语音识别

基础

分帧:

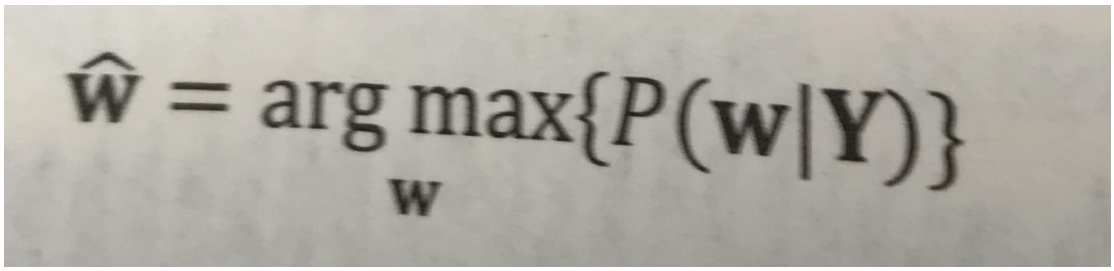
一段语音往往持续数秒到几个小时。但是人说一句话的时候, 一个词持续的时间往往比较短, 仅仅有零点秒，词由音素组成。因此每个音素持续的时间更短。
分帧的目标是：获取短时平稳特征。在这个很短的时间内，音素是一个音素，而不能包含多个音素。一般设置为 25ms

帧移:

音素总有会转移的时候, 因此总有一个 25ms 的帧会包含从一个音素到另外一个音素的转移。所以每隔 10ms 取一帧。

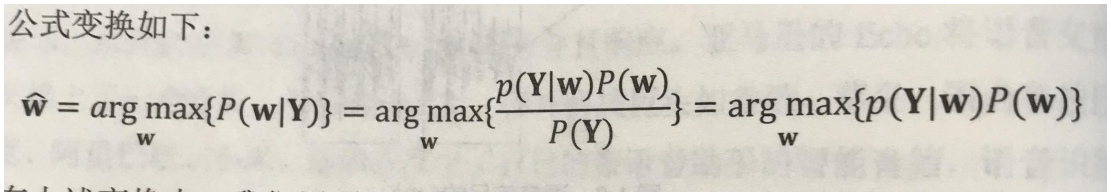
目标函数

Y 是输入的音频信号
w 是输出的词序列



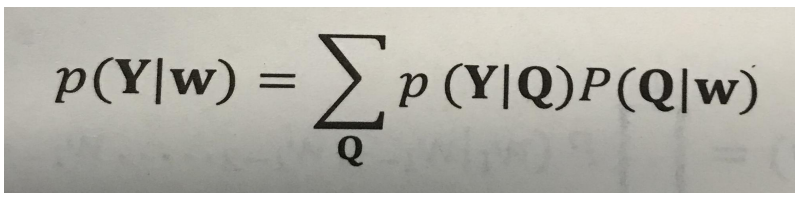
$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}}\{P(\mathbf{w}|\mathbf{Y})\}$$

使用贝叶斯公式



$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}}\{P(\mathbf{w}|\mathbf{Y})\} = \arg\max_{\mathbf{w}}\{\frac{p(\mathbf{Y}|\mathbf{w})P(\mathbf{w})}{P(\mathbf{Y})}\} = \arg\max_{\mathbf{w}}\{p(\mathbf{Y}|\mathbf{w})P(\mathbf{w})\}$$

w 是一系列的词，词由音素序列 Q 构成



$$p(\mathbf{Y}|\mathbf{w}) = \sum_{\mathbf{Q}} p(\mathbf{Y}|\mathbf{Q})P(\mathbf{Q}|\mathbf{w})$$

单词 wl 发音为 q(wl)的序列的概率

$$P(\mathbf{Q}|\mathbf{w}) = \prod_{l=1}^{L} P\left(\mathbf{q}^{(w_l)}|w_l\right)$$

单音子模型

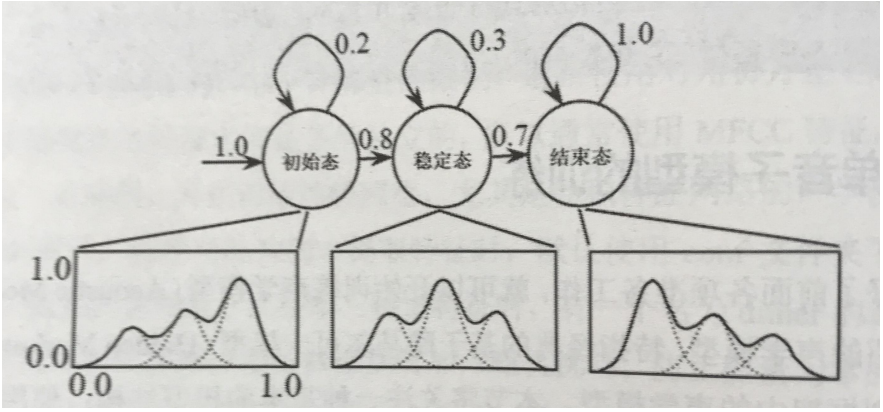
一个音素的实际发音，与左右相邻或相近的音素无关。缺点是：这个假设不符合实际，实际发音往往会有协同发音的现象

三音子模型

speak 这个英文单词在词典中对应的发音序列为 [s p i: k]，而第二个音素[p]的发音因为其左临音素[s]而发生浊化，实际应被读为[b]；
get down 发音序列为 [g e t d au n],实际发音时，[t]因为其后续音素为[d] 所以产生吞音，出现不发音的情况。
oh my god. [d]因为处于句子末尾而被吞音

hmm 三状态

一个单音素，或者一个三音子，就是一个 hmm。一个 hmm 一般会设 3 个隐状态，但是也有会设 2 个隐状态和 5 个隐状态的，但是常用的就是这 3 个隐状态。
这三个隐状态如何理解：
按照传统的 hmm 来说，观测序列已知，来做解码的时候，往往使用维特比算法，求得最优的隐状态序列。对语音来说输入的是帧序列，而一个音素往往是由多个帧才能组合完成。因此就有了某些帧被解码为初始隐状态, 有些帧被解码为稳定态隐状态, 有些是结束态隐状态。初始态、稳定态、结束态 分别对应一个混合高斯模型，混合高斯模型里面的超参数可以自定义



第 t 帧语音特征 ot 在第 i 个状态 si 上的声学得分

$$\mathrm{AmScore}(t,i) = \log P(o_t|s_i)$$

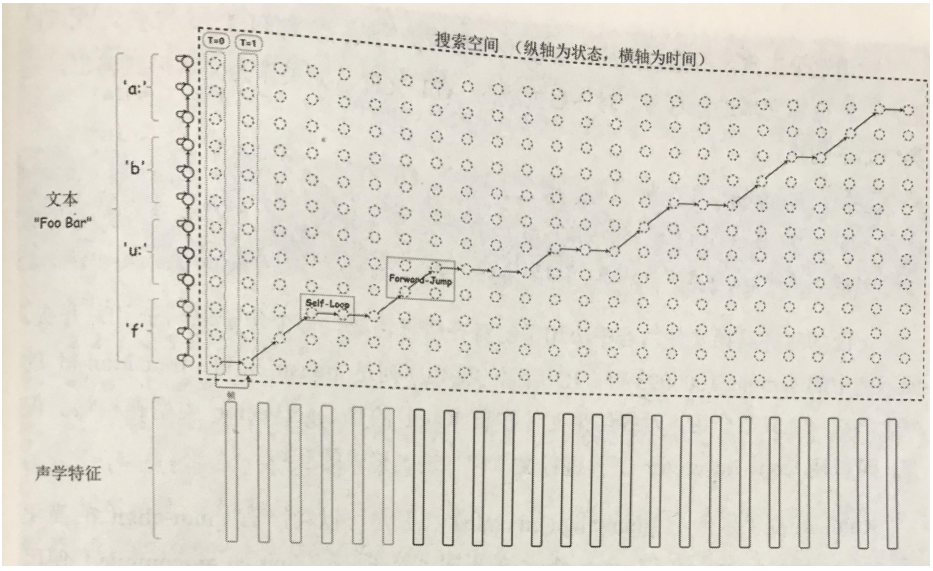
$$\log P(o_t|s_i) = \log \sum_{m=1}^{M} \frac{c_{i,m}\exp(-\frac{1}{2}(o_t-\mu_{i,m})^T(\Sigma_{i,m}^{-1})(o_t-\mu_{i,m}))}{(2\pi)^{D/2}|\Sigma_{i,m}|^{1/2}}$$

为了降低模型参数量，通常协方差矩阵为对角阵

$$\Sigma_{i,m} = \mathrm{diag}(\sigma_{i,m})$$

单音子 gmm-hmm 解码

思考：假如训练好了一个 gmm-hmm 模型来做语音识别，最终的识别结果应该是什么样子的？
如下图所示：



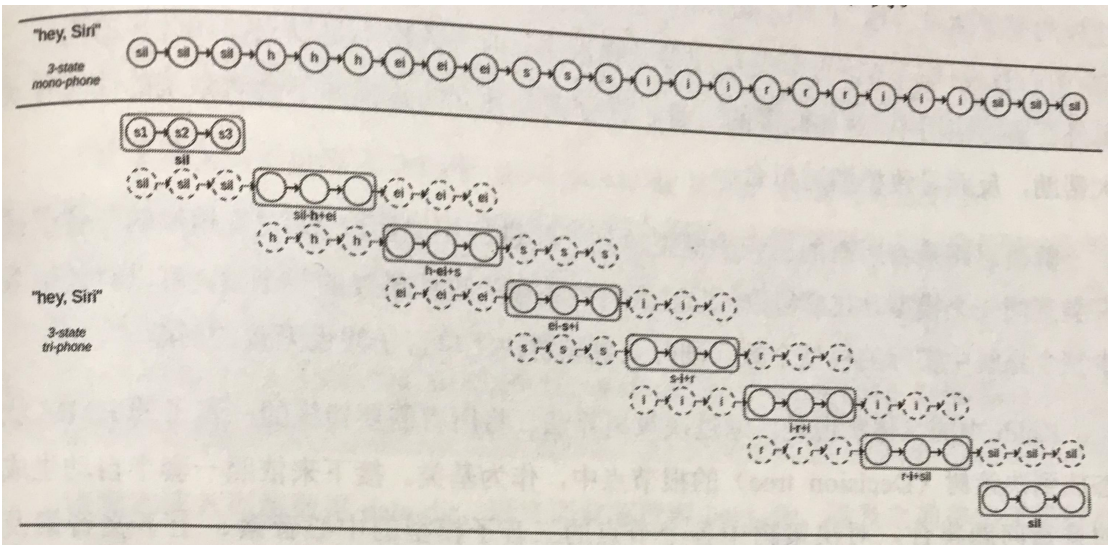
输入的是每一帧的特征，最终的解码是一条路径。
每一个音素都是由一个三个状态的 hmm 模型表示，每一帧都会被解码成某个音素对应的某个隐状态。

self-loop:一个 hmm 中的三个隐状态，自己往自己跳转。
例子：
o1 o2 o3 帧序列被解码为：
f-稳定态 f-稳定态 f-稳定态
这就是 self-loop

forward-loop:由一个音素的结束态转移到另外一个音素的开始态

三音子 gmm–hmm 解码

相对于单音子，多音子更符合现实中的特点
在解码的时候仅仅是把三音子当成单音子就可以了，没有其他区别



参数爆炸问题：

对英文来说，有 40 个单音子，如果做成 3 音子，则会产生 40*40*40=64000 个三音子。模型训练起来速度慢，并且需要大量的数据。因此会采用对三音子进行聚类，同一类的三音子共享参数。

gmm–hmm 训练

传统的 gmm-hmm 训练通过 EM 算法求解
而 kaldi 里面的 gmm-hmm 采用维特比训练进行训练（仅仅是提速了，效果没有打折扣，待更新）

三音子聚类裁剪

一个三音子 hmm 模型可以表示为：　{L}-{C}+{R}.{S}
其中 L: 三音子模型的左上文
　　C: 三音子模型的中音素
　　R: 三音子模型的右下文
　　S:　hmm 模型的三个状态 {S}={1,2,3}
40*40*40*3 个 hmm 状态需要建模，需要构建这么多个高斯混合模型。40*40*40 个 hmm 模型需要建模。

解码器