语音识别系列之决策树

在近日准备的深蓝学院的 GMM-HMM 这一讲的课程中，因为课程篇幅和时长的限制，关于决策树这一部分仅是做了比较浅的介绍，但决策树是基于 HMM 语音识别中的一个难点和要点，所以，这里通过文章做进一步的介绍，方便更多的同学和读者做进一步的学习。

本文假定读者已经有了 GMM、HMM、音素等和基于单音素的 GMM-HMM 语音识别相关的基本知识。

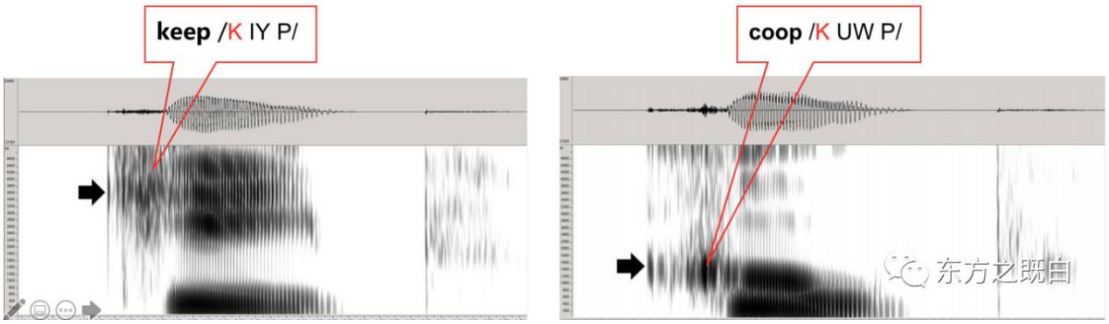
三音素

基于单音素的语音识别系统已经可以完成基本的大词汇量连续语音识别的任务，但该系统存在如下缺点：

1.**建模单元数目少。**典型的，一般英文系统的音素数量在 30～60 个，中文系统音素数目在 100 个左右，面对成千上万小时的训练数据，这么少的建模单元数难以做到精细化的建模，也就难以达到更好的识别率。

2.**音素发音受其所在上下文的影响**，一句话或者一个词中的音素并不是孤立发音，而是整体协同发音，典型的如英文中的连读如"not at all"，吞音如"first time"等等。

这里通过 keep /K IY P/和 coop /K UW P/发音的语谱图来进一步说明一下协同发音的问题，如下图所示，虽然两个单词的/K/是同一个音素，但从语谱图上看，前者特征集中在高频，后者则集中在低频，具有明显的区分性。



那如何解决这个问题？考虑到第二点，我们可以考虑音素所在的上下文(context)进行建模，一般的，考虑当前音素的前一个（左边）音素和后一个（右边）音素，称之为**三音素**，并表示为 **A-B+C** 的形式，其中 B 表示当前音素，A 表示 B 的前一个音素，C 表示 B 的后一个音素。

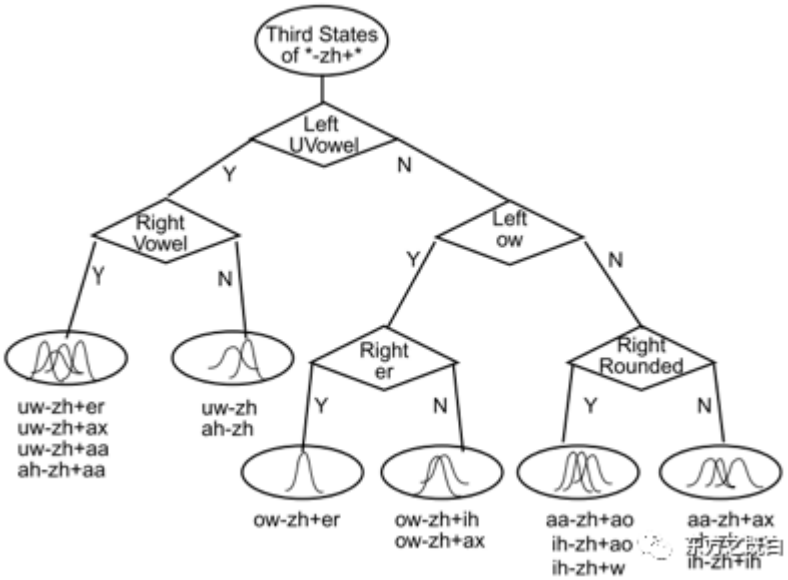
假设音素集有 N 个音素，则共有 N 的三次方个三音素，也就同时解决了建模单元数少的问题。但是，却有如下新问题：

1. 建模单元数又太多了。100 个音素的话，则会有 100 万个三音素，加上每个音素的 HMM 三状态和状态所对应的 GMM 参数，参数量太大了。
2. 数据稀疏问题。有的三音素在训练语料中出现次数很少，则对应的训练数据就很少。
3. unseen data 问题。有的三音素在训练数据中压根就没有出现，如比较极端的 Z-Z+Z 这样的组合，但识别时却有可能出现，这时如何描述未被训练的三音素及其概率。

决策树的引入正是为了解决如上问题。

决策树

下面直接给出决策树的结构（如下图所示），然后回过头看决策树是否解决了我们上述的几个问题，再讨论如何构建一个决策树。



从图中，我们可以看到，决策树有如下几个要点：

1.**决策树是一棵二叉树**（图中无论是椭圆形、菱形都可以当作二叉树的一个节点）。

2.**每个非叶子节点上都有一个问题**（菱形所示，Left UVowel 是问你当前三音素的左边的音素是不是一个 UVowel，Right Vowel 是问你当前三音素的右边的音素是不是一个元音音素），每个三音素对于该问题都会有一个 Yes 或 No 的的答案，那么对所有的三音素来讲，该问题会把所有三音素分成 Yes 集合和 No 集合。

3.**决策树的叶子节点是相近（绑定）三音素的集合**（从根节点经过一些列的问题，他们最终被分到了同一个节点）。

4.**决策树建立的基本单元是状态**（图中，根节点是说这是以 zh 为中心音素的三音素的第三个状态的决策树，第一个状态和第二个状态也都有各自独立的决策树，举例来说，比如 a-zh+c,o-zh+c 假设它们的第一个状态可能绑定在一起，但并不代表说它们的第二个、第三个状态也绑定在一起，因为 2/3 状态有各自独立的决策树），这一点其实和解决上述三个问题关系不大，但这么做也是为了更精细化的建模。

现在回过头看一下，决策树是否能解决上述三个问题。

1.建模单元数过多的问题，可以通过控制决策树的叶子节点数量解决。

2.数据稀疏问题通过相近三音素的绑定在一个叶子节点上解决，一个叶子节点对应一个 GMM，相比之前单个三音素的状态各自对应一个 GMM，训练数据增加。

3.unseen data 问题。通过决策树，通过非叶子节点的问题，所有的三音素最终都可以被分类到一个叶子节点上，从而对应该节点所对应的 GMM，自然的，该问题也解决。以上图决策树举例，对于该决策树中未出现的三音素 zh-zh+zh，进入该决策树先问"Left UVowel"，不是，走右边，再问"Left ow"(左边是不是 ow 这个音素)，不是，继续走右边，继续问"Right Rounded"（右边的音素是不是口型是圆的)，也不是，再走右边到达叶子节点，最终 zh-zh+zh 的第三个状态与 aa-zh+ax eh-zh+ax ih-zh+ih 的第三个状态绑定在一起。这样，即使 zh-zh+zh 从未在训练语料中出现过，通过决策树，我们最终将它绑定到一个与之相近的叶子节点上。

可以看到，上述的三个问题，通过决策树都可以完美的解决。

问题集

所有问题的集合称之为问题集。问题和音素之间的相近性有关。例如英文系统中的问题可以是：

- 爆破音(Stop): B D G P T K
- 鼻音(Nasal): M N NG
- 摩擦音(Fricative): CH DH F JH S SH TH V Z ZH
- 流音(Liquid): L R W Y
- 元音(Vowel): AA AE AH AO AW AX AXR AY EH ER …
- …

可以看到，这是一个关于音素的基本分类，每个类别都可以作为一个问题。这里每个类别的子集也可以作为一个问题，甚至是一个音素也可以作为一个问题（问题就是你是不是这个音

素)。最终的问题是音素的相近性加上音素的位置（如上面的 Left/Right）信息组合而成。问题集的生成方式有两种。一种是语言学家定义，例如上面举例的音素种类划分其实就是语言学家定义的；二是自动生成，语音识别工具 Kaldi 中通过对音素进行聚类自动生成决策树，后续作者会再写一篇文章“语音识别系列之 Kaldi 中的决策树”，其中会介绍 Kaldi 中如何自动生成问题集。

决策树的生成

有了决策树的最终形式和问题集等的概念，那下面的问题就是如何生成决策树？

没有任何初始系统直接构建决策树是不可能的。一般的，我们考虑如何基于已有单音素的系统上构建决策树。也就是目前已有条件：

1.单音素系统

2.问题集

通过单音素系统，我们可以得到单音素各个状态所有对应的特征集合，做一下上下文的扩展，我们可以得到三音素的各个状态所有对应特征集合。例如，假设目前单音素得到一个状态对齐序列如下第一行所示，则其三音素状态序列通过上下文可以扩展为如下第二行所示，其中#表示边界，最后一位数字表示第几个状态。通过这样的方法，我们可以把训练数据上所有的单音素的数据对齐转为三音素的对齐。

```
 K1        K1      K2      K3       K3      IY1      IY2      IY2       IY3      P1      P2      P3
#-K+IY1 #-K+IY1 #-K+IY2 #-K+IY3 #-K+IY3 K-IY+P1 K-IY+P2 K-IY+P2 K-IY+P3 IY-P+#1 IY-P+#2 IY-P+#3
```

考虑到上面描述的决策树的最终的粒度是一个音素的状态，并且决策树的构建是一个自顶向下的过程。比如对于 zh 的第三个状态来讲，我们的初始条件类似决策树这张图中的根节点 "Third states of *-zh+*"，下面所面临的问题是如何一步一步分裂该节点，而分裂该节点，其实就是如何从问题集中选择问题，能够使相近的三音素分类到相同的节点上。

现在假设根节点的所有三音素的对应的特征服从一个多元（多维特征）的单高斯分布，则可以计算出该多元单高斯分布的均值和方差，则可以计算该节点任意一个特征在该高斯上的似然（下图第一个公式），则所有数据在该高斯上的似然并取 log 为(下图第二个公式)。

- Sample $S = (x_1, \ldots, x_m) \in (\mathbb{R}^N)^m$.
- Diagonal covariance Gaussian:

$$\Pr[x] = \frac{1}{\prod_{k=1}^{N}(2\pi\sigma_k^2)^{1/2}} \prod_{k=1}^{N} \exp\left(-\frac{1}{2}\frac{(x_k - \mu_k)^2}{\sigma_k^2}\right).$$

- Log-likelihood for diagonal covariance Gaussian:

$$\begin{aligned} L(S) &= -\frac{1}{2}\sum_{i=1}^{m}\left[\sum_{k=1}^{N}\log(2\pi\sigma_k^2) + \sum_{k=1}^{N}\frac{(x_{ik} - \mu_k)^2}{\sigma_k^2}\right] \\ &= -\frac{1}{2}\left[m\sum_{k=1}^{N}\log(2\pi\sigma_k^2) + m\sum_{k=1}^{N}\frac{\sigma_k^2}{\sigma_k^2}\right] \\ &= -\frac{1}{2}\left[mN(1 + \log(2\pi)) + m\sum_{k=1}^{N}\log(\sigma_k^2)\right]. \end{aligned}$$

同样的，假设通过某个问题将该节点的三音素所对应的特征分成两部分(l 和 r)，则这两部分的似然和为(下图第一个公式)，则分裂前和分裂后的似然变化(增益)为 D=L(Sl)+L(Sr)-L(S)，似然增益越大，说明分裂后的两部分数据之间的差距越大，则越应该使用两个单独的 GMM 分别建模。问题集中的每个问题都可以将该节点分成两部分，哪种分法（哪个问题）最好，**当然是似然增益最大的分法最好。**将 D 的公式进行简化，则最优分法的问题称为最优问题，表示如下（下图第二个公式）。这样就解决了如何从问题集中选择问题进行分裂。

- Log-likelihood difference:

$$L(S_l) + L(S_r) = -\frac{1}{2}mN(1+\log(2\pi)) - \frac{1}{2}\left[m_l \sum_{k=1}^{N} \log(\sigma_{lk}^2) + m_r \sum_{k=1}^{N} \log(\sigma_{rk}^2)\right].$$

- Best question:

$$q^* = \underset{q}{\operatorname{argmin}}\left[m_l \sum_{k=1}^{N} \log(\sigma_{lk}^2) + m_r \sum_{k=1}^{N} \log(\sigma_{rk}^2)\right],$$

with $$\sigma_{lk}^2 = \frac{1}{m_l}\sum_{x\in S_l} x_k^2 - \frac{1}{m_l^2}(\sum_{x\in S_l} x_k)^2$$

$$\sigma_{rk}^2 = \frac{1}{m_r}\sum_{x\in S_r} x_k^2 - \frac{1}{m_r^2}(\sum_{x\in S_r} x_k)^2.$$

在根节点一份为 2 后，下一步如何选择节点、如何选择问题进行下一个分裂可以继续使用似然增益的准则，递归的执行该算法直至达到一定条件终止，通常是分裂已经达到一定数量的叶子节点，或者似然增益已经低于一定阈值。

总结一下决策树的生成流程：

1. 初始条件（单音素系统对齐，一个根节点）
2. 选择当前所有待分裂的节点、计算所有问题的似然增益，选择使似然增益最大的节点和问题对该节点进行分裂。
3. 重复 2 直至算法满足一定条件终止。

参考资料

1.Decision Tree-Based State Tying For Acoustic Modeling

2.http://www.danielpovey.com/files/Lecture3.pdf