声纹识别初探-以 GMM-UBM 为例

2018 年 8 月于信研院声纹识别实验室

1 概述

声纹识别，也称作说话人识别，顾名思义,是一种通过声音判别说话人身份的技术。

声纹识别能作为不同个体判别依据的基础是:每一个声音都具有独特的特征，通过该特征能将不同人的声音进行有效的区分。

这种独特的特征主要由两个因素决定，第一个是**声腔的尺寸**，具体包括咽喉、鼻腔和口腔等，这些器官的形状、尺寸和位置决定了声带张力的大小和声音频率的范围。因此不同的人虽然说同样的话，但是声音的频率分布是不同的，听起来有的低沉有的洪亮。每个人的发声腔都是不同的，就像指纹一样，每个人的声音也就有独特的特征。

第二个决定声音特征的因素是**发声器官被操纵的方式**，发声器官包括唇、齿、舌、软腭及腭肌肉等，他们之间相互作用就会产生清晰的语音。而他们之间的协作方式是人通过后天与周围人的交流中随机学习到的。人在学习说话的过程中，通过模拟周围不同人的说话方式，就会逐渐形成自己的声纹特征。

因此，理论上来说，声纹就像指纹一样，很少会有两个人具有相同的声纹特征。这也使得声纹识别有了发展和落地的空间.

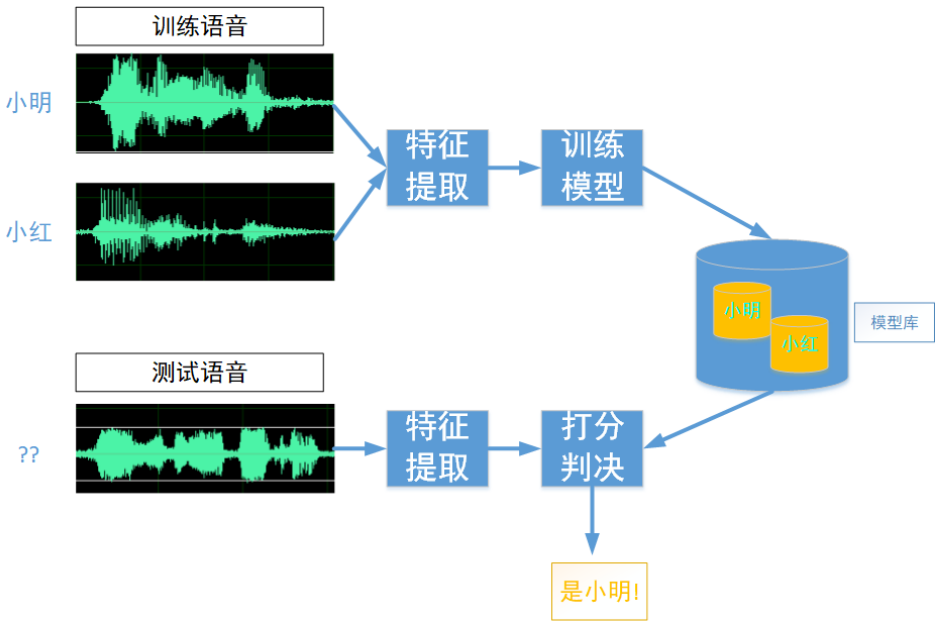
在声纹识别领域,最经典的模型莫过于 GMM-UBM 模型,他在该领域中的卓越表现使得此模型自产生以来就成为许多论文中使用的基线系统,因此,本文将以该系统为例,希望借此帮助未来在声纹实验室实习的学弟学妹见微知著,最终打开声纹识别的大门.

本文将首先介绍声纹识别的基本流程并着重介绍每一步的输入输出形式,以帮助读者对声纹处理过程有一个感性的认识,然后以 GMM 为基础,重点介绍 GMM-UBM 模型. 需要说明的是:本文旨在帮助未接触过声纹处理的同学厘清流程脉络,理解其中的原理,点开相关的技能树,因此涉及到具体数学上的推导略微简略,读者应该以本文为知识地图并按图索骥,在阅读本文的同时辅以相关扩展文献(每一章节后会给出延伸阅读的推荐链接),以期掌握其中的数学原理. 我们有理由相信,对经典模型的深入学习,对于日后的科研工作一定能带来巨大的帮助.

2 简单说话人识别流程

(注:对于本节中出现的新名词,读者只需知其然,后文中会有更详细的介绍)

说话人识别过程分为**训练**和**识别**两个模块。训练模块的内容是:从说话人提供的若干语音中提取能反映个性的特征,并为其建立说话人模型,等待识别模块调用;识别模块的内容是:提取待测语音特征并判断待测语音的身份。说话人识别的系统框图如下。



从图中看出,说话人识别过程主要有三个模块,分别为:特征提取,模型训练以及模式匹配与判决.

2.1 特征提取

声音的时域波形只代表声压随时间变化的关系，而不能代表声音的特征。因此，必须把声音波形转换为声学特征向量.目前有许多声音特征提取方法，如梅尔频率倒谱系数 MFCC、线性预测倒谱系数 LPCC、多媒体内容描述接口 MPEG7 等，其中 MFCC 是基于倒谱的，更符合人的听觉原理，因而是最普遍、最有效的声音特征提取算法。

在提取 MFCC 前，由于人的器官，采样时的噪音对声音采集会产生影响并且声音不具有周期性等原因，需要对声音做前期处理，包括模数转换(即把模拟信号转换成数字信号,进行采样和量化)、预加重、分帧、加窗。

2.2 训练说话人模型

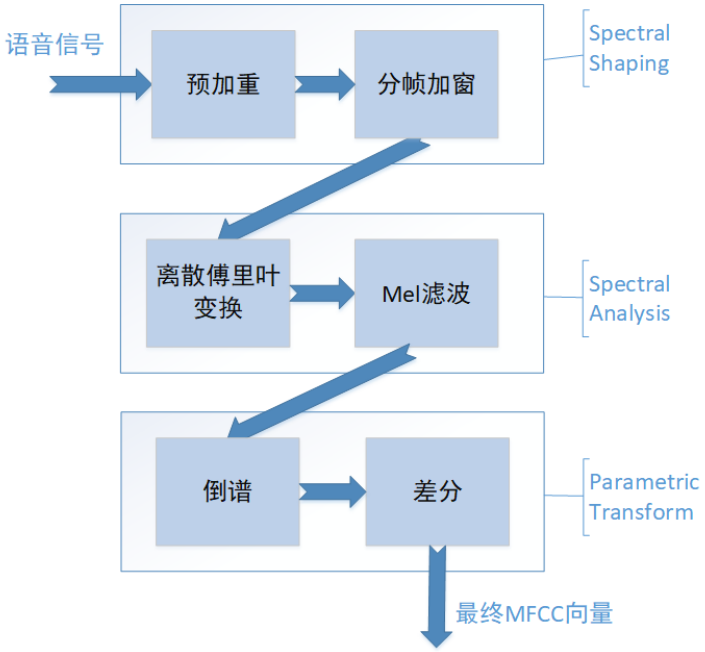
主要包含两部分。由已知的参考说话人的所有训练数据训练出一个通用背景模型 UBM,之后每个说话人根据自己的特征参数在 UBM 的每个单高斯上进行自适应得到说话人模型并等待匹配。

2.3 模式匹配与判决决策

说话人识别系统中,不同的模式匹配最大区别在于说话人模型的表示和测试时语音匹配的方法,常用识别方法中的概率模型更具有灵活性,其似然得分的理论意义更有说服力。其中 GMM 用多个高斯分布的线性组合来近似表征多维矢量的任意连续概率分布,能够有效的描述说话人的特征,其在文本无关的说话人识别领域有很高的识别率。

3 特征提取

特征提取阶段的基本流程如下图所示:



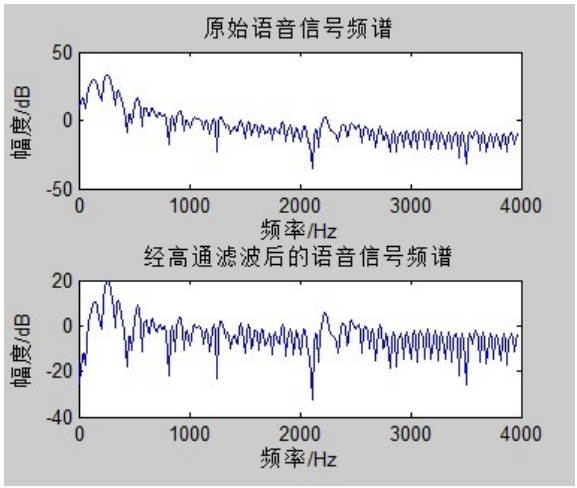
其中最重要的一步就是傅里叶变换. 傅里叶变换作为分析**平稳信号 稳态特性**的强有力的工具,可以**从语音信号中抽取频谱信息(**下文中将会介绍**)**,继而在后续步骤中利用这些信息构造特征向量,是整个特征处理过程中承前启后的关键. 但是,在进行 Spectral Analysis 之前,为了使得语音信号能被傅里叶变换处理,必要的 Spectral shaping 工作必须进行,下面将分别介绍图中每一步的输入与输出.

3.1 预加重

由于语音信号的平均功率谱受**声门激励**和**口鼻辐射**影响，高频端大约在 800Hz 以上按 6dB/倍频程跌落，即 6dB/oct（2 倍频）或 20dB/dec（10 倍频），所以求语音信号频谱时，**频率越高相应的成分越小**，高频部分的频谱比低频部分的难求。为此要在预处理中(对高频信号)进行预加重（Pre-emphasis）处理.

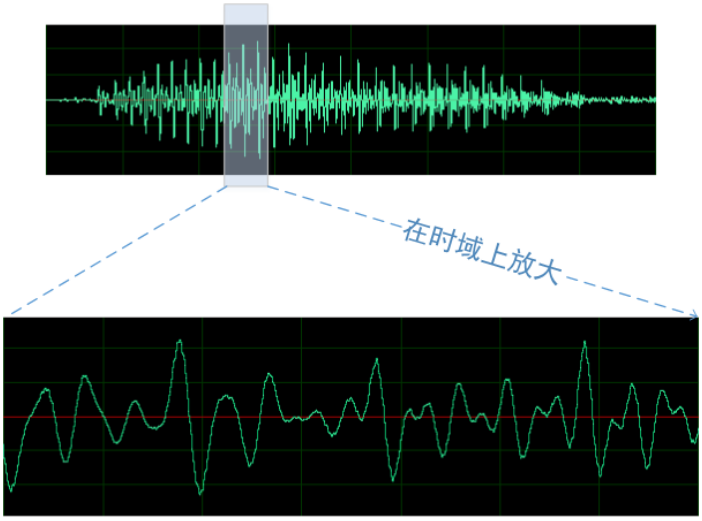
预加重的目的是**加重高频部分，使高频部分的能量和低频部分能量有相似的幅度**，保证在低频到高频的整个频带中，能用同样的信噪比求频谱，同时能更好地**利用高频共振峰**以便于频谱分析或者声道参数分析。

由下图可以看出，预加重后的频谱在高频部分的幅度得到了提升,从而增加了语音的高频分辨率。



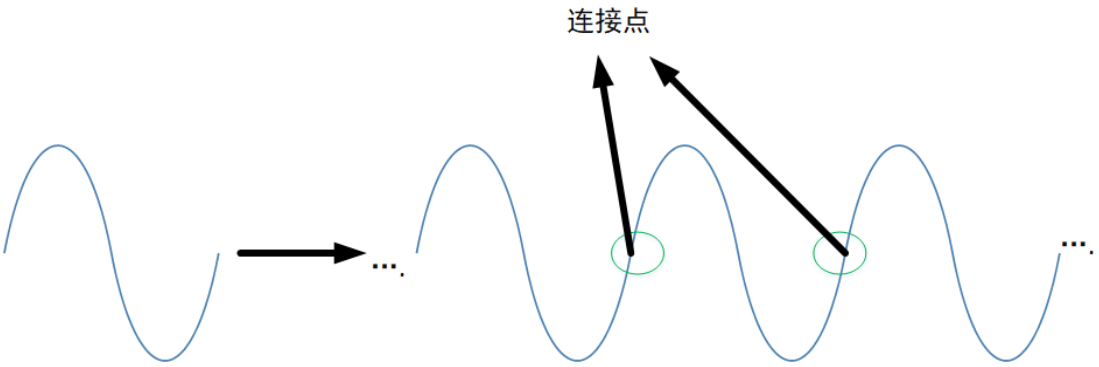
3.2 分帧

对于音频信号，按局部 or 全局的观念、持续时间长短,其特征可以分为长期（long-term）、中期（mid-term）、短期（short-term）. 其中短时特征具备了一个良好的性质:在一个 20-50 毫秒的范围内，语音近似可以看作是良好的**平稳信号**.这种平稳性质对傅里叶变换是必不可少的,因此,我们在进行信号分析之前首先要对信号进行分帧.

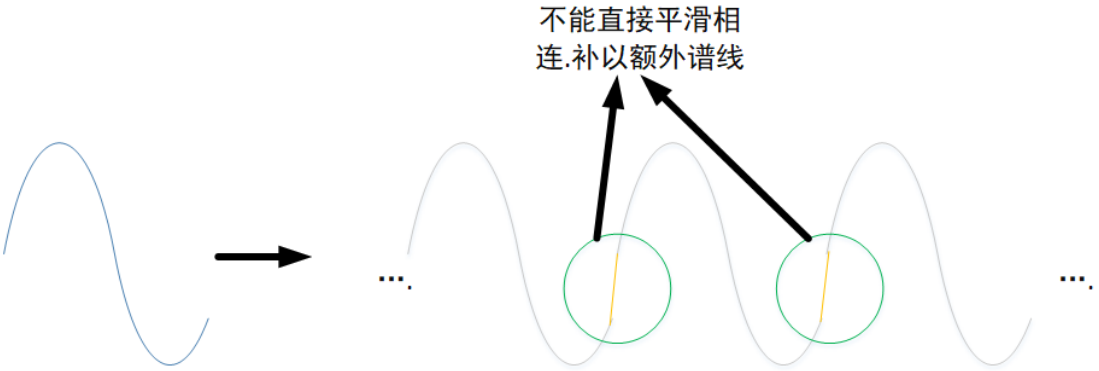


3.3 加窗

短时傅里叶变换(FFT)的基础是假设一帧信号都是无限长的周期信号,也就是一帧数据会被认为是无限重复的，最后一个点之后又连到第一个点，不断拼接下去，如下图所示:

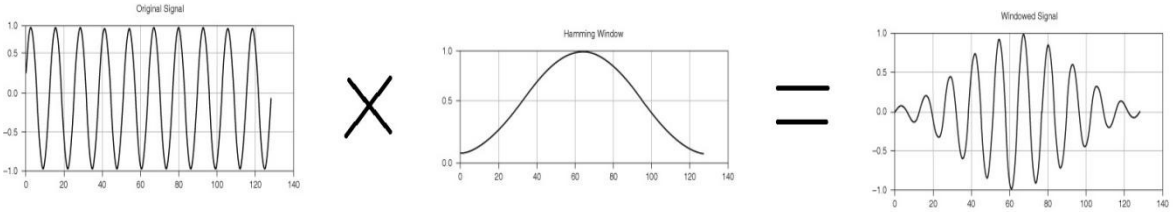


但是,如果分出的一帧信号在端点处不能被平滑地连接,就会出现波形不连续的情况(如下图所示),继而导致 FFT 结果出现**频谱泄漏**现象(*建议查阅延伸阅读链接,直观上将泄露出的频谱理解为黄色谱线对应的频谱*).

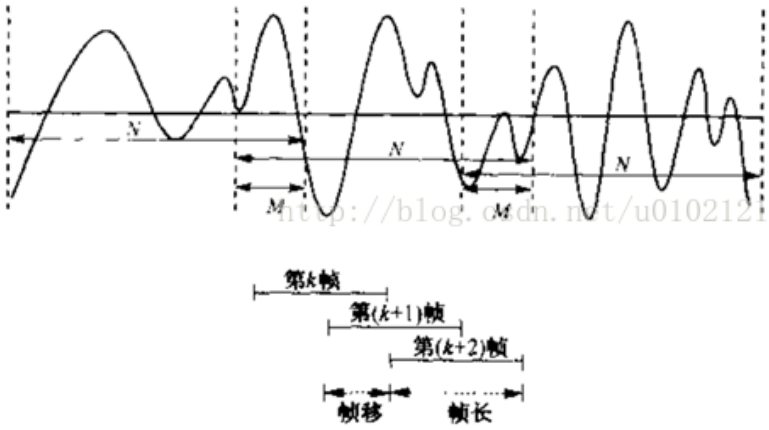


加窗就是为了解决这个问题.通过加窗操作，一帧信号的幅度在两端会渐变到 0，因此,无论分帧的结果是何种形式，该帧信号**左右端点一定能平滑相连**，每一帧也就会表现出周期函数的特征，避免出现**吉布斯效应**(*建议查阅延伸阅读链接做进一步理解*)。同时,渐变对傅里叶变换也有额外的好处，它可以提高变换结果（即频谱）的分辨率.

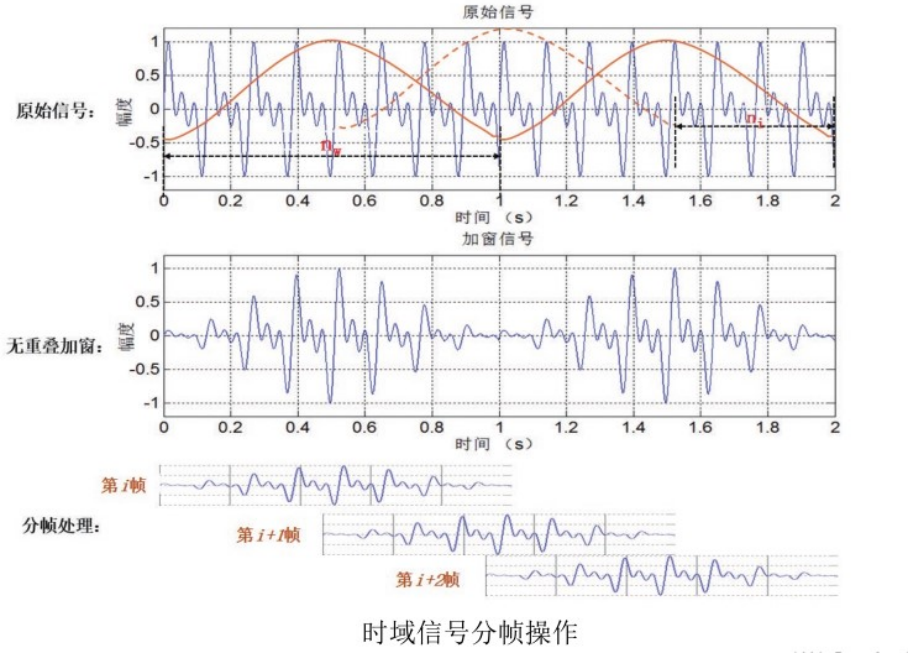
加窗的操作，实际上是用**帧**与一个**窗函数**相乘，如下图所示：



加窗的代价是**一帧信号两端的部分被削弱**了，没有像中央的部分那样得到重视。弥补的办法是，帧不要背靠背地截取，而是**相互重叠**一部分。相邻两帧的起始位置的时间差叫做帧移，常见的取法是取为帧长的一半，或者固定取为 10 毫秒。



分帧和加窗的整个过程可以用下图表示:



时域信号分帧操作

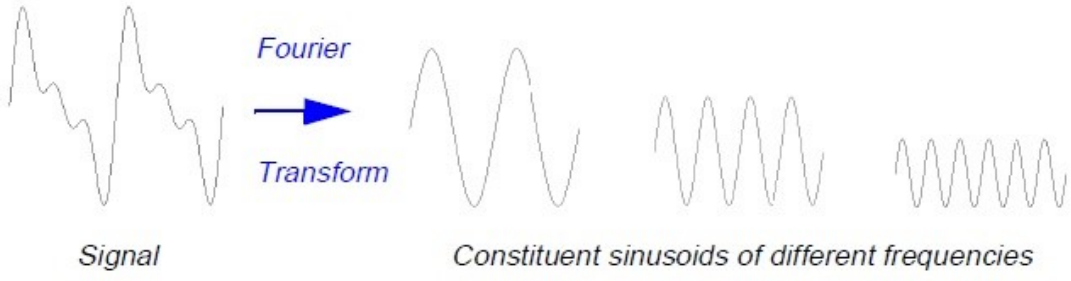
延伸阅读

频谱泄露(spectrum leakage): blog2　blog1
吉布斯现象(Gibbs phenomenon): Wikipedia　zhihu

3.4 提取 MFCC

3.4.1 短时傅里叶变换 FFT

不同频率的声音混合在一起，才组成了人类的语音。为了使语音信号更容易被模型处理，我们可以把这个复杂的声波分解成一个个组件部分。傅里叶变换 Fourier Transform 可以帮助我们做到这一点，他将**时间域的信号**分解为**不同频率的正旋信号**,如下图所示:



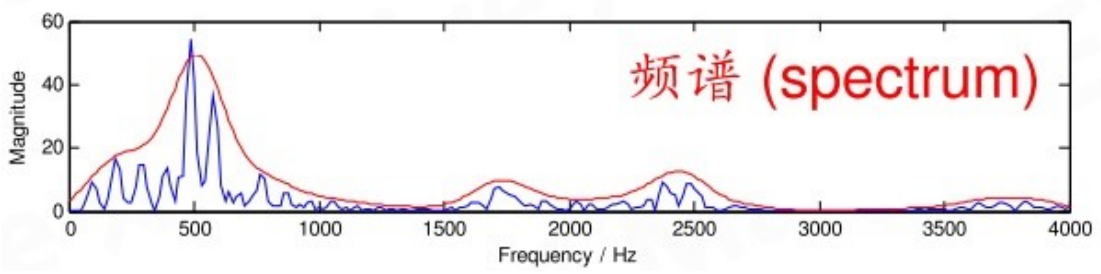
有了这样的分解,我们就可以构建该信号在**不同频率下的能量关系**,也就是相当于将时间域的信号分解为频率域的信号,如下图所示(注意横坐标变化,相当于记录分解出的每一个子频率波形的幅值)

延伸阅读

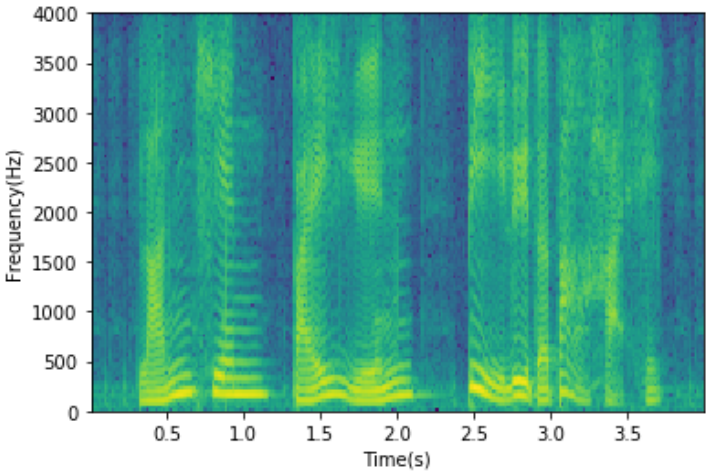
时域与频域: doc 时频动图
傅里叶变换: zhihu

3.4.2 频谱图

对一帧信号做傅里叶变换，得到的频谱图如下图中的**蓝线**所示(也就是 3.4.1 节图 2 右的放大版本)：



图中的横轴是频率，纵轴是幅度。频谱上就能看出这帧语音在 480 和 580 赫兹附近的能量比较强。如果我们把连续的多个帧的频谱图拼接在一起就可以得到下图中正对一条语音信号的完整语谱图:

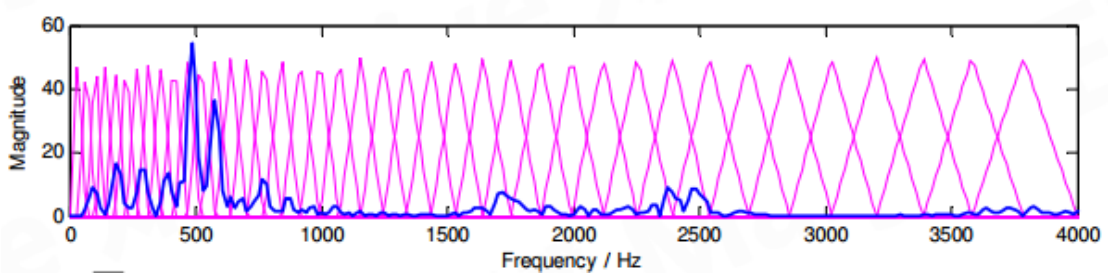


注意:这张图实际上是一个**三维图**,横坐标是时间(即代表一连串帧),纵坐标是频率(也就是这一时刻的帧经过 FFT 后分出的频率),第三维坐标是能量,图中用颜色深度来表示其值. 该图中的某一点含义是:在这一帧分解出的这个频率的波的能量密度.

3.4.3 倒谱分析

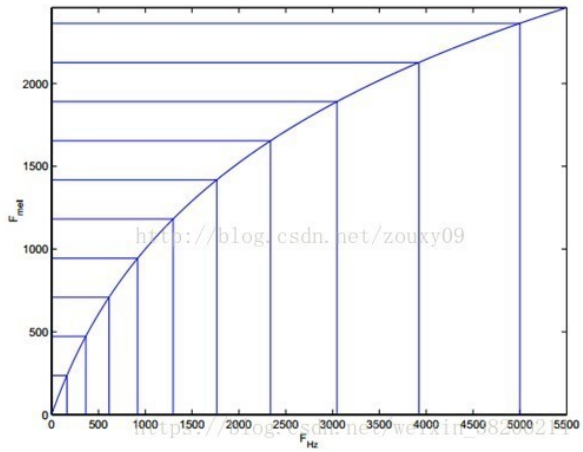
如上一节频谱图所示,语音的频谱，常常呈现出**精细结构**和**包络**两种模式。精细结构就是蓝线上的一个个小峰，它们在横轴上的间距就是基频，它体现了语音的音高——峰越稀疏，基频越高，音高也越高。包络则是连接这些小峰峰顶的平滑曲线（红线），它代表了口型(回忆第一节的内容,可以发现口型实际上决定了语音的特征)。包络上的峰叫共振峰，图中能看出四个，分别在 500、1700、2450、3800 赫兹附近。**倒谱分析的作用就是从一段语音，提取出它的频谱包络和频谱细节**.

考虑到人类的听觉特征: 人耳类似于一个滤波组，只关注特定的频率分量。并且这些滤波器在频率坐标轴上，并不是统一分布的，在低频区域有很多的滤波器，分布比较密集，在高频区域，滤波器的数目会比较少，分布很稀疏。下图用三角滤波模拟了人耳滤波过程,其中粉线为滤波器组,蓝线为一帧语音的频谱图. 从图中可以清晰地看到低频部分三角滤波器明显比高频部分密集:

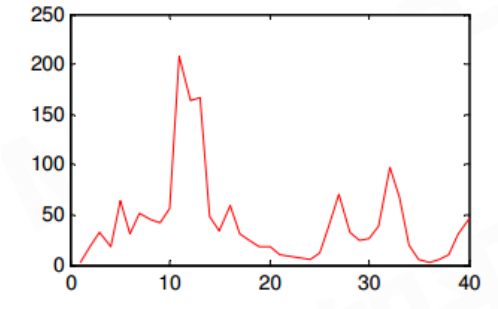


为了使**频谱图更紧凑**,在梅尔频谱分析(MFCC)中,需要先将傅里叶变换得到的线性频谱映射到基于听觉感知的 Mel 非线性频谱中[1]，然后再进行倒谱分析. 其中,原线性频谱的频率和 Mel 非线性频谱的频率之间的关系可由下图表示,横轴表示原频率,纵轴表示映射后的 Mel 频率:

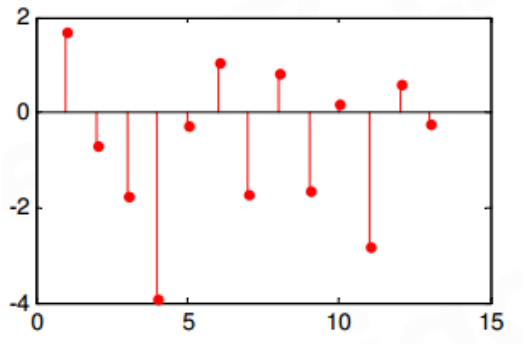
[1] 通过 Mel 滤波实现这种映射,Mel 滤波常用滤波器组是三角滤波器组,它可以消除谐波的作用，突显原先语音的共振峰



下图为滤波后得到的 Mel 非线性频谱图,其中 0-10 和 30-40 之间的 mel 频率差相同,而后者真实频率差更大:



在 Mel 非线性频谱图的基础上,进行的倒谱分析主要包括两步:**取对数**(Log)和**离散余弦变换**(DCT),将连续的 Mel 频谱中的包络和细节函数分别提取出来，如下图 。然后从包络函数中选取第 2 个到第 13 个系数组成一帧信号的 MFCC 向量,具体公式可见第三章.离散化结果如下图所示:



延伸阅读

梅尔频率倒谱系数: cmu 大学教程 MFCC 实现细节 blog

3.4.4 差分

由于语音信号是时域连续的，上述过程提取的特征信息只反应了本帧语音的特性，这种特征

是**静态**的,它没有考虑到*帧与帧之间的关系*所包含的时域连续性. 为了使特征更能体现这种**动态性**，可以在特征维度增加前后帧信息的维度。常用的是一阶差分和二阶差分。

一阶差分就是离散函数中连续相邻两项之差. 定义 X(k)为第 k 个语音帧得到的 MFCC 向量,则 Y(k)=X(k+1)-X(k)就是此函数的一阶差分，物理意义就是**当前语音帧与前一帧之间的动态关系**.

在一阶差分的基础上，Z(k)=Y(k+1)-Y(k)=X(k+2)-2*X(k+1)+X(k)为此函数的二阶差分.二阶差分表示的是一阶差分与一阶差分之间的关系。即前一阶差分与后一阶差分之间的关系，体现到帧上就是**相邻三帧之间的动态关系**。

4 基于 GMM-UBM 的说话人识别基准模型

声纹识别领域的许多研究都会以 GMM-UBM 为基准模型,在介绍 GMM-UBM 之前,我们会补充一些基础知识.包括混合高斯模型 GMM 和通用背景模型 UBM 等.

4.1 混合高斯模型 GMM(Gaussian Mixture Model)

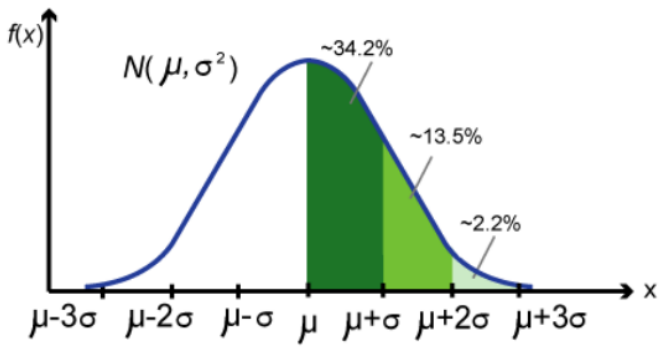
混合高斯模型,从名称中就能看出两个显著的特点: **混合** 与 **高斯**,理解了这两个特点的含义,也就理解了混合高斯模型.

4.1.1 高斯模型 GM

高斯分布又称正态分布,相信每一个学过数理统计的同学对正态分布都非常熟悉.这个钟型的分布曲线不但形状优雅,其密度函数写成数学表达式也非常具有数学上的美感.

当**样本数据 X 是一维数据**（Univariate）时，高斯分布遵从下方概率密度函数（Probability Density Function）：

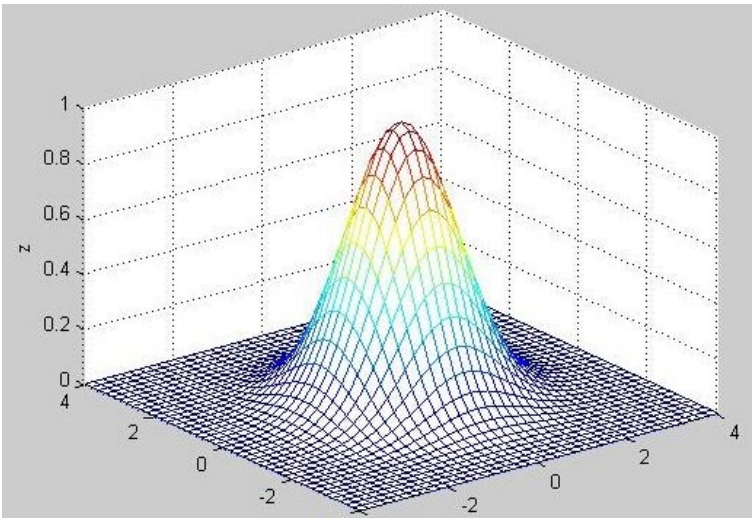
$$p\left(x \mid \theta\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)} = \mathrm{N}\left(x \mid \theta\right)$$



其中 u 为数据均值（期望）， σ 为数据标准差（Standard deviation）。

当**样本数据 X 是多维数据**（Multivariate）时，高斯分布遵从下方概率密度函数：

$$p(x \mid \theta) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right) = \mathrm{N}(x \mid \theta)$$



其中，u 为数据均值（期望），Σ 为协方差（Covariance），D 为数据维度。

4.1.2 混合模型 MM

为了表征所获得的数据,科学家们在最初发明了各种各样的分布来描述他们,然而,仅仅靠单一的某种分布,在实际运用中往往会出现许多问题:

1 **“学的太差”**:单一模型可能**无法很好地总结出数据的特征**
试想一下:你有一个用于判断是否是鱼的模型,当你告知模型”水里游的那个家伙是鱼”, 模型归纳出“在水里游的都是鱼”这样一个结论,那么这个模型可能会把虾、龟、甚至是潜水员都当做鱼,这显然是不合理的.
2 **“学的太好”**:单一模型可能会使得**泛化能力太差**
与上一例相反,如果模型学习到那条鱼的全部特征(包括鱼的共有特征和独有特征)并认为只有包含全部特征的生物才是鱼,那么下次看到另一条鱼时，他并不知道那也是鱼，因为两条鱼总有一些地方不一样的，或者就算是同一条鱼，在河里不同的地方看到，或者只是看到的时间不一样，也会被他认为是不同的.这就使得模型存在一种”**过拟合**”问题.

为了在”学的太差”和”学的太好”之间做一个权衡,科学家们又创造出了一种混合模型.

混合模型是一个可以用来表示在总体分布（distribution）中含有 K 个子分布的概率模型，换句话说，混合模型表示了观测数据在总体中的概率分布，它是一个**由 K 个子分布组成的混合分布**。混合模型不要求观测数据提供关于子分布的信息，来计算观测数据在总体分布中的概率。

不难发现,混合模型本身可以变得任意复杂,因为**通过增加 Model 的个数，我们可以任意地逼近任何连续的概率密分布**,这样就避免了”学的太差”. 同时,对于数据的全部特征,混合模型

可以**用其子模型去拟合某个子特征**,相较于用一个模型去学习全部特征,无疑提高了模型对数据的公共特征的描述能力,从而解决了”学的太好”的问题.

延伸阅读

过拟合: zhihu blog

4.1.3 高斯混合模型 GMM

4.1.3.1 为什么要用 GMM?

在介绍高斯混合模型之前,先请读者思考两个问题:
1 为什么要使用**概率分布**?直接对数据进行聚类(如 k-means)不是更直观的一种方法吗?
2 为什么一定要使用**高斯**分布来混合?使用其他分布的混合模型能否达到相同的效果?

对于问题 1:
聚类的做法确实能够直接得到一个分类结果,但是如果能够得到这个分类结果的概率,我们得到的信息量将会多很多.

比如:我可以把这个概率进一步转换为一个 score ，表示算法对自己得出的这个结果的把握,当我对同一个任务使用不同的方法得到不同的结果后,聚类的做法无法帮助你从众多结果中做出正确的取舍,而有了结果的”分数”,我们就可以**将这些结果做个比对**并选择”把握”最大的那个结果.

另一个很常见的例子是在诸如疾病诊断之类的场所，机器对于那些很容易分辨的情况（患病或者不患病的概率很高）可以自动区分，而对于那种很难分辨的情况，比如，49% 的概率患病，51% 的概率正常，如果仅仅简单地使用 50% 的阈值将患者诊断为“正常”的话，风险是非常大的，因此，在机器对自己的结果把握很小的情况下，会“拒绝发表评论”，把这个任务留给有经验的医生去解决,而这种”把握”,只有通过概率分布才能获得。

对于问题 2:
二项分布混合模型,泊松分布混合模型等等 Mixture Model 显然是可行的,但是高斯分布有着令人难以拒绝的性质.

首先,中心极限定理指出:当数据的采样足够多时,n 个采样的平均数 $\overline{x_n}$ 的分布会接近于一个高斯分布,该高斯分布的均值等于单个分布的均值 u, 方差等于单个分布的方差除以 n,即 $N\left(\mu, \frac{\sigma^2}{n}\right)$. 这意味着,几乎所有采样结果,只要 n 足够大,都可以用**平均数的高斯分布去近似变量的分布**!这个性质可以说相当霸道了…

其次,由于高斯分布的数学表达式中含有自然数 e,我们很自然地想到对其进行 log 化,从而避

免过多的乘法运算,这种良好的计算性质使得高斯混合模型成为主流.

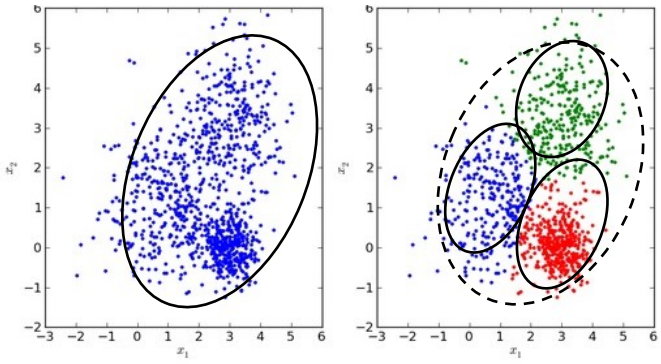
4.1.3.2 GMM 定义

每个高斯混合模型可以看作是由 M 个单高斯模型组合而成的模型，这 M 个子模型是混合模型的隐变量（Hidden variable）,被称为一个" Component", 这些 Component 线性加成在一起就组成了 GMM 的概率密度函数：

$$P\left(x \mid \theta\right) = \sum_{m=1}^{M} c_m \mathrm{N}\left(x \mid \theta_m\right)$$

其中c_m 表示权重系数,全部样本$x = \left\{x_1, \ldots, x_N\right\}$, x_i 表示第 i 个样例,θ_m表示第 m 个子模型的参数集$\left(\mu_m, \Sigma_m\right)$.假如我们已知$c_m$和$\theta_m$，那么整个 GMM 刻画的范围已知，因此$P\left(x \mid \theta\right)$ 则表示当前所有样本点落入 GMM 范围的概率。

一个比较直观的例子是:
我们现在有一组狗的样本数据，不同种类的狗，体型、颜色、长相各不相同，但都属于狗这个种类，此时单高斯模型(下图左)可能不能很好的来描述这个分布，因为样本数据分布并不是一个单一的椭圆(代表一个二维高斯分布)，所以用混合高斯分布可以更好的描述这个问题，如下图右所示,该混合高斯模型(虚线)有三个子分布(实线),可以分别用来表征狗的体型,狗的颜色,狗的长相：



延伸阅读:

中心极限定理: zhihu

4.1.4 模型训练

4.1.4.1 最大似然估计(MLE): 优化目标

如果我们知道模型的参数，那我们一定可以准确的求解答案。但我们现在需要解决的问题正相反，我们知道答案，想要去求解模型的参数，因此我们引入最大似然估计。

我们先解释下**似然函数**的意义：
对于函数 P(x| θ)，x 表示某一个具体的数据； θ 表示模型的参数
如果 θ 是已知确定的，x 是变量，这个函数叫做概率函数(probability function)，它描述对于不同的样本点 x，其出现概率是多少。
如果 x 是已知确定的， θ 是变量，这个函数叫做似然函数(likelihood function)，它描述对于不同的模型参数，出现 x 这个样本点的概率是多少。

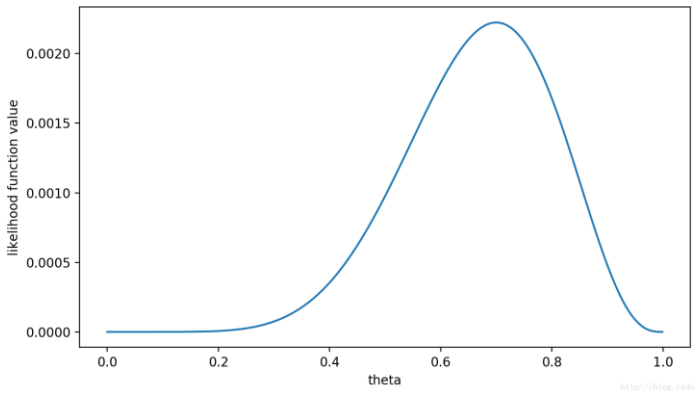
最大似然估计的思想就是我不能准确的算出参数大概是多少，因此我把能最大产生已知数据点的参数当成答案。

举例来说：假设有一个造币厂生产某种硬币，现在我们拿到了一枚这种硬币，想试试这硬币是不是均匀的。即想知道抛这枚硬币，正反面出现的概率（记为 θ ）各是多少？于是我们拿这枚硬币抛了 10 次，得到的数据（x0）是：反正正正正反正正正反。我们想求的正面概率 θ 是模型参数，而抛硬币模型我们可以假设是二项分布。

那么，出现实验结果 x0（即反正正正正反正正正反）的似然函数是多少呢？

$$f(x_0,\theta) = (1-\theta)\times\theta\times\theta\times\theta\times(1-\theta)\times\theta\times\theta\times(1-\theta) = \theta^7(1-\theta)^3 = f(\theta)$$

注意，这是个只关于 θ 的函数。而最大似然估计，顾名思义，就是要最大化这个函数。我们可以画出 f(θ)的图像(对于不能明确画图的函数，一般取导数后置 0 求解)：



可以看出，在 θ =0.7 时，似然函数取得最大值。这样，我们已经完成了对 θ 的最大似然估计。即，抛 10 次硬币，发现 7 次硬币正面向上，最大似然估计认为正面向上的概率是 0.7。(ummm.. 这非常直观合理，对吧？）虽然结果与我们的经验并不符合（原因在于实验次数也就是数据量太小），但 0.7 确实能完美的产生我们的已知数据。

有了以上的认识,我们不难得出 GMM 的似然函数:

$$f\left(x,\theta\right) = \prod_{i=1}^{N} p\left(x_i \mid \theta\right)$$

他表示参数 θ 产生数据 x 的可能性大小.

由于单个点的概率都很小，许多很小的数字相乘起来在计算机里很容易造成浮点数下溢，因

此我们通常会对其取对数，把乘积变成加和,于是 GMM 的似然函数又可以写成:

$$\log \prod_{i=1}^{N} p\left(x_i \mid \theta\right)=\sum_{i=1}^{N} \log \left(p\left(x_i \mid \theta\right)\right)=\sum_{i=1}^{N} \log \left\{\sum_{m=1}^{M} c_m N\left(x_i \mid \theta_m\right)\right\}$$

其中 N 为样例总数, 子分布的参数 $\theta_m = \left(\mu_m, \Sigma_m\right)$.

接下来我们只需要**将这个函数作为优化目标,找到能使这个似然函数取得最大值的那组参数**,就得到了我们要求的最佳参数.

延伸阅读

似然函数与概率函数: zhihu
Em 算法中的最大似然估计(文章第一部分和第二部分): blog

4.1.4.2 期望最大化算法(EM): 优化方法

回想一下,高等数学中,给定一个函数,让你去求这个函数的在取最大值时对应的自变量值,你首先想到的是什么?没错,当然是**求导并令导数等于零**，然后解方程,那么用这个方法去求解 GMM 的似然函数是否可行呢?

显然,如果直接对整个函数求导,似然函数中第二个连加符号会使得求导结果的展开异常复杂. 同时,无论是对 μ_m 还是对 Σ_m 求偏导,结果中都会带有 c_m 项,仅用一个方程是无法解出两个未知量的. 为了解决这个问题,就有了 EM 算法.

从实现算法的角度直观理解 GMM 中的 EM 算法

由于 EM 算法背后的数学原理略显复杂且需要引入**隐型变量**,这将不可避免地使文章变得晦涩难懂且冗长无味,因此,在用数学公式定量描述 EM 算法之前,我们希望从头到尾走一遍算法的流程,从实践的角度总结出 EM 的核心思想. 同时,结合下一节中的数学推导,我们希望读者对 EM 中的概念能有更深刻的理解并能够以此节为蓝本自己动手实现 EM 算法.

首先,笔者想提出一个很有意思的看法: *对于一个样例,我们既可以看作他是完全由某个子分布取样产生的 ,但是反过来,我们也能认为这个样例是由整个混合模型产生的,其中每个子分布只贡献了样例的一小部分.*
在接下来的过程展示中,你将看到如何活用这些想法去实现一个 EM 算法.

第一步:

既然当前我们没有任何辅助决断信息,不妨就先瞎猜一个参数 $\theta=\left\{c_m, \mu_m, \Sigma_m\right\}_{m=1}^{M}$ 作为初始参数,看看用这个初始参数能推导出什么样的结果.

现在,我们已经得到了 GMM 模型中每个子分布的参数,那么我们不仅可以算出**混合模型产生这个样例的概率** $\sum_{m=1}^{M} c_m \mathrm{N}\left(x_i \mid \theta_m\right)$,同样也可以算出**独立情况**下混合模型中某个**子分布产生这个样例的概率** $N\left(x_i \mid \theta_m\right)$,为了将这个概率归一化(每个子分布生成这个样例的概率和为 1),我们重新定义一个值:

$$\gamma\left(i, m\right) = \frac{c_m N\left(x_i \mid \theta_m\right)}{p\left(x_i\right)} = \frac{c_m N\left(x_i \mid \theta_m\right)}{\sum_{j=1}^{M} c_j N\left(x_i \mid \theta_j\right)}$$

其中 c_m 是混合模型中第 m 个高斯分布的权重系数, $N\left(x_i \mid \theta_m\right)$表示独立情况下这个子分布取样这个数据的概率. 而 $\gamma\left(i, m\right)$ 这个值既可以表示数据 x_i 完全由第 m 个子高斯分布产生的概率(对应上文第一种看法),也可以用来表示数据 x_i 中由第 m 个子高斯分布负责产生的比例(对应上文第二种看法)

第二步:
别忘了,我们的终极目标是要为了更新参数!具体到实现环节,我们实际上要更新的是每个子分布的参数,如若将数据视为由混合模型产生(对应上文第二种看法),为了使用**极大似然法**更新**子分布**的参数,我们必须再为这些子分布**"量身定做"一批新数据**,这时候,第一步中的值就派上了用场.

对于每一个数据 x_i,我们可以看作 $\gamma\left(i, m\right) * x_i$ 这部分是由第 m 个高斯分布产生的, 扩展到所有数据,实际上相当于第 m 个高斯分布产生了 $\gamma\left(1, m\right) x_1$, ... ,$\gamma\left(N, m\right) x_N$ 这 N 个点.
由于这些子分布都是单一的高斯分布,他们的似然函数是:

$$\sum_{i=1}^{N} \log\left(c_m N\left(x_i \mid \theta_m\right)\right)$$

相比混合高斯分布的似然函数,求导是不是变得很容易了呢?不难算出,这些新的数据点对应的新参数 $\theta_m = \left(\mu_m, \Sigma_m\right)$是:

$$\mu_m = \frac{1}{N_m} \sum_{i=1}^{N} \gamma\left(i, m\right) x_i$$

$$\Sigma_m = \frac{1}{N_k} \sum_{i=1}^{N} \gamma\left(i, m\right)\left(x_i - \mu_m\right)\left(x_i - \mu_m\right)^T$$

$$c_m = \frac{N_m}{N}$$

其中 $N_m = \sum_{i=1}^{N} \gamma(i, m) * 1$ 代表针对每一个高斯分布新生成的数据的"实际个数".

第三步:
现在我们得到了新的参数 θ ,怎么样,有没有一种似曾相识的感觉?没错,我们要用这个新参数代替第一步的初始参数,并且继续用原始数据重复之前的工作,一直到某一轮迭代中新参数的变化很小,也就是似然函数的值收敛的时候,这样,我们就完成了对模型参数的确定.

有了以上直观的认识,我们可以对 EM 算法做个通俗的定义：
在一般性的问题中，EM 算法包含两步:步骤 1(预计步骤)其实就是根据现有的模型（参数），计算各个观测数据输入到模型中的计算结果,步骤 2(最大化步骤)根据计算结果,用最大似然的方法重新计算模型参数.

从数学原理理解 GMM 中的 EM 算法

看到这里,你或许会有一些疑惑: 既然 EM(expectation maximization)算法被称为期望最大化算法,那么"期望"和"最大化"分别代表什么意思呢?别着急,让我们接着从数学角度做一些探讨.

其实对于"最大化",前文中已经隐晦地做出了一些解释,你可以将它理解为用最大似然方法求解模型参数(也就是将似然函数最大化),至于"期望",则稍微复杂一些,为了避免问题变得太抽象,我们接着从 GMM 的似然函数说起. 回顾一下我们之前要解决的问题:求以下 Log-likelihood function 的最大值:

$$\sum_{i=1}^{N} \log \left\{ \sum_{m=1}^{M} c_m N\left(x_i \mid \theta_m\right) \right\}$$

但是由于在 log 函数里面又有加和，没法直接求。考虑一下前文中对样例的**第一种看法**,我们可以认为 GMM 生成样本数据的过程包含如下两步：**先随机选择一个子分布,每个分布被选中的概率实际上就是他的系数 c_m，然后再从这个子分布所对应的那个普通的高斯分布里进行取样**。这里实际上相当于把 GMM 当作一个**聚类模型**,其中**每一个子分布就是一个类别**,为此我们可以很自然地引入一个隐含变量 z_i 来表示样本数据 x_i 由哪个子分布产生(或者说样本数据 x_i 属于那个类别. 显而易见的是对于每一个数据 x_i 都会有一个与之相关的隐含变量 z_i)，这是一个 M 维向量，如果第 m 个子分布被选中了，我们就将 z_i 第 m 个元素置为 1 ，其余的全为 0 。

那么，再来看看，如果除了之前的抽样的值 x_i 之外，我们同时还知道了每个 x_i 所对应的隐

含变量 z_i 的值，情况又会变成怎么样呢？

因为我们同时观察到了 x_i 和 z_i,所以我们现在的最大化的似然函数就从 $\prod_{i=1}^{N} p\left(x_i\right)$ 变成了 $\prod_{i=1}^{N} p\left(x_i, z_i\right)$,注意到 $p\left(x_i, z_i\right)$ 可以表示为:

$$p\left(x_i, z_i\right) = p\left(z_i\right) p\left(x_i \mid z_i\right) = p\left(z_i\right) \prod_{m=1}^{M} \mathrm{N}\left(x_i \mid \theta_m\right)^{z_i^m}$$

其中 z_i^m 表示隐含变量 z_i 的第 m 维的值(1 或者 0).

对于 z_i 的概率 $p\left(z_i\right)$，当 z_i 的第 m 个元素为 1 的时候，亦即第 m 个高斯分布被选中的时候，这个概率为 c_m ，统一地写出来就是：

$$p\left(z_i\right) = \prod_{m=1}^{M} c_m^{z_i^m}$$

带入上面个式子，我们得到 $p\left(x_i, z_i\right)$ 的概率是一个大的乘积式（对比之前 $P\left(x_i\right) = \sum_{m=1}^{M} c_m \mathrm{N}\left(x_i \mid \theta_m\right)$ 是一个和式）。再替换到最开始的那个 Log-likelihood function 中，得到新的**同时关于 样本 x_i 和隐含变量 z_i** 的 Log-likelihood:

$$\begin{aligned}
\sum_{i=1}^{N} \log\left(p\left(x_i, z_i\right)\right) &= \sum_{i=1}^{N} \log\left(\prod_{m=1}^{M} c_m^{z_i^m} \prod_{m=1}^{M} N\left(x_i \mid \theta_m\right)^{z_i^m}\right) \\
&= \sum_{i=1}^{N} \log\left(\prod_{m=1}^{M} \left(c_m N\left(x_i \mid \theta_m\right)\right)^{z_i^m}\right) \\
&= \sum_{i=1}^{N} \sum_{m=1}^{M} z_i^m \left\{\log c_m N\left(x_i \mid \theta_m\right)\right\}
\end{aligned}$$

情况瞬间逆转，现在**同时关于 样本 x_i 和隐含变量 z_i 的似然函数**中 log 和求和符号相比之前的**只包含样本的似然函数**[2](见脚注)换了个位置，直接作用在普通的高斯分布上了，一下子

[2] $\sum_{i=1}^{N} \log\left(p\left(x_i\right)\right) = \sum_{i=1}^{N} \log\left\{\sum_{m=1}^{M} c_m N\left(x_i \mid \theta_m\right)\right\}$

就变成了可以直接求解的问题。

不过，事情之所以能发展得如此顺利，完全依赖于一个我们伪造的假设：隐含变量的值已知。然而实际上我们并不知道这个值。问题的结症在这里了，如果我们有了这个值，所有的问题都迎刃而解了。回想一下，在类似的地方（比如，在数据挖掘中处理缺失数据的情况），我们是如何处理的呢？

1. 用取值范围类的随机值代替。
2. 用平均值代替。
3. 填 0 或者其他特殊值。

这里我们采取一种类似于平均值的办法：取期望。因为此时我们至少有 样本 x_i 的值，便可以把这个信息利用起来。前面说过， z_i 的每一个元素只有 0 和 1 两种取值，因此按照期望的公式写出来就是：

$$\begin{aligned} \mathrm{E}\left(z_i^m\right) &= 0 \times p\left(z_i^m = 0 \mid x_i\right) + 1 \times p\left(z_i^m = 1 \mid x_i\right) \\ &= p\left(z_i^m = 1 \mid x_i\right) \end{aligned}$$

这里我们还是无法直接求出期望值,联想一下概率论中对这种条件概率的常用解法,不难发现,对上式运用贝叶斯公式进行变形之后,我们可以得到:

$$\begin{aligned} p\left(z_i^m = 1 \mid x_i\right) &= \frac{p\left(x_i, z_i^m\right)}{p\left(x_i\right)} \\ &= \frac{p\left(z_i^m = 1\right) p\left(x_i \mid z_i^m = 1\right)}{p\left(x_i\right)} \\ &= \frac{c_m \mathrm{N}\left(x_i, \theta_m\right)}{\sum_{j=1}^{M} c_j \mathrm{N}\left(x_i, \theta_j\right)} \end{aligned}$$

仔细观察这个期望值,你是否有一种似曾相识的感觉呢?没错,他正好是之前第一步中定义的 $\gamma\left(i, m\right)$! 余下的步骤便变得清晰起来,只需要用这个期望(也就是 $\gamma\left(i, m\right)$)去实现前一节中的第二步(也就是估计有最大可能性的参数),具体做法是:用这个期望值替代 $\sum_{i=1}^{N}\sum_{m=1}^{M} z_i^m \left\{\log c_m N\left(x_i \mid \theta_m\right)\right\}$ 中的未知量 z_i,此后便能解出这个似然函数在偏导数为 0 时对应的 θ_m 和 c_m.(具体解法参考 链接 第三部分)

到此为止,对于 EM 算法在 GMM 中的应用,我们就可以做出如下的理解:

Expectation

直观上: 根据现有的模型（参数），计算各个观测数据输入到模型中的计算结果$\gamma\left(i, m\right)$

数学上: **固定**现有的模型（参数），**求解**隐性变量的期望$\mathrm{E}\left(z_i\right)$

Maximization

直观上: 根据计算结果$\gamma\left(i, m\right)$,用最大似然的方法重新计算模型第 m 个分布的参数

数学上: **固定**隐性变量的期望$\mathrm{E}\left(z_i\right)$,**优化***同时包含样本和隐变量的似然函数*得到新参数

从更抽象的角度理解通俗的 EM 算法

事实上,EM 算法并不仅仅局限于解决 GMM 模型训练,从一个更宽泛的角度去理解 EM 算法将有助于读者在其他问题中活学活用.

为了使讨论衔接,我们依然用之前的符号表示样本和隐变量。我们的问题是要通过最大似然的方法估计出$p\left(x_i \mid \theta\right)$中的参数$\theta$=$\left\{c_m, \mu_m, \Sigma_m\right\}$, $m = 1, \ldots, M$。在这里我们假设这个问题很困难，但是要优化$p\left(x_i, z_i \mid \theta\right)$却很容易。这就是 EM 算法能够解决的那一类问题。

现在我们引入一个关于隐含变量的分布$Q\left(z_i^m\right)$,注意到对于任意的$Q\left(z_i^m\right)$,我们都可以对**只含样本的似然函数**做如下分解:

$$\sum_{i=1}^{N} \log p\left(x_i \mid \theta\right) = \sum_{i=1}^{N} \log \sum_{m=1}^{M} p\left(x_i, z_i^m \mid \theta\right) \quad [1]$$

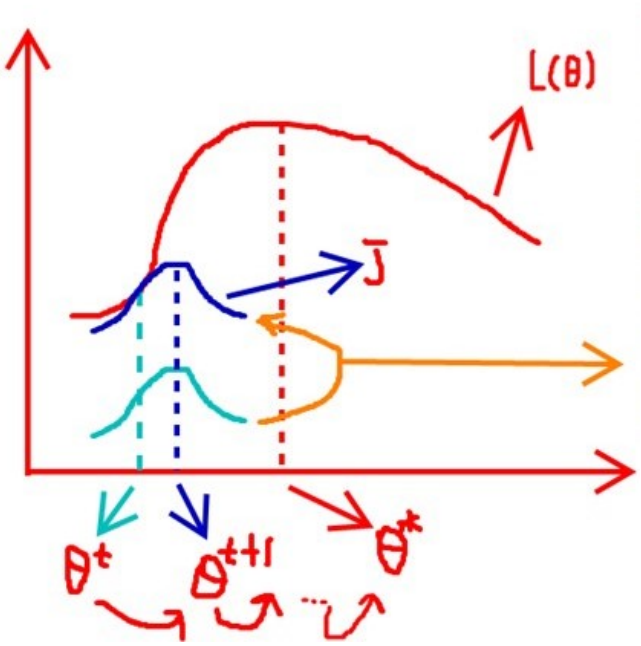
$$= \sum_{i=1}^{N} \log \sum_{m=1}^{M} Q\left(z_i^m\right) \frac{p\left(x_i, z_i^m \mid \theta\right)}{Q\left(z_i^m\right)} \quad [2]$$

$$\geq \sum_{i=1}^{N} \sum_{m=1}^{M} Q\left(z_i^m\right) \log \frac{p\left(x_i, z_i^m \mid \theta\right)}{Q\left(z_i^m\right)} \quad [3]$$

关于第三步中不等式的证明请参考延伸阅读链接,现在,我们只需要知道似然函数存在一个下界:

$$J\left(Q, \theta\right) = \sum_{i=1}^{N} \sum_{m=1}^{M} Q\left(z_i^m\right) \log \frac{p\left(x_i, z_i^m \mid \theta\right)}{Q\left(z_i^m\right)}$$

他是关于参数θ 和隐变量分布 Q 的二元函数.
既然似然函数 **L(θ|x)>=J(Q, θ)**而且 **J 函数相比 L 函数更容易优化**(Σ在 log 外面),那么我们不妨通过**优化这个下界 J,来使得 L(θ|x)不断提高**,最终达到他的最大值.



如上图所示,考虑到 EM 算法的迭代过程,对应的 E 步就是固定 θ,求使得在 θ 点处似然函数值 L=下界函数值 J 时的 Q(图中绿色曲线到蓝色曲线) 之后的 M 步便是固定 Q,调整 θ 使得 J 最大化(图中 $\theta^t \rightarrow \theta^{t+1}$),这时候我们发现似然函数值 L 和下界函数值 J 又不相等了,于是进入下一轮迭代,一直到收敛至似然函数最大时对应的 θ^* 处.

这里,L 函数就是之前说过的**只包含数据的似然函数,** 而 E 步中求得的 Q 就是之前说过的期望 $\mathrm{E}\left(z_i\right)$, M 步中优化的函数 J 则是之前那个**同时包含数据和隐变量的似然函数.**

总结

至此,我们首先以实现 GMM 中的 EM 算法为目的,通过算法流程对 EM 算法有了一个直观的了解,之后引入隐型变量,并尝试从数学角度理解用于 GMM 的 EM 算法,最后,根据总结出的 EM 两个重要步骤,我们抽象出用于一般问题的 EM 算法核心思想,那就是:

通过**引入隐含变量**将似然函数转化为可以求解的形式,然后在隐含变量和参数之间不断**迭代优化**,一直到模型收敛.

延伸阅读:

EM 算法论文 :
EM 算法: 论文 blog(第三部分)
EM 算法收敛性的证明(文章第二部分)
EM 算法严格推导: <Pattern Recognition and Machine Learning> 第九章
wiki 百科 EM 算法动态图(右侧)

4.2 通用背景模型 UBM(universal background model)

世界上没有完美的模型,GMM 也是如此. 虽然 GMM 对数据有着极强的表征力,但是这是以**增加参数为代价**的!如下例所示:

假设对维度为 50 的声学特征进行建模，GMM 包含 M=1024 个高斯分量[3]，并简化多维高斯的协方差为对角矩阵，则一个 GMM 待估参数总量为 1024（高斯分量的总权重数）+1024×50（高斯分量的总均值数）+1024×50（高斯分量的总方差数）=103424，超过 10 万个参数需要估计!

这种规模的变量别说目标用户几分钟的训练数据，就算是将目标用户的训练数据量增大到几个小时，都远远无法满足 GMM 的充分训练要求，然而在实际应用中，从用户体验和成本的角度上考虑(需要花时间对特定说话人采集的语音信息)，针对目标用户可采集到的语料甚至难以超过小时！稀缺的数据极有可能让模型”学的太差”,导致模型识别效果不尽人意.

为了解决数据稀缺导致的训练不充分问题,牛人们提出了一种有效的改进方法: 既然没法从目标用户那里收集到足够的语音，那就换一种思路，可以从其他地方收集到大量非目标用户的声音，积少成多，我们将这些非目标用户数据（声纹识别领域称为背景数据）混合起来充分训练出一个大的 GMM，这个 GMM 可以看作是对**语音的表征**,也就是说,我们用这个大模型学习到了每个人的声音中**共有的特征(common feature)**. 比如:对于单词”yes”,不同的人说这个单词的音调或者音色都不尽相同,但是都在一定程度上符合”yes”的发音特征,这也是上述 GMM 模型被称为通用背景模型(UBM)的原因: **致力于拟合通用的语音特征**.

仅仅有 UBM,是无法完成声纹识别的,因为它是从大量身份的混杂数据中训练而成，所以**不具备表征具体身份的能力**, 或者说,对于能区分出不同的人的**独有特征(specific feature),**UBM 显得无能为力,那么,怎样才能表征这些特定说话人的个性信息呢?

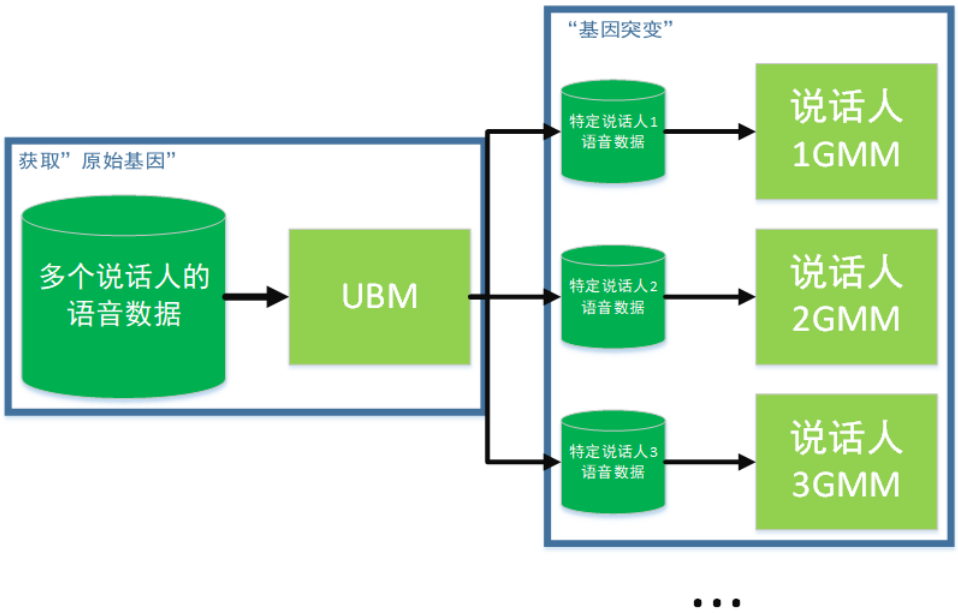
回想一下万亿年前的世界,被海洋包裹的地球只生活着单细胞生物,这些原始的生物经过漫长的进化,得到了如今千差万别的物种,这个过程中发挥了重要作用的就是**基因突变.** 所谓基因突变,是指在原有的基因基础上,产生一些微小的变化,而正是这些极不显眼的变化,造就了大自然的无与伦比的丰富性和多样性.

用这种思想去看待语音信息,我们可以将 UBM 从大量混杂数据中学习到的共有特征当作语音的”原始基因”,在他的基础上,我们用特定的目标说话人数据继续训练 UBM,使得”原始基因”中某一小部分产生”突变”,变化的那部分就很自然地反应了特定说话人的个性信息.在接下来的一节中,我们会具体介绍如何进行所谓的”变异”.

4.3 GMM-UBM 模型

一般性的 GMM-UBM 模型训练流程可用下图表示:

[3] 这里混合高斯分布的子分布个数 M 是个超参数,M 越大代表着需要训练的参数就越多. 他的设定一般需要根据数据量的大小,数据量越大,能训练出的饱和参数就越多,因此,M 就能取越高的值.



他们可以被简单抽象成两个步骤: 获取”原始基因” 和 ”基因突变”.

获取”原始基因”:

由于 UBM 其实就是一个普通的 GMM,因此对于 UBM 的训练,只需要采用上文提到的 EM 算法,就能得到全部的参数(高斯权重、均值和方差).

“基因突变”:

所谓的突变,在具体模型中指的是某个子分布的参数发生了变化,这在数学上可以理解为:特定说话人的语音数据散落在 UBM 某些高斯分布的附近,调整过程就是将 UBM 的每个高斯分布向目标用户数据偏移(如下图所示).

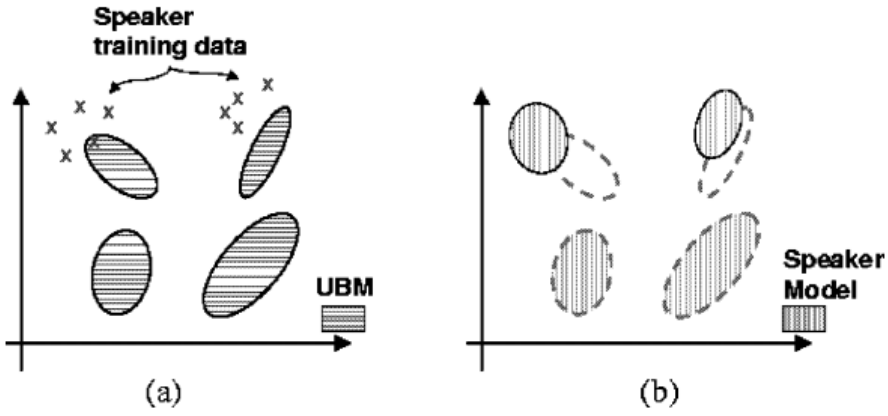


FIG. 3. Pictorial example of two steps in adapting a hypothesized speaker model. (a) The training vectors (x's) are probabilistically mapped into the UBM mixtures. (b) The adapted mixture parameters are derived using the statistics of the new data and the UBM mixture parameters. The adaptation is data dependent, so UBM mixture parameters are adapted by different amounts.

现在,让我们重新对这一步要解决的问题进行描述:我们已经有了**模型的定义(UBM 和特定说话人 GMM 定义相同),**也有了**模型的初始参数(UBM 的参数作为 GMM 初始参数),**同时还有用于**训练的数据**,我们的目标是得到更好的模型参数,你想到了什么?

没错,这不就是**只迭代一次**的 EM 算法要解决的问题嘛!回想一下前文中描述的 EM 算法直观理解,只迭代一次意味着,只进行一次**第一步**和一次**第二步**,其中第一步获得的期望

$$\gamma(i,m)=\frac{c_m N(x_i \mid \theta_m)}{p(x_i)}=\frac{c_m N(x_i \mid \theta_m)}{\sum_{j=1}^{M} c_j N(x_i \mid \theta_j)}$$

在这里可以解释为数据 x_i 和 UBM 中第 m 个高斯分布的拟合程度.此后接着进行第二步,我们就得到了该高斯分布的新参数(权重,均值,协方差).

得到新参数后,我们又遇到了一个更棘手的问题,如何利用他们为模型提供说话人特性信息,或者说,如何利用新参数帮助旧参数完成"突变"?这里有一个很自然的想法:直接用新参数替代旧参数,但是实际上效果很不好,因为旧参数(旧基因)中包含着共同的特征信息,完全舍弃的做法会使得这个子分布丢失一部分表征能力.一个更为妥当的办法就是将新参数和 UBM 原参数按照比例进行"融合":

$$\text{最终参数}=\alpha*\text{新参数}+(1-\alpha)*\text{UBM旧参数}$$

具体比例 α 的计算可以参考论文 Speaker Verification Using Adapted Gaussian Mixture Models 中的 3.4 节.

上述算法被称为极大后验概率(MAP)算法或者自适应算法,我们在此总结出该算法的主要步骤:

1 使用特定说话人语音数据对 UBM 进行一轮 EM 迭代,得到新的参数
2 将新参数和旧参数进行融合

总结

相比原始的 GMM 模型,采用预先训练 UBM 的方式可以减少近一半的待估参数,因为原始方法中,每个特定说话人的 GMM 都是单独训练的,这意味着每个说话人 GMM 模型中的参数(权重,均值,协方差)都要进行估计!而在 GMM-UBM 中,GMM 由 UBM 获得,其中只有需要"突变"的 UBM 参数才会被调整,某种程度上,未突变的参数便可以看作是不同说话人 GMM 的"共享参数".这种方式大大减少了训练参数和训练时间.

延伸阅读

GMM-UBM 论文第三节关于 GMM 新参数的更新

5 未知语音评判打分

假设我们已经训练好了所有说话人的 GMM 模型(知道了所有的参数),对于一段未知的语音很容易就能得到第 s 个说话人模型产生这个语音的概率 $p(y \mid \lambda_s)$,其中 y 表示待测语音,

$\lambda_s = \{u_m, \Sigma_m, c_m\}_{m=1}^{M}$ 表示第 s 个说话人的 GMM 模型参数.这里便能自然地想到一个很简单的打分方法:遍历所有说话人 GMM 模型,找到 $p(y \mid \lambda_s)$ 最大的那个,既然他产生待测语音的概率最大,就认为这段语音是他说的.

更进一步,我们还可以利用之前的 UBM 模型,用 $p(y \mid \lambda_{UBM})$ 表示语音不是被第 s 个 GMM 产生的概率,两者相比得到:

$$\frac{p(y \mid \lambda_s)}{p(y \mid \lambda_{UBM})} \begin{cases} \geq \xi, \text{认为说话人是s} \\ < \xi, \text{认为说话人不是s} \end{cases}$$

其中 ξ 为判决阈值(decision threshold).

延伸阅读

GMM-UBM 论文第二节关于判断似然性计算

6 评测声纹识别系统性能

我们成功建立了模型之后，如何评价模型的好坏变成了一个重要的因素来判断我们的成果。我们介绍如下指标来衡量。

6.1 基本技术指标

声纹识别在算法层面可通过如下基本的技术指标来判断其性能：

错误拒绝率 （False Rejection Rate, FRR）：分类问题中，若两个样本为同类（同一个人），却被系统误认为异类（非同一个人），则为错误拒绝案例。错误拒绝率为错误拒绝案例在所有同类匹配案例的比例。

错误接受率（False Acceptance Rate, FAR）：分类问题中，若两个样本为异类（非同一个人），却被系统误认为同类（同一个人），则为错误接受案例。错误接受率为错误接受案例在所有异类匹配案例的比例。

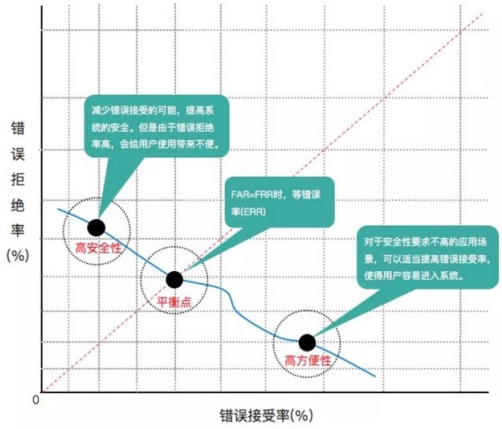
等错误率 （Equal Error Rate，EER）：调整阈值，使得误拒绝率（False Rejection Rate，FRR）等于误接受率（False Acceptance Rate，FAR），此时的 FAR 与 FRR 的值称为等错误率。

准确率 （Accuracy，ACC）：调整阈值，使得 FAR＋FRR 最小，1 减去这个值即为识别准确率，即 ACC=1 － min(FAR+FRR)

速度： （提取速度：提取声纹速度与音频时长有关、验证比对速度）：Real Time Factor 实时比（衡量提取时间跟音频时长的关系，比如：1 秒能够处理 80s 的音频，那么实时比就是 1:80）。验证比对速度是指平均每秒钟能进行的声纹比对次数。

ROC 曲线： 描述 FAR 与 FRR 之间相互变化关系的曲线，X 轴为 FAR 的值,Y 轴为 FRR 的值。从左到右，当阈值增长期间，每一个时刻都有一对 FAR 和 FRR 的值，将这些值在图上描点连成一条曲线，就是 ROC 曲线。

阈值： 在接受/拒绝二元分类系统中，通常会设定一个阈值，分数超过该值时才做出接受决定。调节阈值可以根据业务需求平衡 FAR 与 FRR。 当设定高阈值时，系统做出接受决定的得分要求较为严格，FAR 降低，FRR 升高；当设定低阈值时，系统做出接受决定的得分要求较为宽松，FAR 升高，FRR 降低。在不同应用场景下，调整不同的阈值，则可在安全性和方便性间平平衡，如下图所示：



6.2 性能指标

除了基本技术指标之外，还有其它的一些在实际应用场景中的性能指标如下：

1. 环境噪音鲁棒性

不同场景下的产品都会有不同的环境噪音，即使是同一产品也会有不同的背景环境，比如智能音箱，在家庭使用和在公司使用，环境噪音也会不一样，在使用声纹识别前需要对这一黑科技的环境噪音鲁棒性进行评估，这一指标表明此技术在不同环境噪音下的适应能力，避免在公司调试时都是好好的，一到用户环境就不灵光了。为了测试声纹识别系统的环境噪音鲁棒性，可以收集产品在不同应用环境下的语音数据进行评测。

2. 信道鲁棒性

信道即为声音信号传输的通道，由于声音从麦克风采集后到声纹识别系统中经过了很多环节，包括有不同的麦克风类型、不同的音频 CODEC、不同的传输通道等，这些都会对声纹特征存在影响，还是以智能音箱来举例，假如在注册时是用手机端 app，而验证使用时则是直接对着音箱说话，手机 MIC 和音箱 MIC 就是两个不同的信道，这种情况下可能会降低验证的准确率，在专业术语上叫信道失配。因此，除了在产品层面做规避，也需要考虑声纹识别技术在不同信道中的表现。

3. 语音内容鲁棒性

我们说话内容都可能包含了数字 、中文、英文，在读特定内容和说口头禅的时候，我们会不自觉表现不一样的说话方式，比如说口头禅或熟悉的话时就会表现得很自然随意，而拿着文稿照着念时，就显得一本正经。在做声纹识别技术评估时，也需要考虑到对语音内容的鲁棒性。

4. 时变鲁棒性

个体变化通过长时的积累，会对个体的发音有特点有影响，进而影响声纹识别系统的识别性能。好的声纹识别系统能在一年，甚至在三年内都不需要重新注册而能正常使用，否则你可能会遇到，三个月前注册了声纹用着都是好好的，三个月后怎么就不认人了呢，这就尴尬了。

5. 表达方式鲁棒性

说话人的表达方式对声纹识别的性能也有影响，比如情感的变化、语速的变化、音量的变化和聊天的区别。还是以智能音箱为例，你在注册声纹时是很开心的，当有一天，你心情不好想和 TA 聊天时，却怎么也不认你了，这时你砸了 TA 的心都有了。同样，在做声纹识别评估时都需要考虑到在不同表达方式下的表现。

6. 群体普适性

群体是具有某种(些)共同特征的不同个体组成的集合。不同群体之间存在某些特征的差异，声音上的差异就是其中之一，这种差异会影响声纹识别系统的普适性。这种差异主要体现在性别、年龄、地域划分的不同人群人声纹差异。

7. 假冒攻击防范能力

2017315 用照片直接攻破人脸识别系统的事仍让大家对生物识别系统有所担心，同样，声纹识别系统在用声音进行身份认证的过程中，也会存在用假冒声音来企图骗过系统，因此，声纹识别系统应具备活体检测技术，应正确鉴别声音的用户身份，能够拒绝假冒的验证信息，对于利用各种手段形成的假冒声音，应该能正确区分。

假冒声音包括通过如下几种方式生成的声音，声纹识别系统应提供对如下几种攻击的防范能力。

7.1.波形拼接攻击

攻击者将目标说话人的语音录制下来，通过波形编辑工具，拼接出指定内容的语音数据，以放音的方式假冒目标说话人，试图以目标人身份通过声纹识别系统的认证。

7.2.录音重放攻击

攻击者录制目标说话人的语音进行播放，以目标人身份试图通过声纹识别系统的认证。

7.3.语音合成攻击

攻击者用语音合成技术生成目标说话人的语音，以放音的方式假冒目标说话人，试图以目标人的身份通过声纹识别系统的认证，

7.4.语音转换攻击

攻击者用语音转换技术得到目标说话人的语音，以放音的方式假冒目标说话人，试图以目标说话人的身份通过声纹识别系统进行的认证，

7.5.语音模仿攻击

攻击者通过模仿目标说话人，试图以目标说话人的身份通过声纹识别系统的认证。