Project 1：基于线性分类器和语音短时能量的简单语音端点检测算法

519030910352 郭奕玮

摘要：本次 Project 通过提取、归一化语音帧级别的短时能量和短时过零率特征，基于决策树思想，在开发集数据上进行了简单的 VAD 模型训练，并基于 AUC 和 EER 指标，对预测结果进行了平滑操作以求得最优评估结果。通过选择合适的决策树结构以及最优的平滑参数，本次实验得到了可解释性强且简洁、易于移植的语音活性检测算法。

1. 数据预处理及特征提取

1.1. 数据均值标准化

由于本次项目将要使用语音短时过零率特征，如果语音数据含有直流分量，将在很大程度上影响指标计算准确性。从图1中可以看出绝大部分音频不存在显著直流分量，但是为排除特殊值影响，仍然对每条音频做了减去均值的处理。

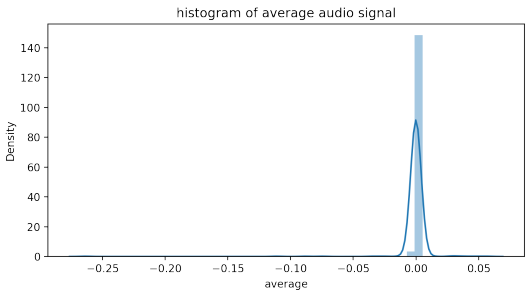

图 1: 开发集语音信号平均值分布

1.2. 短时能量提取和归一化

语音信号的短时能量 E_s 通过式(1)给出，其中 x_i, N 分别表示一帧中的采样值和帧长。

$$E_s = \sum_{i=0}^{N-1} x_i^2 \tag{1}$$

然而直接以能量值本身作为特征存在尺度上问题，因为每条音频（或者每个说话人）讲话的背景噪声和音量存在差异，将会导致**绝对的能量值并不具有泛性的信息**。图2展示了开发集上语音帧短时能量分布情况。显见这些语音帧能量最小值几乎为 0，而平均值和最大值有比较大的方差。这揭示了对短时能量做 **Min-Max 归一化**很有必要，即对每一条语音，分帧并计算每一帧的能量之后，将这些帧的能量线性映射到 $[0, 1]$ 内。下文所称能量，在不特殊说明情况下，均指归一化能量。

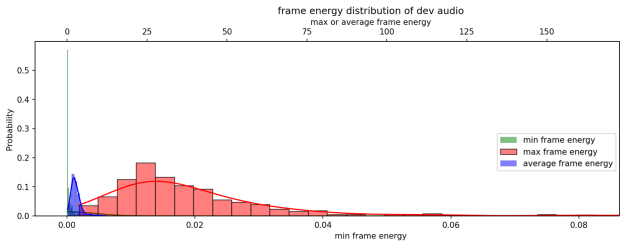

图 2: 开发集音频短时能量平均值、最大值、最小值分布。其中因为数据尺度的原因，上坐标轴表示平均值和最大值，下坐标轴表示最小值。

进行归一化之后，我们按照开发集的标注，对语音和非语音帧的能量分别分析，得到图3。从图中我们可以看出语音和静音帧短时能量有显著特点：静音帧能量集中接近 0，而虽然有许多语音帧的能量离 0 较远，但仍然有一大部分在 0 附近。这个特点和语音本身的特性非常相符，因为语音中的浊音具有显著的周期性，能量也较高；而清音段能量低，与噪声难以区分。这启示我们依靠能量可以有效分出浊音，而静音和噪音需要另外的特

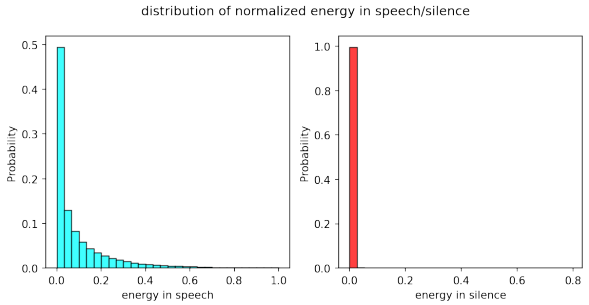

图 3: 开发集语音和静音帧上的（归一化）能量分布

征加以区分。这是本文使用**决策树模型**的出发点。

1.3. 短时过零率提取和分析

根据清音和静音过零率存在差异的先验知识，我们提取每帧的信号短时过零率 Z_s 如式(2)所示。

$$Z_s = \frac{1}{2N}\sum_{i=1}^{N-1}|\,\mathrm{sign}(x_i) - \mathrm{sign}(x_{i-1})|, \qquad (2)$$

$$\text{其中}\ \mathrm{sign}(x_i) = \begin{cases} 1 & \text{if } x_i > 0 \\ 0 & \text{if } x_i = 0 \\ -1 & \text{if } x_i < 0 \end{cases}. \qquad (3)$$

从而我们可以分析语音/静音帧中的过零率情况。总体的分布情况见图4。可以发现语音（包括清音和静音）中的过零率主要集中在较低部分，而静音过零率有比较大的变化范围；我们认为这是因为从声学角度，静音时长远少于浊音，因此在总体分布上过零率差异不显著。基于图3的启示，我们进一步分析低能量帧中语音和静音的过零率分布，以及语音中的过零率分布，分别如图5、图6所示。

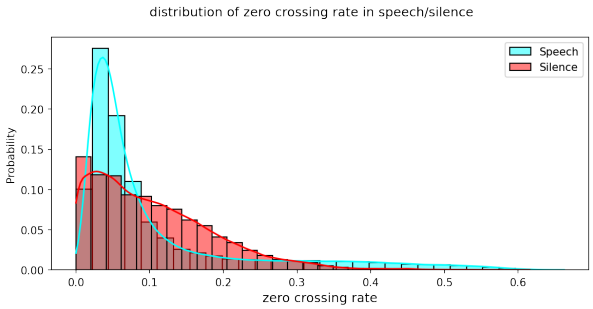

图 4: 开发集语音和静音帧的短时过零率分布

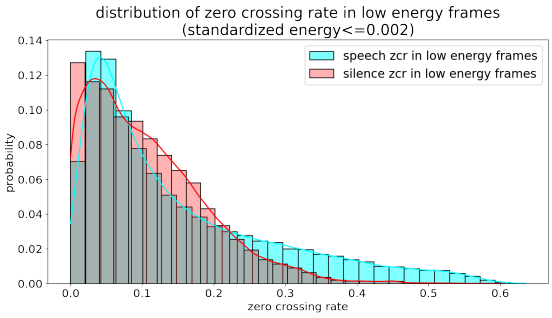

图 5: 开发集中，低能量（不高于 *0.002*）帧的过零率分布

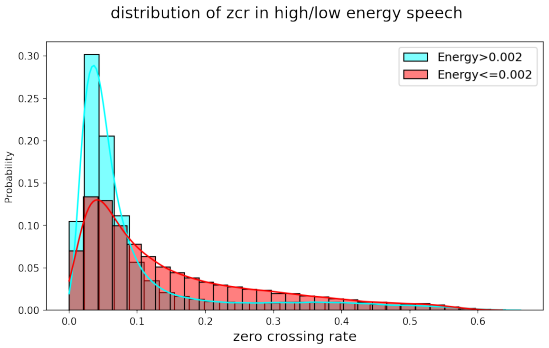

图 6: 开发集语音帧中的过零率分布

可以发现：第一，低能量帧中，语音帧比静音帧更有可能拥有比较大的过零率（图5中在高过零率区域语音比重很大）；第二，语音帧中，低能量帧相较于高能量帧更可能具有较大的过零率（图6中高过零率区域，低能量帧比重很大）。此外，随机抽取样本，观察其过零率和 VAD 标签如图7，可以发现高过零率音段基本都是语音。这些结论从很大程度上说明过零率确实可以用于鉴别低能量音段中的语音和静音。

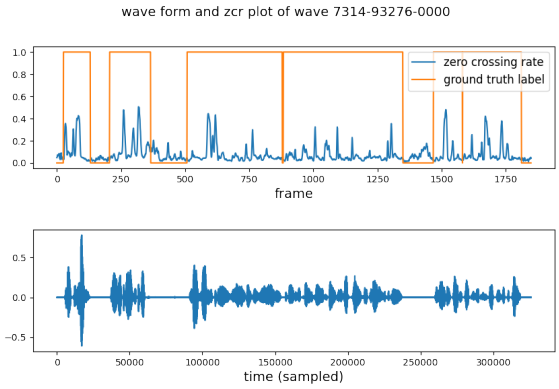

图 7: 某个样本音频的过零率、波形和 *VAD* 标签

2. 算法描述

2.1. 决策树模型构建和展示

基于上一节对短时能量和过零率进行的分析，我们可以知道可以先对能量进行一个阈值分类，然后对低能量帧再进行过零率分类。这个思想和机器学习中的决策树算法不谋而合。在这个问题上其有几大优点：

1. 可解释性强，分类依据与理论符合度高；且模型简单，是分段线性的分类器。

2. 训练速度快，计算量很小。

3. 易于移植到不同的数据集上，即在超参数确定的情况下，不依赖手动调节分类参数就能自行拟合不同情况下的数据集。

然而不恰当使用决策树将会导致严重的过拟合现象，因此需要对决策树的**最大层数**和**最大叶节点数**进行严格限制。因而本次 Project 使用 sklearn 中的 tree.DecisionTreeClassifier 作为分类器，以开发集数据训练，以帧级别的准确率、AUC、EER 等数值以及对决策树结构的人为判断为指标调整树参数，最终决定设置最大深度和最大叶节点数目均为 5，节点分裂指标为**熵 (entropy)**。得到的决策树结构可视化如图8。

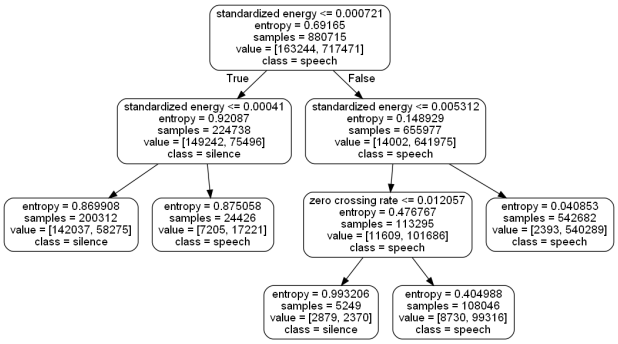

图 8: 某次训练得到的决策树结构

从图8中可以看出决策树模型的决策过程和理论比较符合，其只在能量非常小，或者能量不太小但是过零率低的情况下判断为静音；那些抑或能量高，抑或能量低但是过零率高的帧都被判为语音，这与我们的认知相匹配。其本质上是一个阈值分类器，但是通过最大化划分纯度的方法选取了最佳阈值。

此外我们还尝试了其他传统的简单分类器，诸如 Logistic 回归、贝叶斯分类器、支持向量机等，或效果不如人意，或训练速度太慢（如支持向量机），结果见表1。决策树是其中最为清晰和简单的方法。

2.2. 后处理：预测结果平滑化

由于决策树模型仅对每一帧单独分类，并不考虑帧之间所存在的隐含关系，所以时常导致分类结果短时内变化过大的问题。于此我们可以对预测结果进行平滑的后处理操作。本次 Project 选择**均值平滑**，即用一个 uniform kernel 对预测序列进行卷积，然后按照 0.5 的阈值重新分类。其中对音频边界帧需要做特殊处理。这个方法能有效减少语音段中突然出现的 0，或静音段中突然出现的 1。至于平滑窗长，我们进行了 grid search，遍历窗长以使 AUC、EER 最优，结果如图9所示。

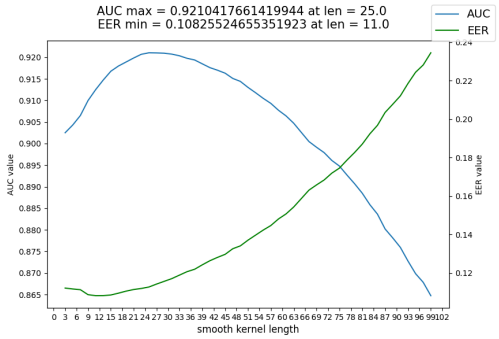

图 9: 平滑窗长和 *AUC*、*EER* 的关系

则选择 21-27 之间的平滑窗长将会比较合适。某个样本的平滑结果如图10所示，可见平滑操作起到了应有的效果。从本质上说，均值平滑等价于一个简单的状态机，即在窗长的范围内，超过一半帧为语音则中间帧判为语音，少于一半则判为静音。

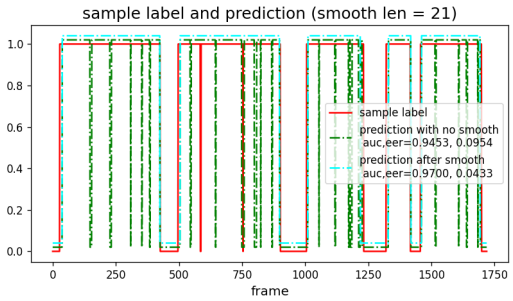

图 10: 某一样本平滑前后预测结果

综上所述，本次 Project 训练和预测的流程如算法1所示。

Algorithm 1: Project 1 全过程

// Begin training
$E, Z \leftarrow \{\}, \{\}$;
for *every audio* A_i *in develop set* **do**
  $A_i \leftarrow A_i - \text{average}(A_i)$;
  $A_i \leftarrow \text{Frame}(A_i)$;
  $E_i \leftarrow \text{ExtractEnergy}(A_i)$;
  $E_i \leftarrow (E_i - \min(E_i))/(\max(E_i) - \min(E_i))$;
  $Z_i \leftarrow \text{ExtractZCR}(A_i)$;
  $E \leftarrow E \cup E_i,\ Z \leftarrow Z \cup Z_i$;
end
Split training and testing set;
Train Decision Tree T by $\{E, Z\}, Y$;
// Begin Prediction
for *every audio* A'_i *in test set* **do**
  Do the same pre-processing as train;
  $\hat{Y}_i \leftarrow T.\text{predict}(E'_i, Z'_i)$;
  $\hat{Y}_i = \text{smooth}(\hat{Y}_i, \text{some smooth length})$;
  output $\hat{Y}_i$;
end

3. 实验结果

我们主要利用 utils 中的 EER 和 AUC 指标进行评测，并计算帧上的准确率 Acc（正确帧数/总帧数）。图11给出了其中一次训练模型后，平滑前后模型在开发集上的 AUC、EER 以及准确率，并绘制了相应的 RoC 曲线。选用的平滑长度为 25。由于本次 Project 预测输出的即为 0 或 1，并不输出概率值，因此 RoC 曲线仅为两段折线。EER 在 RoC 曲线中表现为曲线与直线 TPR+FPR= 1 的交点对应的 FPR 值。

表 1: 不同传统机器学习算法的实验结果

Model Name	AUC	EER	Acc
Logistic Regression (from sklearn.linear_model)	0.7000	0.5495	0.8570
Naive Bayes (from sklearn.naive_bayes)	0.8244	0.3302	0.7271
KNN (sklearn.KNeighborsClassifier)	0.8981	0.1615	0.9355
Decision Tree (from sklearn.tree)	**0.9241**	**0.1063**	**0.9432**
SVM (from sklearn.svm)	*Training unbearably slow*		

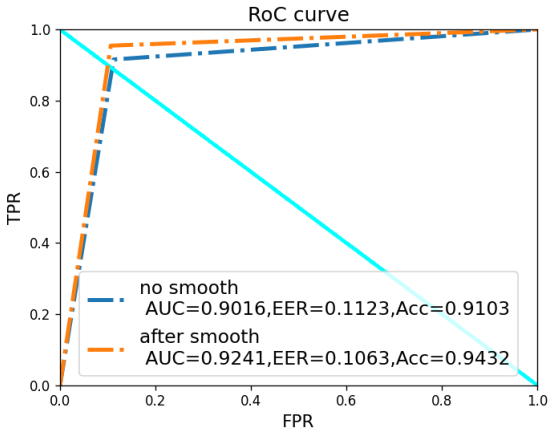

图 11: 平滑前后在开发集上的评测结果

作为对比，我们尝试了不同的传统简单分类器模型，结果如表1所示。决策树在其中各项指标均处于最优。

可见此模型对帧级别的 VAD 任务具有不错的表现，且平滑处理对于性能提升有显著的效果。另外此方法易于解释和移植，所需要的超参数仅有平滑长度和决策树的结构，易于调优。本次 Project 让我对音频处理和 VAD 有了基本的认识，在 Project2 中将尝试加入滤波等预处理，并提取更为复杂的频域特征，利用更复杂的模型及更合适的评测指标以求得更好的语音活性检测结果。