下4个分类。

agnes的流程为：

1. 将k-means聚类结果转化为初始簇
2. 计算每两个簇之间的距离，找到距离最近的两个簇
3. 合并两个簇
4. 如果簇的数量大于4，重复（2）-（4），否则算法结束

在这一步的聚类过程中，使用**DTW**衡量特征值之间的距离。

## 4. 算法2： dbscan + agnes

算法的思路是使用dbscan划分多个类，再使用agnes层次聚类算法聚类到FAMILY指定的4类。

dbscan是一种基于密度的聚类算法，在具有噪声的空间数据库中发现任意形状的簇，它将簇定义为密度相连的点的最大集合。

算法的流程为：

1. 随机选取一个核心点
2. 对该核心点进行扩充，遍历该核心点邻域内的所有核心点，寻找与这些数据点密度相连的点，直到没有可以扩充的数据为止
3. 将这些点标记为一类
4. 如果仍有核心点，重复以上过程