语音识别微入门

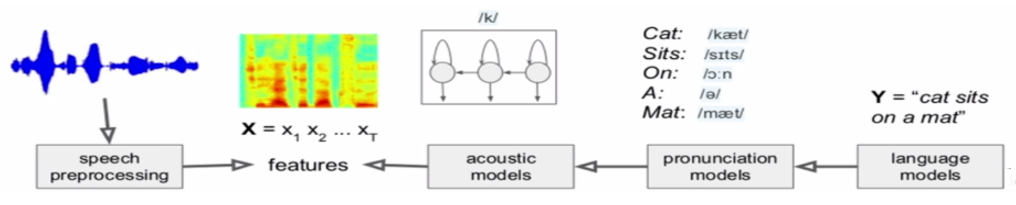
语音识别，通俗来讲，就是将语音信息转化为文字文本信息。

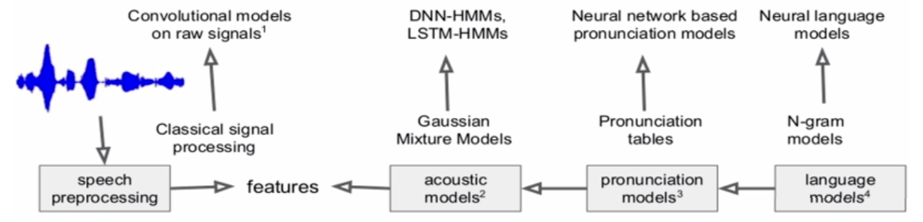
语音信息，通过音频采集装置，以时序声波作为其显示表现形式，可以看做音频在时间域上的表现形式（通过一般的频域变换，如傅里叶变换，也可以将时域音频信息转变为频域音频信息）。因而，其基本形态可以看做是序列数据（与nlp语言序列、推荐历史序列相对应）；文字文本信息，就是将语音信息“翻译”后对应的文字序列，其中“翻译”的过程也就是语音识别的过程。

语音识别流程大致与数据挖掘流程一一对应

数据采集 -> 数据预处理 -> 特征工程 -> 模型建立 -> 结果分析

音频采集 -> 音频处理 -> 特征构建 -> 声学模型+语言模型 -> 识别评价





传统的语音识别流程（上）和基于深度学习的语音识别流程（下）

音频采集

音频采集主要由专业的设备采集得到，如麦克风等。具体地，为了后续的处理更加方便快捷，最好具有降噪、保持音频清晰度和保真度的性质，最大限度保留音频的音质且具有稳定性。

音频处理

得到了对应的音频序列后，数据是以声波的形式展现的，一般是利用ADC模数转换器将模拟信号转换为表示一定比例电压值的数字信号，即可得到对应的音频数字信号。

然后再对数字信号做进一步的处理：VAD（语音激活检测），可以把一段音频信号中有效的语音序列给截取出来，也相当于去噪声处理。

在VAD中，

SNR：signal-noise ration，即信噪比，指一个电子设备或者电子系统中信号与噪声的比例。信噪比的计量单位是dB，其计算方法是10lg(Ps/Pn)，其中Ps和Pn分别代表信号和噪声的有效功率，也可以换算成电压幅值的比率关系：20Lg(Vs/Vn)，Vs和Vn分别代表信号和噪声电压的“有效值”。

（1）当SNR较大时，可以使用一些简单的指标去评价一段语音的有效性，如过零率ZCC（一段信号中数字信号穿过零轴的比率）和信号短时能量STE（一段信号中数字信号的平方和）。一般来说，真正的语音信号，即非噪声信号的ZCC以及STE都较高，因此可以设置阈值将语音序列进行分割，得到有效语音序列和噪声序列，再分别处理；

（2）当SNR较小时，这些简单的指标已经不起区分作用了，因此需要更精细更复杂的方法，可以使用深度神经网络（DNN）做激活检测，在该方法下，语音激活检测即为一个分类问题。即构造一些以音频信号序列作为输入以及其有效性作为输出的数据集来进行训练。

特征构建

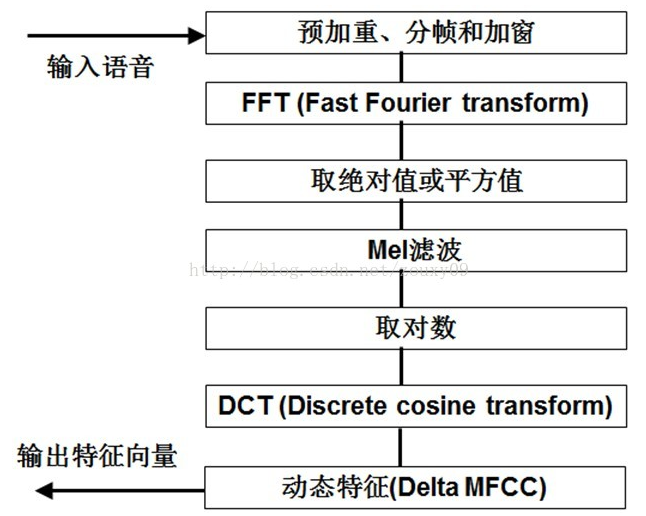
基本的有效语音段被提取完成后，就需要对数字信号进行特征构建了，因为未经处理的raw数字信号难以提取语音识别的有效信息。

常见的语音数字信号的特征包括线性预测倒谱系数LPCC以及梅尔频率倒谱系数MFCC：前者是根据声管模型建立的特征参数，是对声道响应的特征表征；后者则是基于人的听觉特征提取出来的特征参数，是对人耳听觉的特征表征。

MFCC（Mel Frequency Cepstrum Coefficient, MFCC）

声道形状（舌头、牙齿等）决定了音素的描述，而声道的形状在语音短时功率谱的包络中显示出来，而MFCC（Mel Frequency Cepstrum Coefficient, MFCC）能够比较准确地将该包络表征出来，是最常用的语音识别特征。

其基本流程为



预加重：xt=xt-kxt-1，k一般取0.97，对语音的高频部分进行加重，去除口唇辐射的影响，增加语音的高频分辨率；

分帧和加窗：由于语音信号具有短时平稳性，因此可以采用可移动的有限长度的窗口进行加权的方法来实现短时分帧处理，且一般的分帧方法为交叠分段的方法，前一帧和后一帧的交叠部分称为帧移，帧移与帧长的比值一般为0~0.5；

梅尔频率倒谱系数：考虑到了人类的听觉特征，先将线性频谱映射到基于听觉感知的Mel非线性频谱中，然后转换到倒谱上。将普通频率转化到Mel频率的公式是：



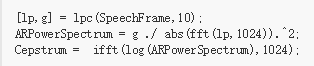
倒谱（Cepstrum）分析：一种信号的傅里叶变换经对数运算后再进行傅里叶反变换得到的谱。

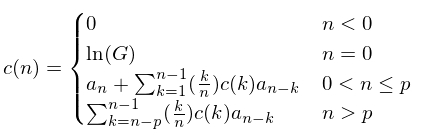
频谱图的峰值就表示语音的主要频率成分，这些峰值被称为共振峰（formants），而共振峰就是携带了声音的辨识属性（就是个人身份证一样）。频谱的包络（Spectral Envelope）就是一条连接这些共振峰点的平滑曲线。将原始的频谱由两部分组成：包络（低频成分）和频谱的细节。这里用到的是对数频谱，所以单位是dB。那现在我们需要把这两部分分离开，这样我们就可以得到包络了。对频谱做FFT。在频谱上做傅里叶变换就相当于逆傅里叶变换Inverse FFT (IFFT)。需要注意的一点是，我们是在频谱的对数域上面处理的。这时候，在对数频谱上面做IFFT就相当于在一个伪频率（pseudo-frequency）坐标轴上面描述信号。

实际上，可以直接把倒谱频域直接求出来再添加低通滤波即可求出包络。实际逆变换一般是通过DCT离散余弦变换来实现，取DCT后的第2个到第13个系数作为MFCC系数。

LPCC（Linear Prediction Cepstral Coefficients）

与MFCC区别在于没有Mel频率转换以及在频域信号进入ln前的计算公式不一样，主要利用了LPC（Linear Prediction Coefficients）进行计算，LPC即是AR算法，利用前面k个时间节点的信号值回归当前的信号值，可以通过最小化回归误差进行系数的求解。而LPCC即通过下面操作进行求解（lp：系数；g：预测误差方差）；也可以通过最下面的公式将LPC转换为LPCC。





基于滤波器组的Fbank特征（Filterbank）

亦称MFSC，Fbank特征的提取方法就是相当于MFCC去掉最后一步的离散余弦变换，跟MFCC特征相比，Fbank特征保留了更多的原始语音数据。

随着深度学习的不断发展，直接从raw\_input中通过深度学习算法如DNN、RBM、CNN、RNN等学习有效且有区分性的特征表示。

声学模型+语言模型

基本的语音识别模型由两部份构成：声学模型（AM，Acoustic Model）和语言模型（LM，Language Model），这是由贝叶斯公式P(W|S)=P(S|W)\*P(W)/P(S)，分母可以利用正比于去掉得出的结论，前面部分可以看做是AM，后面部分可以看做是LM。

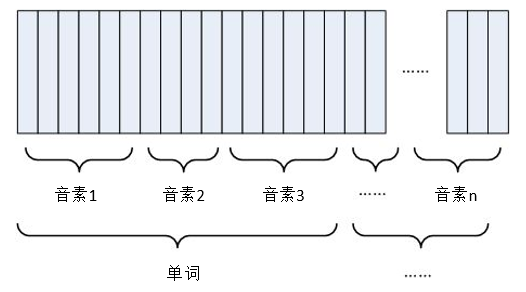
声学模型

该模型的作用是计算给定文字序列时语音序列的概率，进一步将分窗后的语音序列进一步组合成声学符号，如音节（syllable）或者更小的颗粒度音素（phoneme），记为Q。

P(S|W)=argmax(W)∑P(S,Q|W)=argmax(W)∑P(S|Q)P(Q|W)≈argmax(W,Q)P(S|Q)P(Q|W)。

乘积两部分分别为声学部分和字典部分，可以以连乘的形式进行计算。

为了计算声学部分，假设Q是音素，那么我们需要对音频信息进行音素划分，一般是将多帧合并成一个音素，除了单个音素表示，还可以使用三音子模型（考虑前后各一个音素，这里分为Word-internal和Cross-word，区别在于不同词语之前是否分享音素），而多个音素可以组合成一个词语（对应字典部分 ）。



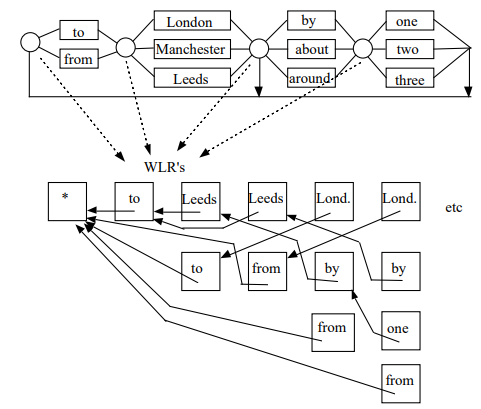
求解声学部分，需要先知道各帧对应的状态，传统的做法将会用到GMM+HMM的做法：使用GMM建立隐状态->观察变量的概率模型（不同隐状态下的概率是混合高斯模型），HMM是可以以音素作为基本状态（对单词建立HMM模型），也可以对每个音素建立HMM模型，具体介绍在Appendix。进一步发展，有将GMM模块用深度学习代替的算法，如DNN/RNN/CNN+HMM等

单词的语音识别（Isolated Word Recognition）

此时我们只需要使用最基本的HMM，以音素和语音特征作为HMM的基本状态以及观察序列，针对每个词建立一个GMM-HMM模型，此时转移概率需要根据单词的音素字典进行相邻以及自环定义，应用传统的EM算法以及维特比算法即可求解参数以及对应的音素序列。

连续语音识别（Continuous speech recognition）

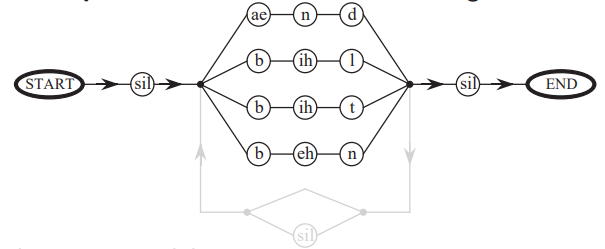
（使用了Embedding Training嵌入式训练，与提前标注语音的音素然后进行训练有大区别）使用多个HMMs构成的句子模型形成的网络来进行最大概率词序列的识别。与单个词的语音识别不同的是，在实施维特比算法进行隐状态路径的寻找时，普通的算法需要记录上一个最大的隐状态值，但是连续语音识别任务更关注的是词与词之间的分割位置，因此这里不记录隐状态，而是记录Word Link Records（WLR，由四部分构成：当前时间点下的logP；指向上一个WLR的路径id；刚识别的单词；可以追溯词边界的WLR的生成时间），有多少个词就会有多少个WLR。

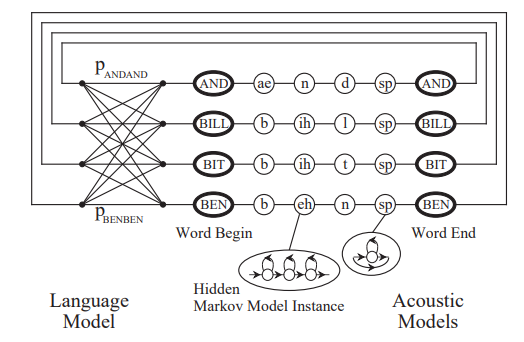


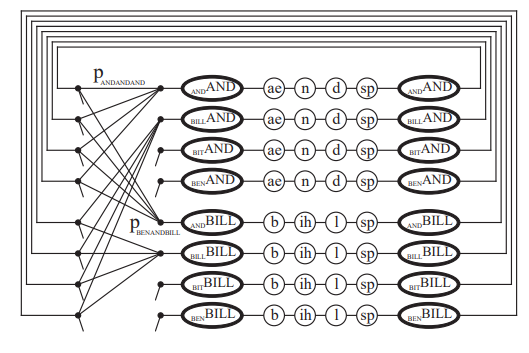
如果用传统的维特比算法，则需要在每个词位置遍历所有可能的词，花费的计算复杂度将很巨大，需要使用pruning后的算法：Beam Search（集束搜索）。

|  |
| --- |
| Beam Search流程（各个状态处于激活或者未激活状态） |
| 初始化时各个入口状态将会初始化为激活状态，其他为未激活状态。  1）对于每个活动状态  a.正常扩展word内部路径  b.记录任意状态下logP的最大值，称为gMax  2）令logP < gMax - B的所有状态转为未激活，其中B为beam宽度  3）扩展路径，通过word外部链接，只有当后续状态都在beam内，即logP< gMax - B  4）重新激活所有具有扩展到HMM入口状态的路径的状态。 |

语言模型也可以集成在连续语音识别中，只需在每个词识别好后再添加对应的unigram或者bigram或者trigram语言模型。







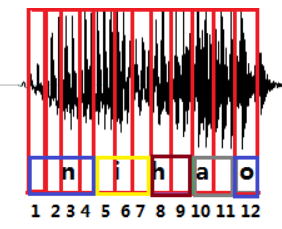
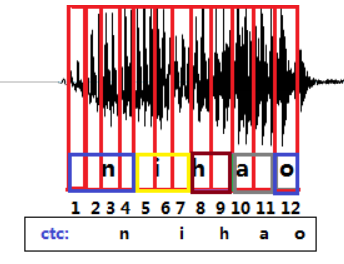
除了上面利用维特比或者beamSearch算法进行搜索外，还有利用WFST（Weighted Finite-State Transducers）进行语言模型和发音词典模型的搜索，WFST把整个语音识别的解码过程细分成四个部分，称为HCLG解码，即声学模型 -> 上下文相关音子模型 -> 发音词典模型 -> 语言模型，四个部分。

引入深度学习的语音识别模型

深度学习作用在声学模型中，第一种是把GMM-HMM中的GMM变为DNN/RNN/CNN等，利用了贝叶斯分析，将观察值的后验概率转化为似然概率。另外一种是端到端的深度学习模型，这种深度学习模型直接从音频数据出发，通过深度学习模型直接输出音素序列，而不需要数据的对齐。代表算法是CTC算法（Connectionist Temporal Classification）以及基于Attention的算法。

CTC算法

CTC是一种损失函数，用来衡量输入的序列数据经过神经网络之后与真实输出的差距，它不要求输出音素序列跟真是音素序列一一对齐，只需要输出正确按顺序排行的音素序列即可。损失函数为

=> 

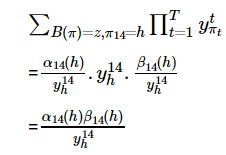
因此，在CTC算法下，不需要输出各个音素持续了多少帧，而是输出最后的音素序列。以RNN-CTC为例子，为加权分帧加窗后并已计算好MFCC特征的语音序列，只需将每一帧的MFCC输入到RNN后可以得到对应的音素softmax概率，输出最大概率音素即可。

在网络训练的时候，需要知道所有符合顺序的音素序列的概率之和。如果暴力写出所有符合顺序，那将会产生很大的复杂度，因此需要进行动态规划去求解该损失函数的梯度值。

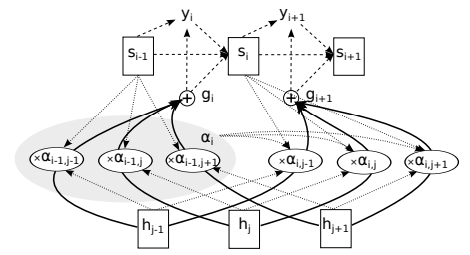
以”nihao”作为基本例子，下面的y是14时刻音素为h的概率，α代表14时刻为h的所有可能的前向路径的概率和，β代表14时刻为h的所有可能的后向路径的概率和，因此最终在14时刻出现音素h的所有路径的概率和可以由α和β表示，进一步可以对该概率求导并反向传播。

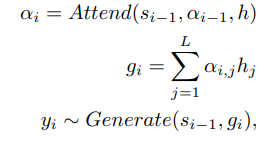




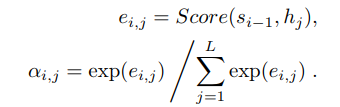


Attention机制









H是由MFCC语音特征用过深度学习网络，如双向RNN得到的输出隐状态，是所有输出的集合，然后再输入到上面的结构中，generate是RNN网络。其训练级别是音素级别的数据，因此在loss损失函数上选用了PER音素错误率，跟WER类似可以使用编辑距离进行计算。

进一步，有对CTC与attention模型进行结合的深度学习算法，流程跟上面一致。

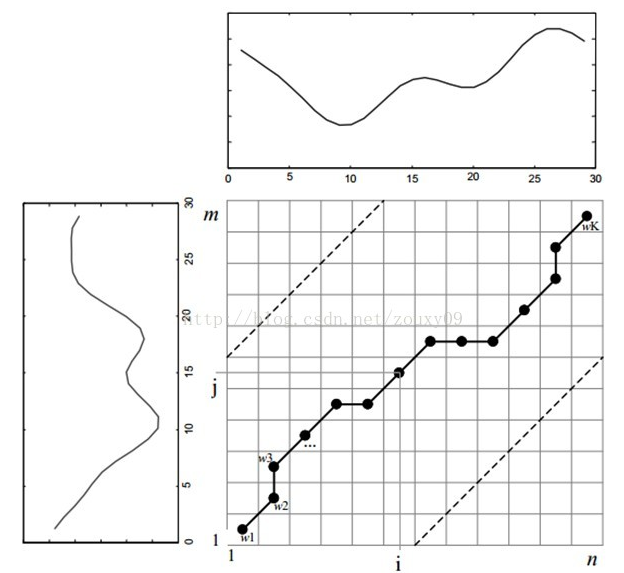
**其他语音信号处理知识**

动态时间规整（DTW, Dynamic Time Warping）

用于衡量两段不同长度的语音的相似程度或距离，主要思想是利用动态规划的算法对两条序列进行对齐并最终将距离计算出来。

假设两条语音进行特征构建后的长度分别为m和n。

1.首先构造出语音序列之间的两两距离矩阵，可以使用欧氏距离衡量对应的两个特征的距离；2.然后通过动态规划找到下图的路径（{wk=(i,j)}），为了使得两条序列的对齐是有效的，该路径含有三个约束。1）边界条件：w1=(1, 1)和wK=(m, n)，各部分的先后次序不可能改变，所选的路径必定是从左下角出发，在右上角结束；2）连续性：如果wk-1= (a’, b’)，那么对于路径的下一个点wk=(a, b)需要满足 (a-a’) <=1和 (b-b’) <=1。保证下一个对齐的点只能是相邻点并且各个点参与路径；3）单调性：如果wk-1= (a’, b’)，那么对于路径的下一个点wk=(a, b)需要满足0<=(a-a’)和0<= (b-b’)。保证W上面的点必须是随着时间单调进行的。



通过约束发现当前点只能往上、往右以及往右上角，因此动态规划下容易有递推式

，最终即可求出路径之和以及对齐结果。

基音周期（Pitch Detection）

人在发音时，根据声带是否震动可以将语音信号分为清音跟浊音两种。浊音又称有声语言，携带语言中大部分的能量，浊音在时域上呈现出明显的周期性；而清音类似于白噪声，没有明显的周期性。发浊音时，气流通过声门使声带产生张弛震荡式振动，产生准周期的激励脉冲串。这种声带振动的频率称为基音频率，相应的周期就成为基音周期。

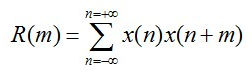
通常，基音频率与个人声带的长短、薄厚、韧性、劲度和发音习惯等有关系，在很大程度上反应了个人的特征。此外，基音频率还跟随着人的性别、年龄不同而有所不同。一般来说，男性说话者的基音频率较低，而女性说话者和小孩的基音频率相对较高。

基音检测的方法大致上可以分为三类：

1）时域估计法，直接由语音波形来估计基音周期，常见的有：自相关法、并行处理法、平均幅度差法、数据减少法等；

2）变换法，它是一种将语音信号变换到频域或者时域来估计基音周期的方法，首先利用同态分析方法将声道的影响消除，得到属于激励部分的信息，然后求取基音周期，最常用的就是倒谱法，这种方法的缺点就是算法比较复杂，但是基音估计的效果却很好；

3）混合法，先提取信号声道模型参数，然后利用它对信号进行滤波，得到音源序列，最后再利用自相关法或者平均幅度差法求得基音周期。

自相关函数，可以发现当m取信号周期的整数倍时，R达到最大，利用该性质可以求得周期；但是语音信号是非稳态信号它的特征是随时间变化的，但在一个很短的时间段内可以认为具有相对稳定的特征即短时平稳性，因此语音具有短时自相关性，这个时间段约5ms-50ms，对信号进行分帧处理，即可认为在每一帧中信号是稳态的，可以利用短时自相关函数，n起始，m为窗口长度，即帧数。

**Appendix**

**HMM简介**

HMM（Hidden Markov Model）即为隐马尔科夫模型，一般用于解决对含有隐藏状态的序列数据的建模问题。

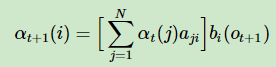
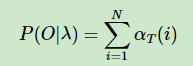
一般地，定义隐藏状态集合Q，个数为N；观测状态集合V，个数为M；长度为T的观察序列O以及对应的状态序列I，初始时刻下隐状态的分布概率Π。HMM还有两个重要的前提假设：（1）齐次马尔科夫链，任意时刻的隐状态只依赖于它的前一个隐状态，对应的转移概率定义为aij=P(it+1=qj|it=qi)，转移概率矩阵A与时间无关；（2）观测独立性假设，任意时刻的观测状态只依赖于当前时刻的隐状态，对应的观测概率bj(k)=P(ot=vk|it=qj)，观测概率矩阵B与时间无关。HMM可以表示为(A,B,Π)。

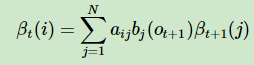
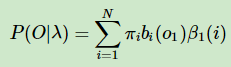
|  |
| --- |
| HMM观测序列生成的过程 |
| 输入的是HMM的模型λ=(A,B,Π),观测序列的长度T  输出是观测序列O={o1,o2,...oT}  1）根据初始状态概率分布Π生成隐藏状态i1  2) for t from 1 to T  a. 按照隐藏状态it的观测状态分布bit(k)生成观察状态ot  b. 按照隐藏状态it的状态转移概率分布aitit+1产生隐藏状态it+1所有的ot一起形成观测序列O={o1,o2,...oT} |

HMM的三个经典问题以及解决算法

（1）评估观察序列概率。即给定模型λ=(A,B,Π)和观测序列O={o1,o2,...oT}，计算在模型λ下观测序列O出现的概率P(O|λ)

利用前向后向算法（可以看做是动态规划算法）进行解决。如果直接使用暴力解法的话，复杂度会非常高（P(O|λ)=∑P(O,I|λ)=∑P(O|I,λ)\*P(I|λ)）。

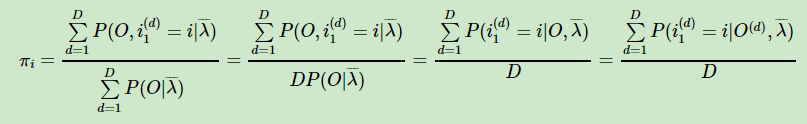
1.定义时刻t时隐藏状态为qi, 观测状态的序列为o1,o2,...ot的概率为前向概率，不难发现有前向递推式：，即下一隐状态可以由任意隐状态转化得到。最终计算T时刻是α的和即可

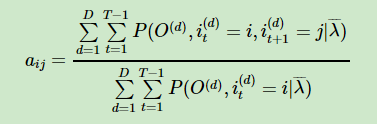
2. 定义时刻t时隐藏状态为qi, 从时刻t+1到最后时刻T的观测状态的序列为ot+1,ot+2,...oT的概率为后向概率，不难发现后向递推式：，同理可以反推到1时刻的后向概率。

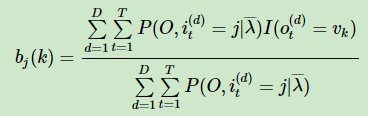
（2）模型参数学习问题。即给定观测序列O={o1,o2,...oT}，估计模型λ=(A,B,Π)的参数，使该模型下观测序列的条件概率P(O|λ)最大

一般训练的样本是不包含隐状态的，若包含可以直接使用最大似然法进行求解，否则需要使用EM算法进行迭代优化：在E步中求出联合分布P(O,I|λ)基于条件概率P(I|O,λcur)的期望：；在M步最大化上述期望，得到更新后的参数λ：。

利用拉格朗日乘子法，求导为0可以得到对应的A、B以及π的值。







（3）预测问题，也称为解码问题。即给定模型λ=(A,B,Π)和观测序列O={o1,o2,...oT}，求给定观测序列条件下，最可能出现的对应的状态序列

使用维特比算法进行解码，该法是基于动态规划求序列最短路径的方法。

HMM定义了两个局部状态：

1.时刻t隐状态为i的所有路径的概率最大值，且满足t+1时递推式

2. 时刻t隐藏状态为i的所有单个状态转移路径中概率最大转移路径中t-1时刻的隐藏状态



|  |
| --- |
| 特比算法流程 |
| 输入：HMM模型λ=(A,B,Π)，观测序列O=(o1,o2,...oT)  输出：最有可能的隐藏状态序列I∗={i∗1,i∗2,...i∗T}  1）初始化局部状态：    2) 进行动态规划递推时刻t=2,3,...T时刻的局部状态：    3) 计算时刻T最大的δT(i),即为最可能隐藏状态序列出现的概率。计算时刻T最大的Ψt(i),即为时刻T最可能的隐藏状态。    4) 利用局部状态Ψ(i)开始回溯。对于t=T−1,T−2,...,1：    最终得到最有可能的隐藏状态序列I∗={i∗1,i∗2,...i∗T} |