青蛙叫声聚类分析

武笑石
2016011595
软件 73

摘要

在这次作业中，我利用 scikit-learn 所提供的框架针对青蛙叫声聚类分析的任务进行了实验。实验中，我用到了两种聚类方法、多种特征选择、以及多种模型评价方法。此外，我自己手动实现了逻辑回归模型，并分析了其与 scikit-learn 库中的实现的异同。

1. 实验报告结构

这份实验报告将分为七部分进行介绍。第二部分将交代实验环境，第三部分交代了实验代码的文件结构以及使用方法。第四、五、六部分按照特征选择、模型、评估的结构分别进行介绍。第七部分展示并分析了实验结果。

2. 实验环境说明

操作系统	Windows 10 education
CPU	Intel i5-8250U
python 版本	3.6.10
虚拟环境	conda
关键依赖	numpy=1.18.4 pandas=1.0.3 scikit-learn=0.23.1 matplotlib=3.2.1 seaborn=0.10.1

3. 代码结构说明

3.1 文件结构

task2.py	程序的入口，数据预处理、模型搭建、模型评估逻辑均在此文件 中实现。代码中有完整清晰的注释。
kmeans.py	自己实现的 KMeans 聚类算法。我的实现利用到了 numpy,，但具体的算法逻辑上完全由我手工实现。代码中同样有完整清晰的注释。
data/	训练数据，为确保程序正常运行，请务必将 data 文件夹置于上述两个代码文件的同级目录。

3.2 使用方法说明

本报告中的所有实验共享同一个入口，也就是执行 task2.py 文件。可以通过配置不同参数的方式改变实验条件。以下表格介绍各命令行参数的含义。

此外，可以直接执行:

python task2.py –help

以查看更详细的使用说明。

命令行参数	意义	取值范围
--drop_feature_list	一个列表，该列表中出现的 feature 将不会在训练中被使用。	数据集中出现的类别。
--algorithm	指定使用的聚类算法。自己手工实现的聚类算法同样可以在这里指定。	“kmeans”, “spectral”, “mykmeans”,
--class_number	类数。科：4，属 8，种：10	4，8，10
--visualize	是否做可视化	无参数

--max_iter	solver 最大迭代次数。对 random forest 不适用。	正整数

4. 特征选择

我使用了递归特征消除的方法获得了一组效果较好的特征，效果对比将在第七部分给出。

5. 模型

此次作业中对三种不同的模型进行了测试。分别为：KMeans 算法、spectral clustering 及自己手工实现的 KMeans 算法。

特别需要说明的是，我自己实现的类调用了 numpy 中的函数，但所有迭代逻辑、参数更新逻辑完全由我手工实现，numpy 的函数只是为了简化代码、方便数学计算。

在实验中，我实现的 KMeans 算法稳定性差于算法库中提供的算法。其原因为，算法库中的 KMeans 初始化方法为 KMeans++，可以确保初始化时生成的几个初始质心之间距离比较远，较为稳定、且可较快收敛。我实现的 KMeans 中没有应用特别的初始化方法，因此稳定性劣于算法库，但在某些情况下，算法效果会超过 KMeans++。相关内容将在第七部分实验结果部分进行进一步更详细的说明。

MFCC 算法的输入为一段音频信号，输出为多个离散值特征。操作过程主要为两次不同类型的傅里叶变换，两次变换之间穿插了时间轴和强度轴上的两次非线性变换，其中强度轴上的非线性变换为对数运算，时间轴上的非线性变换将物理刻度转换为了人的感官刻度。

我选用的距离衡量方法为欧氏距离。我参考了论文[1]，该论文验证了 MFCC 特征结合欧氏距离可以很好地完成语音检测任务，因此在此任务中理应也可以获得较好的效果。

6. 评估方法介绍

本次实验中应用到了两种评估方法，一种依赖 ground truth，另一种不依赖 ground truth。

由于数据集本身提供了 ground truth，因此最直接的方法是类似有监督学习，直接比较预测值与真实值的吻合程度。由于聚类问题在对比时有标签与预测值之间的对应问题，我使用了 scikit-learn 中提供的 adjusted_rand_score 方法，该方法会输出最为吻合的一种排列下的比较结果。

不依赖 ground truth 的方法中，我选择了 silhouette_score。根据 scikit-learn 官方文档的介绍，该方法主要衡量的是分类结果对应的类内方差以及与最近邻类之间的类间方差，可以较好地反映聚类的效果。

7. 实验结果

在本次实验报告涉及到的所有实验中，聚类对象都为科，也就是分类的类别数恒为 4。

这部分中，我将在 7.1 节对比我实现的和库里提供的 KMeans 之间的性能差异，并分析差异原因。在对比不同模型时，我使用了全部特征。

此外，我将在 7.2 节对比不同数据输入特征选择下，模型的运行效果。在此部分，我用的是效果最好的 KMeans 方法。

在 7.3 节，我将展示利用 t-SNE 对聚类结果进行可视化的结果。

在 7.4 节我将展示不同超参数下 Spectral Clustering 的表现。

模型	adjusted rand score	silhouette score
KMeans	0.782	0.379
Spectral Clustering（参数 1）	0.756	0.356
Spectral Clustering (参数 2)	0.737	0.383
KMeans 个人实现（最好）	**0.822**	0.325
KMeans 个人实现（最差）	0.761	0.385
KMeans +特征选择	0.802	**0.433**

7.1 模型对比

整体而言，scikit-learn 提供的 KMeans 效果更加稳定，而且整体上效果不错。我实现的 KMeans 方法不够稳定，但是在最好情况下，在于 ground truth 对比时能达到更好的分数。

我分析这里差异的原因是由于我实现的算法并没有应用到特殊的初始化方式，完全随机生成初始质心。而算法库中提供的方法使用了 KMeans++的初始化方法，该方法整体思想是在初始化时尽量拉开质心之间的距离，对于稳定初始化较为有利。此外，scikit-learn 内置的 KMeans 有重复多次取最优的操作，也起到了稳定模型的作用。

值得注意的是，adjusted rand score 与 silhouette score 两种方法之间并不存在严格的正相关关系。例如我实现的 KMeans 在 adjusted rand score 取得最高分数时另一项指标分数较低，而反之亦然。因此单单凭借一个指标来评价聚类算法的优劣是不合理的。

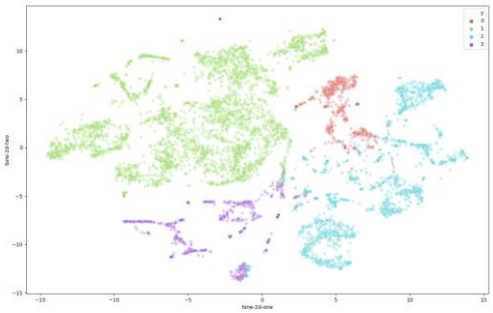
相对来讲，adjusted rand score 更多反映的是结果与 ground truth 的符合程度，因此如果 ground truth 本身在特征空间上就没有很好的聚散结构，那么这一指标就并不适合衡量聚类的结果，甚至这个问题都不适合用聚类的方法来处理，既然有 ground truth，完全可以使用监督学习的思路。

另一方面，silhouette score 则是分类与空间结构之间的关系的一种量化，直接衡量聚类的效果，因此其分数能够更好地反映聚类算法的优劣。

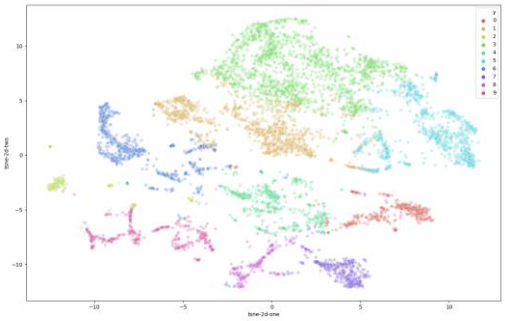
7.2 特征选择

利用递归特征消除方法，我找出了较优的一组特征选择： ['MFCCs_ 3', 'MFCCs_ 7', 'MFCCs_ 8', 'MFCCs_10', 'MFCCs_11', 'MFCCs_12', 'MFCCs_13', 'MFCCs_14', 'MFCCs_17', 'MFCCs_19', 'MFCCs_22']。如上表所示，在这组特征选择下，两种评价方式的结果都取得了较大的提升。

7.3 t-SNE 可视化结果



图上的四种不同颜色分别对应了聚类算法计算输出的四种不同类别，可以看到不同类别的样本在数据降维后相互区分开来，取得了较好的聚类效果。下图是对不同青蛙物种进行聚类的结果（分为十类），可以看到不同类别之间的划分也相当清晰。



7.4 不同超参数的影响

在 Spectral Clustering 的实验中，我尝试调整了算法中关于距离度量的核参数 gamma。在表中的参数 1 设定下，gamma 被设定为了 1.0，参数 2 设定下 gamma 为 1.5。

从 silhouette score 分数可以看出，提高 gamma 后聚类效果变好，但与 ground truth 吻合度下降。

参考文献

[1] Hardware Implementation of Speech Recognition Using MFCC and Euclidean Distance