Project 1：语音端点检测

519030910402 包清泉

1. 基于统计模型分类器和语音频域特征的语音端点检测算法

图1展示了以 GMM 为例的工作流，其中分别为预处理与特征提取、分类器以及后处理三个部分。

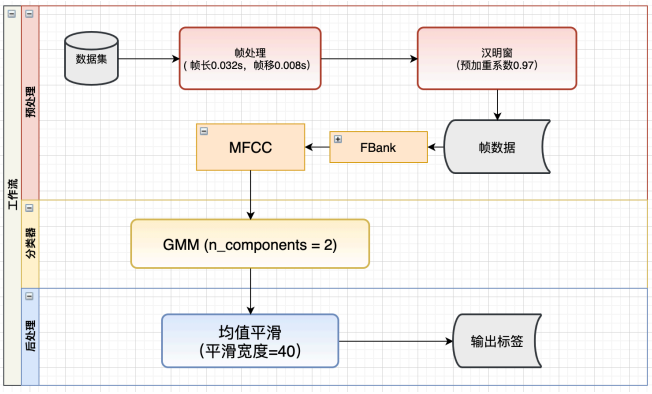

图 1: 工作流

1.1. 数据预处理及特征提取

预处理与特征提取分为 3 步骤，分帧、加窗和特征提取。

1.1.1. 分帧

我们假设语音信号具有短时平稳的特性，故而可以通过分帧的方式对于每一帧的信号进行特征提取和分析。在本文中，我们采用帧长 $0.032s$ 和帧移 $0.008s$ 的参数，使得每一帧之间有一定的重叠，更好表现语音的连续变化。

1.1.2. 加窗

对于每帧的数据，我们使用 Hamming 窗进行处理，从而减少窗边界的影响 [1]。Hamming 窗的数学表示如等式1，

$$w(n) = 0.54 - 0.46\cos(2\pi n M - 1) \quad 0 \le n \le M - 1 \tag{1}$$

另外，我们还对每一帧的数据进行了预加重处理，见公式2，从而提升信号的信噪比。

$$x_n = x_n - \alpha x_{n-1} \tag{2}$$

其中我们设置预加重系数 $\alpha = 0.97$。

1.1.3. Fbank 特征提取

Filter Bank 使用一系列的带通滤波器对原数据的频谱进行特征提取，一般而言，我们使用三角滤波器，于是它代表了每一段频域上语音的信号特征的强弱。形式上，对于一个大小为 M 的 Filter Bank，语音数据为 $\mathbf{x} \in \mathcal{R}^N$，FBank 可以表示为公式3，

$$Fbank[i] = \sum_{j=1}^{N} \mathcal{F}(\mathbf{x})[j] \cdot \mathbf{bank_i}[j] \tag{3}$$

其中 $\mathcal{F}(\cdot)$ 是傅立叶变换，$\mathbf{bank_i}$ 表示第 i 个滤波器的频域值。

具体而言，我们在 Mel 域上进行 FBank 的操作。Mel 域是一个经验性的人耳基音感知域，也即它是通过对人耳可以区分的两种纯音频率之差作为标度，因而富有人类先验知识。Mel 域和线性频域可以通过公式4进行转换，

$$Mel(f) = 2595\log_{10}(1 + \frac{f}{700}) \tag{4}$$

1.1.4. MFCC 提取

Mel Frequency Cepstral Coefficient (MFCC) 是从 Mel 域的 FBank 系数衍生而来。由于 FBank 中各个维度的值线性相关度较高，且方差较大，故而我们使用 $\log(\cdot)$ 和离散余弦变换 (DCT) 获得

MFCC，数学上如公式5，

$$c_n = \sqrt{\frac{2}{N_{fb}}} \sum_{j=1} N_{fb} \log(mel_j) \cos\left(\frac{\pi n}{N_{fb}}(j-0.5)\right) \tag{5}$$

其中 c_n 是第 n 个 MFCC 系数，$n = 1, 2, ..., N_{mfcc}$，mel_j 是第 j 个 Mel 域的 FBank 系数，N_{fb} 和 N_{mfcc} 分别是 Mel 域的 FBank 系数和 MFCC 系数的数量。

1.2. 算法描述

1.2.1. GMM

我们使用了 Gaussian Mixture Model (GMM) 对特征进行建模。一般地，一个 GMM 的概率分布函数可以表示为

$$f(\mathbf{x}) = \sum_{i=1}^{K} c_i \mathcal{N}(\mathbf{x}; \theta) \tag{6}$$

其中，K 表示 GMM 的高斯分量的个数，$\mathcal{N}(\mathbf{x}; \theta)$ 是参数为 θ 的高斯分布。

对于语音和非语音的每一帧特征 x_t，我们假设其分别遵循两个 GMM 分布 μ_{vocal} 和 $\mu_{non-vocal}$。我们使用贝叶斯推断的方式对一个未知语音特征进行分类。具体而言，首先计算该特征在每个类的似然度，进行比较，从而判断类别，数学描述见公式7。

$$label(\mathbf{x}) = \begin{cases} \text{Vocal} & \text{if} \quad \mu_{vocal}(\mathbf{x}) > \mu_{non-vocal}(\mathbf{x}) \\ \text{Non-vocal} & \text{otherwise} \end{cases} \tag{7}$$

1.2.2. LSTM

Long Short Term Memory (LSTM) 是一种特殊的 Recurrent Neural Network (RNN)，是一类专门用于处理序列到序列的神经网络。一个典型的 LSTM 单元如图2所示，由遗忘门、输入门和输出门三部分组成，它相比于 Vinilla RNN 拥有更好的收敛性，因而广泛使用。

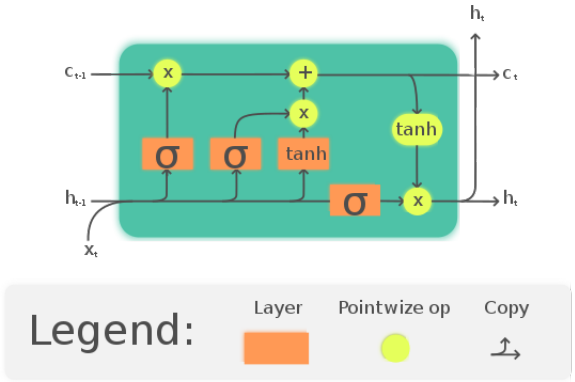

图 2: *A typical LSTM cell[2]*

1.3. 后处理

我们使用了均值平滑作为预测结果的后处理。它的数学形式可以用卷积来表达，如公式8。

$$\mathbf{x} = \mathbf{x} * \mathbf{w} \tag{8}$$

其中 $\mathbf{w}$ 是一个长度为 $winlen$，大小为 $\frac{1}{winlen}$ 的序列。

1.4. 实验结果

1.4.1. 参数和模型设置

我们分别设置 Mel 域的 FBank 的数量和 MFCC 的数量 $N_{fb} = 40, Nmfcc = 20$ 进行相应特征提取。对于 GMM 算法，每种标签的 GMM，我们均设置其高斯分量的个数 $K = 2$，后处理部分采取窗长 $winlen = 40$，最后预测时的阈值为 0.45。

对于 LSTM 算法，结构上，我们使用两个 LSTM CELL 进行叠加，输出与特征向量相同维度的向量，再经过两层线性层（本实验中，是分别尺寸为 40*20 和 20*1 的两层线性层，中间夹以 ReLU 激活层）输出 0-1 值，后处理部分采取窗长 $winlen = 15$，最后预测时的阈值为 0.5。训练过程中，我们将数据集进行拼接，并每隔 1024 帧视作一段序列，以 $BatchSize = 128$，优化器为 Adam，损失函数为最简单的 2 范数，学习率 $lr = 0.009$，训练次数 $epoch = 128$ 进行训练。

表 1: 实验结果

Model	AUC	EER	ACC(dev)
all 1	0.5	0.9999	0.815
MFCC20 + GMM	0.9185	0.0970	0.9300
MFCC20 + GMM + Smooth	0.9788	0.0700	0.9597
Mel40 + LSTM + Smooth	**0.9902**	**0.0389**	**0.9726**

1.4.2. 标准介绍与实验分析

我们选用了三个指标来评价我们的模型在开发集上表现的优劣，分别是 Area Under Curve (AUC), Equal Error Rate (EER), 和准确率 (Accuracy)。前二者的指标均是基于 Receiver Operating Characteristic 曲线形成的指标：AUC 越大，其反映了模型的预测值对于各阈值的泛化性能越佳；而 EER 越小，其反映了当模型的误识率和误拒率相等时模型预测出错越少。

我们首先在训练集上，对各个模型的参数进行估计，随后使用该参数对开发集进行测试，结果如表1所示。可以发现，LSTM 的拟合效果最佳，且进一步实验发现，epoch 越大，其在开发集的 ACC 效果越好；另外 Smooth 操作也能较大提升模型的准确率，无论是 GMM 或是 LSTM，原因应在于 Smooth 操作让输出标签更加平滑，更符合最小标度为 10 毫秒的真值标签。

最后我们提交基于 LSTM 模型的预测标签，所有代码可在以下查看https://github.com/QingquanBao/VAD#spectral-feature--gmm

2. References

[1] Fredric J. Harris. On the use of windows for harmonic analysis with the discrete fourier transform. In *Proc. IEEE*, pages 51–83, 1978.

[2] Wikipeida. Long short-term memory.