第 6 章 说话人识别原理与技术

【内容导读】智能音箱兴起之时，每当电视广告里演示音箱唤醒功能时，家里的音箱也会被随之误唤醒。想象一下，千家万户的音箱在同一时刻被同一唤醒命令唤醒，遥相呼应，可谓壮观。然而如何只让音箱的主人能唤醒自家的音箱呢？如何让自己的各类智能设备，不论是音箱、手机语音助手、智能家居设备、智能车载系统还是未来会普及的各类家用机器人都只听命于自己呢？那就不可避免地会用到说话人识别技术。本章将从介绍说话人识别的基本概念、发展历程、经典模型和识别方法，主要包括 GMM-UBM、i-vector、PLDA、基于深度学习的声纹识别模型。

6.1 说话人识别概述

6.1.1 说话人识别概念

说话人识别(Speaker Recognition)，俗称声纹识别(Voiceprint Recognition)，也称话者识别，是一种重要的现代身份认证技术，与指纹识别、掌纹识别、人脸识别和虹膜识别等生物识别技术同属于模式识别的范畴[1]。由于每个人的发音器官(声带、咽腔、口腔、鼻腔、舌头、牙齿、嘴唇、软腭等)和发音习惯(语速、语调、韵律、地域口音和个性化口音等)都有很大的差异性，加之年龄、性别、性格等个性化因素的影响，每个人的“声纹”也具有唯一性。声纹的概念最早源于军事和刑侦活动中的语谱图纹路分析，研究人员发现不同人即便发相同的音，在纹路上也有细微差异[2]。声纹不仅具有唯一性，而且具有动态稳定性的特点，因此常被用来进行身份认证。

一段语音中不仅包含了语言语义信息，同时也包含了韵律特征(即音高、时长、强度等超音段特征)传递的副语言信息(“怎么说”)和非语言信息(“谁在说”)。 [3]。语音识别是共性识别，强调文本内容，而不考虑说话人是谁；说话人识别是个性识别，只关注说话人信息，不考虑文本内容。在语音信号处理中，识别某种属性信息需要抑制其他属性信息表达而只强化该主属性信息表达。例如语音识别技术关注于语言语义信息而抑制说话人、情感等副语言、非语言信息表达；情感识别技术抑制非情感信息的表达；同理，说话人识别技术更多关注于说话人身份信息而抑制文本语义等信息表达。各类信息互为补充，从各个层面反应出丰富的整体语音信息。由此可见，说话人识别技术与其他语音技术相辅相成，是完备语音智能体中不可或缺的重要一环。

应用现代声纹技术设计的自动说话人识别系统可以自动辨识和确认说话人身份，其通常由训练阶段(Training stage)和测试阶段(Test stage)两部分组成。训练阶段通过对丰富的说话人训练语料进行说话人特征表征和分类模型训练，得到具备高区分度的说话人识别模型；测试阶段将待识别说话人(Test speaker)语音特征输入到训练好的说话人识别模型，进而判定其是否来自注册集合的目标说话人(Target speaker)。注册集合中说话人与训练集合的说话人是否保持一致取决于该任务是开集还是闭集。说话人识别的经典流程图如图 6-1 所示。

根据任务类型的不同，说话人识别一般可分为两类：

(1)**说话人辨认**(Speaker Identification)是判定待识别说话人来自注册集中的哪个说话人，是“一对多”的任务。其又可根据待识别说话人是否一定来自注册集中的说话人细分为闭集辨认和开集辨认。例如，刑侦案件中判定已收集到的犯罪嫌疑人的声音来自一群人中的哪个人，

进而指认罪犯。如果待识别声音有可能不来自其中的任何一个人，那么就是开集辨认。常用的性能评价指标有说话人辨认错误率等。说话人辨认错误率即辨认错误的样本数占整体样本的比例。

(2)**说话人确认**(Speaker Verification)[4]是判定待识别语音是否来自注册集中的某个说话人，是“一对一”的问题。即判断待识别语音与指定的某一说话人是否一致，答案只有是或否。常用的性能评价指标有等错误率(EER)、检测代价函数(Detection Cost Function, DCF)等。其中等错误率指的是错误接受率和错误拒绝率相等处的错误率，检测代价函数则通过设置错误拒绝和错误接收的惩罚代价来判别说话人验证的效果。详见 6.5.2 节。

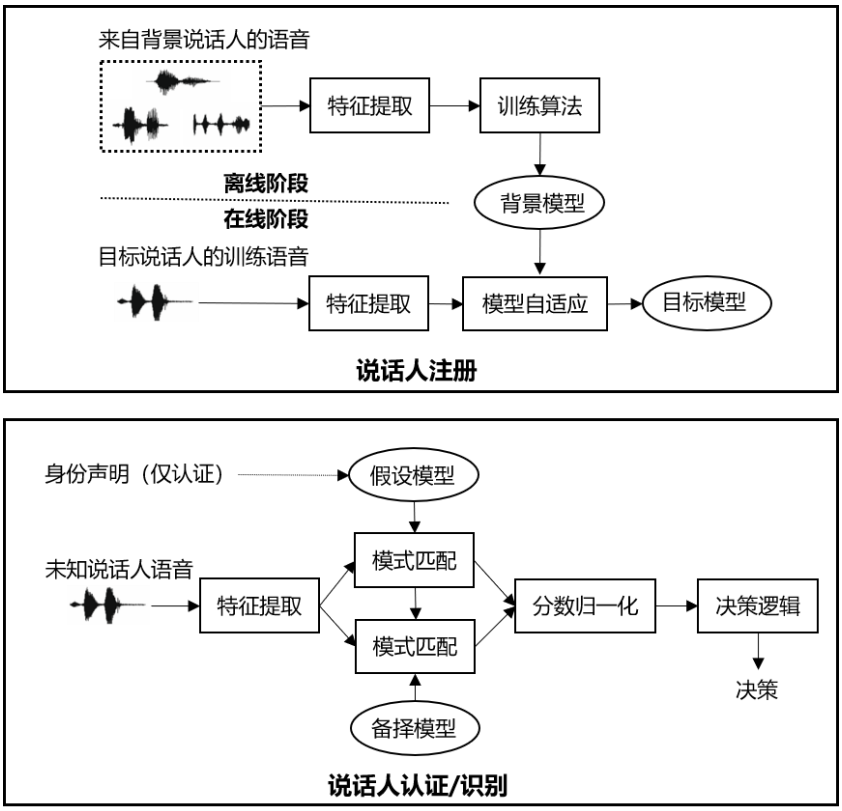


图 6-1 说话人识别的流程图(图引自[1])

以上分类均为一段音频中只包含单一说话人。说话人识别中还有另外一个重要的研究方向，即一段音频中两个及以上的说话人交替说话，需要对所有说话人讲话的时间起始点进行界定，称之为说话人分割聚类(Speaker Diarization)，也称说话人追踪、说话人日志等，该技术常被用于会议纪要等场景。三者的关系如图 6-2 所示。

根据语音文本内容，说话人识别一般又可分为两类:

(1)**文本相关**(Text-dependent)要求说话人说特定的文本，文本内容与训练阶段一致，或者现场提示[[6],[7],[8],[9]]。其一定是语种相关的。

(2)**文本无关**(Text-independent)不限定说什么文本[[10],[11],[12]]，语种相关和语种无关(跨语种说话人识别)均可。

说话人研究中一般会指出具体的研究类型，如“文本无关的说话人确认”。上述讨论的是一段语音为单一说话人的一般情况。

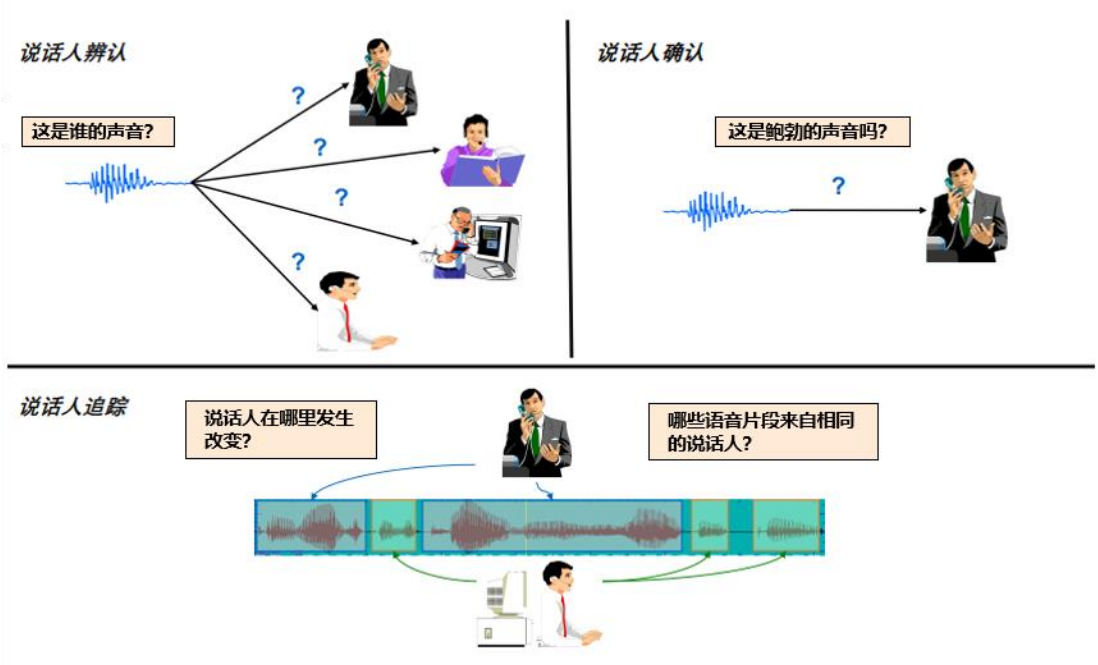


图 6-2 说话人识别任务的分类(图引自[5])

6.1.2 说话人识别技术优势

语言是人类文明的智慧结晶，言简意赅，语音作为其声音载体，具有视觉等其他感官不具备的得天独厚的优势。语音是人类获取信息的主要形式之一，也是人与外界交流中使用最方便、最有效、最自然的交际工具和信息载体。说话人识别与其他生物身份认证方式相比，具有以下优势:

(1)安全性与交互性兼备。说话人识别技术既可应用于银行、安保等对安全性要求高的场景，也可广泛应用于智能家居、公共服务等对安全性没那么高但是需要良好的交互性等场景。因为说话人识别领域里的评价指标等错误率由错误拒绝率(False Rejection Rate, FRR)和错误接受率(False Acceptation Rate, FAR)组成，也称“拒真率”和“认假率”，两者可以通过阈值的设定动态偏向安全性还是交互性的应用场景。语音的双向传递特点也大大提升了交互体验。

(2)可穿透性和光鲁棒性。声音可以穿透墙壁等障碍物，同时不受恶劣光线环境的干扰影响，相比与人脸识别和虹膜识别具有不可比拟的优势。

(3)非接触性。说话人识别不需要待识别者接触识别设备，相比于指纹识别和掌纹识别有一定的优势，特别是在突发的公共卫生危机、远程认证等方面有巨大的应用潜力。

(4)泄露安全性。语音信号是唯一性和高可变性的完美统一，虹膜和指纹等生物特征是静态的，样本一旦泄露具有很大的危险，而语音信号的动态可变性一定程度上可以增强其安全性。

(5)采集成本低。语音采集设备相比于指纹设备和照相设备成本极低，手机、普通麦克风均可完成采集任务。数据易获得。

6.1.3 说话人识别应用前景

说话人由于其无可比拟的技术优势，被广泛应用于军事活动、刑事侦探、金融安全、国

防安全、公共安全等安全场景[[13],[14],[15],[16]]。例如说话人辨认可以应用于军事和刑侦领域对于目标说话人或犯罪嫌疑人的追踪，以及保险诈骗等金融安全活动。说话人确认可用于声纹门锁以及智能家居、车载语音系统以及公共服务领域等，提升人机交互体验。说话人分割聚类可用于刑侦记录、会议纪要等档案记录场景，减少传统记录方式的工作量和难度。

此外，与说话人识别相关的新兴的研究方向也在不断发展，如反声纹欺诈，用来检测模仿、语音合成、语音转换以及录音回放。随着语音合成技术的日趋成熟，仿冒的声音越来越逼真，在不久的将来只依靠人耳很可能无法区分到底是仿冒的说话人还是真实的说话人。到那时，反声纹欺诈系统会像现在的防火墙系统一样不可或缺。

6.1.4 说话人识别技术难点

近些年说话人识别技术得到快速发展，但尚未达到非常成熟的应用程度，特别是在高精度的安全领域。越来越复杂的应用场景也导致对于说话人识别系统的鲁棒性要求越来越高。纵观影响说话人识别技术发展的若干因素，主要的技术难点和瓶颈问题可总结为以下几个方面：

(1)噪声和混响的干扰。当前说话人识别技术对于近场安静的识别环境已经可以达到很好的性能，但现实场景中更为普遍的是复杂场景下的说话人识别。噪声和混响会很大程度损坏音频的质量，降低识别性能。

(2)跨设备识别。单一信道(如 android、ios 或高保真麦克风)的说话人识别取得了令人满意的成绩，但是真实场景下训练、注册、评估的说话人音频往往不能保证来自同一信道。消除信道之间的差异还需进一步的研究。

(3) 短语音识别。智能家居和公共服务领域对于说话人识别通常使用很短的控制命令，短语音识别相对于长语音包含信息更少，为识别增加了难度。

(4)仿冒语音攻击。模仿、语音合成、语音转换和录音回放对于自动语音识别系统造成了巨大的威胁，大大降低了系统的识别性能。

(5)跨语种识别。训练和测试阶段使用不同语种(或者方言)会导致语言失配，降低识别准确率。

(6)时变影响。人的发音器官随着年龄增长会发生变化，特别是变声期后，相对应的声纹也会发生改变。其他因素如生病、情绪等也会影响声纹发生改变。

(7)性别影响。男性和女性发音器官存在很大差异，性别差异会导致说话人模型出现失配现象。

目前，对于说话人识别的研究主要围绕上述几个方面展开，主要目标是提升声纹系统的鲁棒性和识别性能，特别是研究声纹的动态不变性和唯一性之间的关联。不论是信道、语种失配还是环境场景、文本长度失配，消除说话人无关因素的影响，更多保留说话人自身的特点是下一步研究的重点和难点。

6.1.5 说话人识别发展历程

“未见其人，先闻其声”，根据声音辨别身份是人类和动物的本能。随着人工智能时代的到来，借助于机器超大容量的存储记忆和运算能力，自动说话人验证系统可以准确区分成千上万的说话人，远远超过了人的辨识记忆能力。早在 20 世纪 40 年代，说话人识别就已经被应用于军事活动和刑侦案件中，60 年代之后逐渐由人耳听辨过渡到机器自动识别。70 年代到 80 年代涌现了以动态时间规整(Dynamic Time Warping, DTW)[17]，矢量量化(Vector Quantization, VQ)[18]和隐马尔可夫模型(Hidden Markov Model, HMM)[19]等一大批有效的特

征方法和说话人识别模型。

在特征提取及表征方面，经历了从早期的基于产生机理、听觉机理等知识驱动(knowledge-based)的声学层面特征到后来的基于数据驱动(Data-driven)的表征层面特征。声纹领域具有代表性的基于听觉机理的特征有语谱图(Spectrogram)，滤波器组特征(Filter Bank Feature, fbank)，梅尔倒谱系数(Mel Frequency Cepstral Coefficients, MFCC)，感知线性预测(Perceptual Linear Prediction，PLP)，基于语音产生机理的逆求解(Acoustic-to-articulatory inversion)[20]。特征和声门(Subglottal)[21]特征、基音(Pitch)特征、韵律(Prosody)特征、倒谱(Cepstrum)特征等。这些特征大部分都是从语音识别等其他方向引入到说话人识别领域，在信号处理中具有通用性的特征，但这些底层的大部分声学特征对于说话人识别并不具备特定的说话人领域的先验知识。因此，在这些底层特征基础之上，通过引入说话人先验知识，侧重对说话人身份信息表达的 i-vector 表征层特征被提出。以上特征虽然都取得了不错的效果，但基于先验知识的特征分析都是无监督的学习过程，在分类之前都不会受到后端属性标注的指引，因此说话人的区分性会较差。

90 年代高斯混合模型(Gaussian Mixture Model, GMM)由于其更好的拟合能力被人们广泛应用于描述语音特征序列。接下来的 20 年内，高斯混合模型-通用背景模型(Gaussian Mixture Model-Universal Background Model, GMM-UBM)、联合因子分析(Joint Factor Analysis, JFA)和身份向量(Identity-vector, i-vector)模型陆续被提出并得到广泛认可，其中 i-vector 模型[22]由于鲁棒的性能成为 2010 年前后说话人识别领域的经典模型。这些模型基本都是基于生成式模型(Generative model)。生成模型通常假设数据总体服从某个分布，这个分布可以由一些参数确定，如正态分布由均值和标准差确定，在此基础上对联合分布概率进行建模。生成模型通常不需要大量的训练数据，但是不是以直接学习说话人之间的可区分性为最终目标。近 10 年来，基于深度学习的深度说话人模型得到快速发展，其中最具有代表性的是 x-vector 模型。这类判别式模型(Discriminative model) 直接以非线性函数的方式对条件概率进行建模，能够找到不同说话人之间的更好的分类界面以获得更好的说话人识别性能。通过复杂的网络层级结构加上高效的可区分性说话人训练方法，实现更强的说话人表征能力。从早期的融入 ASR 音素层信息的 DNN/i-vector[23]，针对语音时序性提出的 LSTM[6]，到基于 DNN 的 d-vector、TDNN 的 x-vector[24]以及 CNN 的 deep embedding[25]，与说话人相关的特征表达进一步得到加强，与说话人无关的信道信息、语言信息等被进一步剥离。但由于神经网络强大的训练模式背后仍是可解释性差的黑盒训练机制，通常需要比生成式模型更大量的数据，模型也更容易过拟合，如何将领域先验知识与强大的神经网络结合仍是下一步的研究重点。

上述的特征提取和建模方法展示了可区分性说话人训练的模型发展历程，模型建立之后，仍需要相应的分类模型对生成的说话人特征进行分类处理。常用的分类模型有 Cosine similarity[25]，线性鉴别分析(Linear Discriminative Analysis，LDA)[26]，概率线性鉴别分析(Probabilistic Linear Discriminative Analysis，PLDA)[27]等。近些年来，端到端模型的提出将前端特征学习(提取)和后端分类融为一体进行有监督学习，在这种体系下，说话人识别系统不再严格区分前端和后端的概念，一体化的学习可以最大化区分训练集中的不同说话人。

近二十年来，世界各国都对这个研究方向保持高度重视。自 1994 年开始，每年美国国家标准技术研究院(National Institute of Standards and Technology, NIST)都组织全世界范围的说话人识别评测(Speaker Recognition Evaluation, SRE)[28]或语种识别评测(Language Recognition Evaluation, LRE)[29],吸引众多国际一流语音研究机构参与。尽管随着研究的不断深入和发展，识别错误率持续下降，测试任务难度逐渐提高，但在实际应用环境下与我们所期待的性能指标仍有差距。除了 SRE 系列比赛，还有 Voxceleb[30]，ASVspoof Challenge[31]等定期举办旨在解决声纹相关难题的比赛。声纹识别的准确度和鲁棒性也在不断朝大规模商

用的方向快速发展着。

6.2 传统说话人识别算法

6.2.1 经典前端特征

常用的声纹特征主要是短时频谱特征，基于声道的共振规律和语音信号的短时平稳假设，对语音信号进行加窗、分帧，计算得到每一帧语音的频谱特征。常见的短时频谱特征有：语谱图(Spectrogram)[32]、线性预测倒谱系数(Linear Prediction Cepstral Coefficients, LPCC)[33]、梅尔频率倒谱系数(Mel Frequency Cepstral Coefficents, MFCC)[26]、感知线性预测(Perceptual Linear Prediction, PLP)[34][35]等。这些声学特征大都是基于语音生成机理或者听觉感知机理。其中，MFCC 系列特征是声纹识别中最常用的底层声学特征，无论是传统的基于 i-vector 的说话人识别系统，还是基于 x-vector 的说话人识别系统，都以 MFCC 特征作为前端输入特征。除此之外，常用的说话人识别特征还包括描述声门激励的特点的声源特征，描述语音信号动态特性的时序动态特征，描述语音信号中的音节重音、语调、语速和节奏的韵律特征以及音素等常用语辅助说话人识别的语言学特征。

在反声纹欺诈领域中，常用的特征还有常数 Q 倒谱系数(Constant Q Cepstral Coefficients，CQCC)[36]特征，基于相位信息[37]的特征。

6.2.2 经典识别模型

6.2.2.1 GMM-UBM [38]

高斯混合模型(GMM)是一种随机模型，已成为说话人识别系统中被广泛使用的模型。它将空间分布的概率密度用多个高斯概率密度函数的加权来拟合，可以平滑的逼近任意形状的概率密度函数，并且是一个易于处理的参数模型，具备对实际数据极强的表征力。然而，GMM 模型的效果与其训练的数据以及参数量是成正比的，GMM 规模越庞大，表征力越强，参数规模也会等比例的膨胀，需要更多的数据来驱动 GMM 的参数训练才能得到一个更加泛化的 GMM 模型。

在实际场景中每一个说话人的语音数据很少，这将导致无法训练出高效的 GMM 模型。并且由于多通道的问题，训练 GMM 模型的语音与测试语音存在失配的情况，这些因素都会降低声纹识别系统的性能。所以 DA Reynolds 的团队提出了一个通用背景模型(Universal Background Model, UBM)。我们可以用 UBM 和少量的说话人数据，通过自适应算法(如最大似然线性回归 MLLR[39]、最大后验概率 MAP[40]等)来得到目标说话人模型。在基于 GMM-UBM 的说话人识别系统中，首先使用 EM 算法对说话人无关的领域模型或通用背景模型(UBM)进行训练，通过从大量的说话人那里收集的几十个或数百个小时的语音数据。背景模型表示说话人无关的特征向量的分布。在系统中加入新的说话人时，背景模型的参数与新说话人的特征分布相适应。然后，采用经过调整的模型作为说话人的模型。这样，就不用从零开始估计模型参数，而是利用先验知识(“通用的语音数据”)。可以根据背景模型调整所有参数，或者只调整其中的一些参数。

图 6-3 展示了 GMM-UBM 说话人识别模型的自适应过程。

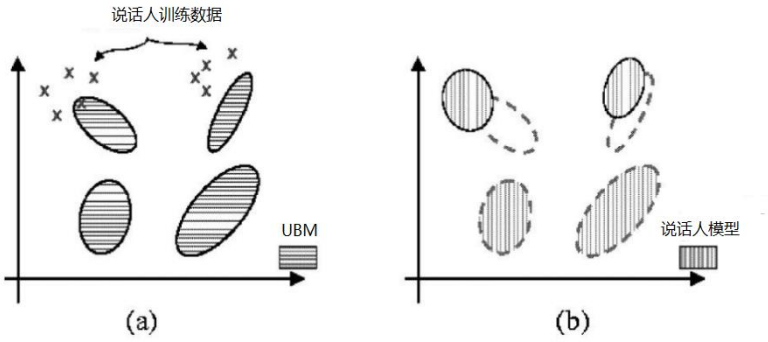


图 6-3 GMM-UBM 模型中的 MAP 算法自适应(图引自[41])

1.首先利用来自不同说话人的大量语音数据建立一个相对稳定且与说话人特性无关的高斯混合模型(GMM)，该模型为通用背景模型(UBM)。如图 6-3(a)所示。

2.基于最大后验估计算法(MAP)，利用说话人的语音数据在 UBM 上自适应得到该说话人的 GMM。如图 6-3(b)所示，该说话人的每个声学子空间(即为一个 GMM 混合分量)由一个说话人相关的高斯分布所描述；而该说话人相关的高斯分布是由与其对应的说话人无关的高斯分布通过 MAP 自适应得到。

具体的计算过程如下：

1. 给定 UBM 模型与说话人的训练数据集$\mathbf{X} = \{x_1, \ldots, x_t, \ldots, x_T\}$，计算$x_t$与 UBM 第$i$个高斯分量分布的相似度：

$$\Pr(i|x_t) = \frac{\boldsymbol{\omega}_i \boldsymbol{P}_i(x_t)}{\sum_{j-1}^{M} \boldsymbol{\omega}_j \boldsymbol{P}_j(x_t)} \tag{6. 1}$$

2.计算新的权重、均值和方差参数：

$$n_i = \sum_{t=1}^{T} \Pr(i|x_t) \tag{6. 2}$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|x_t) x_t \tag{6. 3}$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|x_t) x_t^2 \tag{6. 4}$$

3.得到的新参数和 UBM 原参数融合，得到最终的目标说话人模型：

$$\overline{\boldsymbol{\omega}}_i = [\boldsymbol{\alpha}_i^w \boldsymbol{n}_i / T + (1 - \boldsymbol{\alpha}_i^w) \boldsymbol{\omega}_i] \gamma \tag{6. 5}$$

$$\overline{\boldsymbol{\mu}}_i = \boldsymbol{\alpha}_i^m E_i(x) + (1 - \boldsymbol{\alpha}_i^m) \boldsymbol{\mu}_i \tag{6. 6}$$

$$\overline{\boldsymbol{\sigma}}_i^2 = \boldsymbol{\alpha}_i^v E_i(x^2) + (1 - \boldsymbol{\alpha}_i^v)(\boldsymbol{\sigma}_i^2 + \boldsymbol{\mu}_i^2) - \overline{\boldsymbol{\mu}}_i^2 \tag{6. 7}$$

其中归一化因子γ使得各混合度的权重之和为 1，$\{\boldsymbol{\alpha}_i^w, \boldsymbol{\alpha}_i^m, \boldsymbol{\alpha}_i^v\}$为自适应参数，满足：

$$\boldsymbol{\alpha}_i^\rho = \frac{\boldsymbol{n}_i}{\boldsymbol{n}_i + r^\rho} \tag{6.8}$$

其中自适应参数$\boldsymbol{\alpha}_i^\rho$，$\rho \in \{w, m, v\}$，$r^\rho$为常数，一般取$r^\rho$=16。

GMM-UBM 模型最重要的优势就是通过自适应算法对模型参数进行估计，避免了过拟合的发生，同时我们不必调整目标用户 GMM 的所有参数(权重，均值，方差)只需要对各个高斯成分的均值参数进行估计，就能实现最好的识别性能。实验表明，这可以让待估的参数减少超过一半，越少的参数也意味着更快的收敛，不需要大量的目标用户数据即可完成模型的良好训练。

6.2.2.2 JFA(联合因子分析)

在传统的基于 GMM-UBM 的识别系统中，由于每个高斯成分相对独立，不具有相关性，使得不同子空间之间无法实现信息共享。2005 年，P. Kenny 提出了联合因子分析(Joint Factor Analysis,JFA)[42]方法。该方法把 GMM 均值向量表示的超向量空间进行了分解。JFA 认为，说话人的 GMM 模型的差异信息，是由说话人差异和信道差异这两个不可观测的部分组成的，公式如下：

$$\boldsymbol{M} = \boldsymbol{S} + \boldsymbol{C} \tag{6.9}$$

其中，$\boldsymbol{S}$ 为说话人相关的子空间，表示说话人之间的差异；$\boldsymbol{C}$ 为信道相关的子空间，表示同一个说话人不同语音段的差异；$\boldsymbol{M}$ 为 GMM 均值超矢量，是说话人相关部分 $\boldsymbol{S}$ 和信道相关部分 $\boldsymbol{C}$ 的叠加。

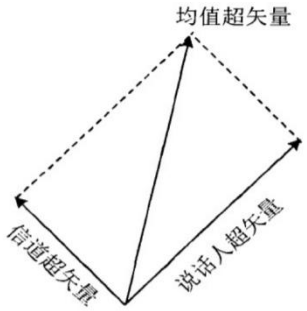


图 6-4 均值超矢量分解示意图(图引自[43])

如图 6-4 所示，联合因子分析实际上是用 GMM 超矢量空间的子空间对说话人差异以及信道差异进行建模，从而便可以去除信道的干扰，得到对说话人身份更精确的描述。

说话人子空间 **S** 是由语言因子 $\boldsymbol{m}$、说话人因子 $\boldsymbol{x}$ 和残差因子 $\boldsymbol{y}$ 三个变量经过线性变化所产生的。$\boldsymbol{x}$、$\boldsymbol{y}$ 服从 N(0,1)的高斯分布：

$$\boldsymbol{S} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{y} \tag{6.10}$$

信道子空间 $\boldsymbol{C}$ 则是由表征信道特性的信道因子来产生的，$\boldsymbol{z}$ 服从N(0,1)的高斯分布：

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{z} \tag{6.11}$$

6.2.2.3 i-vector

在 GMM-UBM 模型里，每个目标说话人都可以用 GMM 模型来描述。因为从 UBM 模型自适应到每个说话人的 GMM 模型时，只改变均值，对于权重和协方差不做任何调整，所以说话人的信息大部分都蕴含在 GMM 的均值里面。GMM 均值矢量中，除了绝大部分的说话人信息之外，也包含了对应语音的说话人、信道、语种、情感等信息。然而 GMM 获取的均值向量维度较高，为了方便后续信道补偿以及比对打分，需要寻找在低维度空间内保留判别性的特征表示来代表说话人。联合因子分析(Joint Factor Analysis, JFA)可以对说话人差异和信道差异分别建模，从而可以很好的对信道差异进行补偿，提高系统表现，然而 JFA 估算出来的说话人子空间与信道子空间存在互相掩盖的问题。

由此 N. Dehak 提出了基于单因子分析的 i-vector[44]，单因子分析在超向量空间内不严格区分说话人空间以及信道空间，直接估算包含说话人信息与信道信息的总体差异空间(Total Variability Space) 将超向量映射到总体差异空间中，以此得出对应语音的包含说话人及信道信息的低维度的 i-vector，将说话人差异和信道差异作为一个整体进行建模。这种方法改善了 JFA 对训练语料的要求，和计算复杂度高的问题，被各个研究团队广泛使用。

给定说话人的一段语音，与之对应的高斯均值超矢量定义如下：

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{\omega} \tag{6. 12}$$

与 JFA 不同的是，i-vector 将说话人子空间 $\boldsymbol{S}$ 和信道子空间 $\boldsymbol{C}$ 统一在一个全变量空间 $\boldsymbol{T}$ 中，采用“全局差异空间因子”$\boldsymbol{\omega}$ 同时描述说话人因子 $\boldsymbol{x}$ 和信道因子 $\boldsymbol{z}$。

其中：

$\boldsymbol{m}$ 为与说话人及信道无关的均值超矢量，即为 UBM 的均值超矢量，该超矢量与具体说话人以及信道无关；

$\boldsymbol{T}$ 为低秩总体差异子空间，是一个低秩的矩阵；

$\boldsymbol{\omega}$ 为全局差异空间因子，它的后验均值，即为 i-vector 矢量，它先验地服从标准正态分布；

在给定的公式中，$\boldsymbol{m}$ 与 $\boldsymbol{M}$ 可以预先计算出，而全局差异空间矩阵(**T**)和全局差异空间因子($\boldsymbol{\omega}$)是我们需要估计的。

因此，在 i-vector 系统中，我们需要基于以下两个关键步骤：

1. 全局差异空间矩阵 $\boldsymbol{T}$ 的计算；
   (1) 假设每一段语音都是来自不同的说话人；
   (2) 计算训练数据库中每个音频所对应的 Baum-Welch 统计量；
   (3) 随机产生 $\boldsymbol{T}$ 的初始值。采用如下 EM 算法，迭代估计 T 矩阵：
   ① E-Step：计算隐变量 $\boldsymbol{\omega}$ 的后验分布，$\boldsymbol{\omega}$ 的后验均值和后验相关矩阵的期望形式；
   ② M-Step：最大似然值重估，重新更新 $\boldsymbol{T}$ 矩阵；
   ③ 多次迭代之后，得到全局差异空间矩阵 $\boldsymbol{T}$。
2. i-vector 的计算；
   (1) 计算训练数据库中每个音频所对应的 Baum-Welch 统计量；
   (2) 将已知变量 $\boldsymbol{M}$、$\boldsymbol{T}$ 与 $\boldsymbol{m}$ 代入公式中求出 $\boldsymbol{\omega}$，最后计算 $\boldsymbol{\omega}$ 的后验均值，即 i-vector。

一般情况下 i-vector 的维度在 400-600 之间。该矢量可以代表说话人的身份，具有较强的区分性，而且维度相对较低，可以大幅减少计算量。

其中，Baum-Welch 统计量的计算过程如下：

已知一个 UBM，有 C 个高斯，每个高斯表示如下:

$$\boldsymbol{\lambda}_c = \{\omega_c, \mu_c, \sigma_c^2\},\ \ c = 1,2,\ldots,C \tag{6.13}$$

ω_c为权重，μ_c为均值，σ_c^2为方差;

给定一段T帧语音$\boldsymbol{O} = \boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_T$,其零阶和一阶 Baum-Welch 统计量计算如下:

$$\boldsymbol{N}_c = \sum_{t=1}^{T} \mathrm{P}(c|\boldsymbol{O}_t, \boldsymbol{\lambda}_c) \tag{6.14}$$

$$\boldsymbol{F}_c = \frac{1}{\boldsymbol{N}_c} \sum_{t=1}^{T} \mathrm{P}(c|\boldsymbol{O}_t, \boldsymbol{\lambda}_c)\,(\boldsymbol{O}_t - \boldsymbol{\mu}_c) \tag{6.15}$$

6.2.2.4 PLDA

i-vector 中既包含了说话人信息，也包含了信道信息。因此，其通常依赖于后端区分性模型来实现对说话人因子的“提纯”，进一步提高 i-vector 模型对说话人的区分能力。PLDA(Probabilistic Linear Discriminate Analysis)[45]可以看作是 LDA 的概率形式，最早由 Prince 针对人脸识别问题所提出。

标准的 PLDA 可以认为是有监督版本的联合因子分析，它将总体差异空间中的 i-vector 用两个子空间表示，如下:

$$\boldsymbol{\eta}_{ij} = \boldsymbol{\mu} + \boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{U}\boldsymbol{x}_{ij} + \boldsymbol{\varepsilon}_{ij} \tag{6.16}$$

其中，$\boldsymbol{\eta}_{ij}$代表第i个说话人的第j段语音的 i-vector；$\boldsymbol{\mu}$为所有 i-vector 的全局均值；$\boldsymbol{V}$是说话人空间矩阵(Eigen Voice) ，用于描述说话人的特征；$\boldsymbol{U}$是信道空间矩阵(Eigen Channel) ，用于描述信道的特征；$\boldsymbol{y}_i$与$\boldsymbol{x}_{ij}$是其对应子空间内的因子，服从高斯分布； $\boldsymbol{\varepsilon}$是残差项，服从协方差矩阵为对角阵的高斯分布。

PLDA 训练

对于 PLDA 的训练，根据公式，PLDA 的模型参数一个有 4 个，分别是 i-vector 均值$\boldsymbol{\mu}$，空间特征矩阵 $\boldsymbol{V}$ 和 $\boldsymbol{U}$，$\boldsymbol{\varepsilon}$噪声协方差。由于模型含有隐变量，模型的训练过程采用经典的 EM 算法迭代求解[45]。

PLDA 测试

对于说话人确认任务，每组试验都需要一个目标说话人和一个测试说话人。分别提取目标说话人和测试说话人的 i-vector，使用 PLDA 模型计算它们之间的似然度评分。

假定目标说话人的 i-vector 为$\boldsymbol{\eta}_i$，测试说话人的 i-vector 为$\boldsymbol{\eta}_j$，使用贝叶斯推理中的假设检验理论，计算两个 i-vector 由同一个者隐含变量$\boldsymbol{\beta}$生成的似然程度。H_1为假设$\boldsymbol{\eta}_i$和$\boldsymbol{\eta}_j$来自同一个说话人，两者共享同一个说话人因子隐含变量; H_0为假设$\boldsymbol{\eta}_i$和$\boldsymbol{\eta}_j$来自不同的说话人，它们由不同的说话人因子生成。使用对数似然比(log-likelihood)计算出最后的得分为:

$$\mathrm{score} = \ln \frac{\mathrm{P}(\boldsymbol{\eta}_i, \boldsymbol{\eta}_j|\mathrm{H}_1)}{\mathrm{P}(\boldsymbol{\eta}_i, \boldsymbol{\eta}_j|\mathrm{H}_0)} \tag{6.17}$$

求取得分后，得分高表示来自同一说话人，得分低表示来自不同说话人。

6.2.2.5 总结

通过以上几小节的学习，我们大致了解了说话人识别传统算法的发展流程。首先是使用高斯混合模型拟合说话人模型，但为了克服实际场景中每一个说话人的语音数据少，无法训练出高效的 GMM 模型的问题，研究人员提出了 GMM-UBM 模型，通过预训练一个通用背景模型，只需要使用少量的目标说话人数据，就可以训练出更出色的说话人模型。为了解决 GMM-UBM 模型中的信道干扰问题，JFA 通过分别建模并信道补偿的方法，消除信道的干扰。而 i-vector 模型，解决了 JFA 方法的说话人子空间与信道子空间存在互相掩盖的问题。PLDA 解决了信道失配问题，提纯说话人因子。从 i-vector/PLDA 提出至今，一直都是说话人识别领域的热门模型，即便在当前神经网络在说话人识别领域广泛应用，i-vector 模型也凭借模型的稳定在科研领域与应用领域被广泛使用。

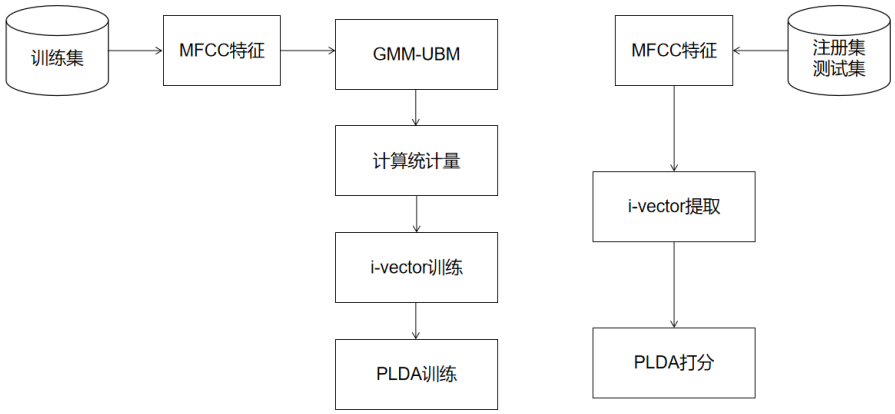


图 6- 5 基于 i-vector 技术的说话人识别系统

图 6- 5 展示了基于 i-vector 的说话人识别系统，在训练过程中，首先需要对训练集提取 MFCC 特征，并使用 MFCC 特征训练 GMM-UBM 模型，接下来计算统计量并训练 i-vector 的 **T** 矩阵，最后进行 i-vector 的提取并对 PLDA 模型进行训练。在测试阶段，首先需要提取注册语音和测试语音的 i-vector，接着使用 PLDA 进行打分，取得结果。

在深度神经网络被应用在语音领域之后，研究者们又提出了 DNN i-vector 模型，如图 6-6。与 GMM i-vector 模型不同的是，DNN i-vector 采用基于深度神经网络训练的语音识别模型替换了基于最大期望(EM) 算法训练的 GMM，以此获得更精确的语言因子，进而预测出更准确的说话人因子。

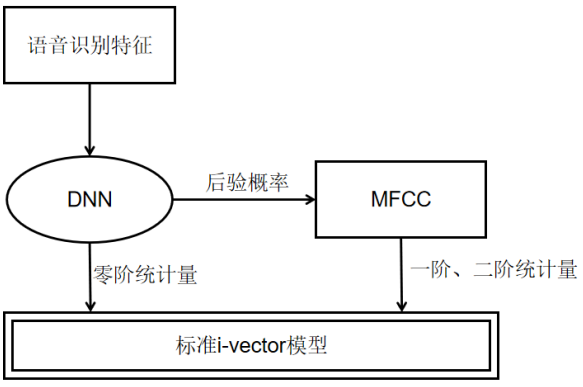


图 6-6 DNN i-vector 混合框架流程图(图引自[23])

1.使用大量的数据训练一个能够将声学特征很好的对应到音素的神经网络,每一帧特征通过神经网络后，就会被分配到某一音素上去。

2.对每一句话在所有的音素进行逐个统计，按照每个音素统计得到相应的信息，得到一个高维特征矢量。

3.使用 i-vector 建模方法对高维特征进行建模:

$$\boldsymbol{M} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{\omega} \tag{6.18}$$

其中 $\boldsymbol{m}$ 是所有训练数据得到的均值超矢量，$\boldsymbol{M}$ 则是每一句话的超矢量，$\boldsymbol{T}$ 通过大量数据训练得到的全变量空间矩阵，$\boldsymbol{\omega}$则是降维后得到的 i-vector。

6.3 基于深度学习的说话人识别算法

6.3.1 深度说话人特征

尽管传统的说话人识别算法在过去的几十年中取得不错的成绩,但是面对复杂多变的语音环境，仍然存在一些难以克服的困难，例如语音时长过短导致包含的信息不足等问题[46]。而深度学习[47]技术因其强大的学习能力得到了极大的发展，也自然扩展到了说话人识别领域。深度说话人系统将原始音频特征作为神经网络的输入，各层神经网络对特征进行处理，随着层数的增加，原始特征中与说话人特征无关的信息(如语音内容、信道信息等)被逐渐减弱、消除。通常网络输入的语音特征为帧级别特征，而输出则为句子级别的说话人嵌入。由于深度说话人特征提取器训练过程中是有监督的，与传统的 i-vector 相比，深度说话人特征的鉴别性更强，并且可以通过调整网络结构及参数等更加方便的提升说话人特征的鉴别性。

合适的网络结构对于提取有效的说话人特征起着至关重要的作用。Variani 等人[6]在 2014 年提出了基于深度神经网络的说话人特征学习，并用于文本相关的说话人识别中。其采用四层全连接层的网络结构对语音特征进行处理,将最后一个隐层的输出进行平均以得到句子级别的特征，称之为 d-vector，如图 6-7 所示。

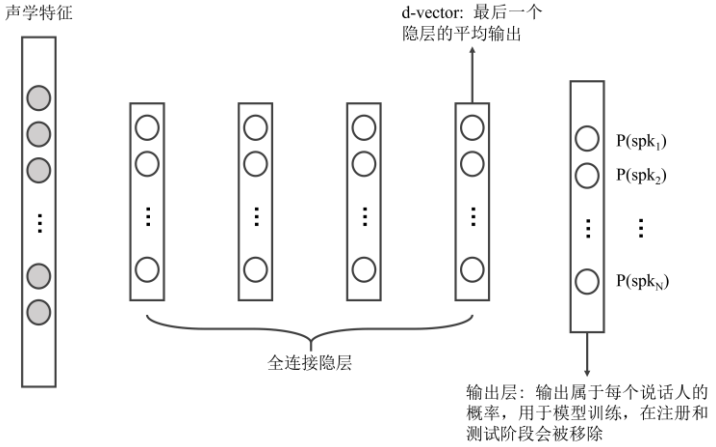


图 6-7 d-vector 结构流程

在 2017 年，David Snyder 提出了一个更加先进的网络来提取说话人嵌入(x-vector)[24]。x-vector 经典训练结构如图 6-8 所示。$\mathbf{X} = \{x_1, x_2, \ldots, x_T\}$为输入的原始语音特征，为了有效

提取语音信号中的动态属性，时延神经网络(TDNN)[48]被应用于提取 x-vector，第一层至第三层均为 TDNN，第四和第五层为全连接层，和 d-vector 中不同的是，该模型通过利用说话人特征中的高阶统计信息以使得说话人嵌入更加稳定，即 x-vector 采用了统计池化层来将帧级别的特征转化为句子级别的特征，具体方式是取帧特征的均值加上标准差。在池化层之后，连接着两个全连接层以生成说话人嵌入，如图 6- 8 所示。经验表明，在第一个嵌入层得到的说话人嵌入(embedding a)，通常会结合 PLDA 进行后端打分；而在第二个嵌入层得到的说话人嵌入(embedding b)往往会结合余弦相似进行后端打分。

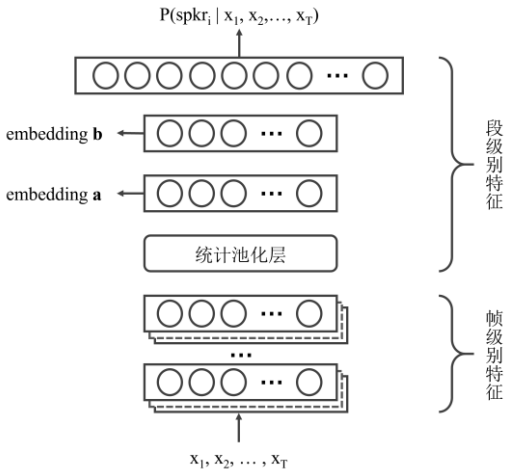


图 6- 8 TDNN 网络结构

6.3.2 后端判别算法

我们通常将提取说话人特征的过程称为前端提取过程，而对说话人进行识别的过程称为后端判别过程，即对提取到的说话人特征进行相似度打分以判断他们是否属于同一个说话人。正如 6.2 节中给出有关 PLDA 的介绍，i-vector 往往会结合 PLDA 进行后端打分，而在基于深度学习的说话人识别系统中，考虑到模型强大的学习能力可以有效的去除说话人特征中的信道信息，经常使用更加简单的余弦相似度(Cosine similarity)作为后端的判别标准。线性鉴别分析(Linear Discriminative Analysis，LDA)作为一种常用降维技术，能够实现最大化类间距离和最小化类内距离，可以在进行后端打分之前对说话人特征进行 LDA 变换降维。

6.3.3 端到端的识别模型

端到端的说话人识别模型旨在通过输入两段语音直接判断他们是否属于同一个人，这与上节中所说的进行前端特征提取并进行后端判别这一系列过程有所不同。在端到端的识别模型中，使用成对训练(Pair-wised training)或者三元组损失(Triplet loss)[49]的策略，即输入的训练数据是二/三元组。模型训练的目标是要使得对于来自同一个说话人的数据元组相似度尽可能的大，而对于来自不同说话人的数据元组相似度尽可能的小。在训练完成后，直接输入两段语音，模型就会给出他们是否属于同一个说话人的判别结果。

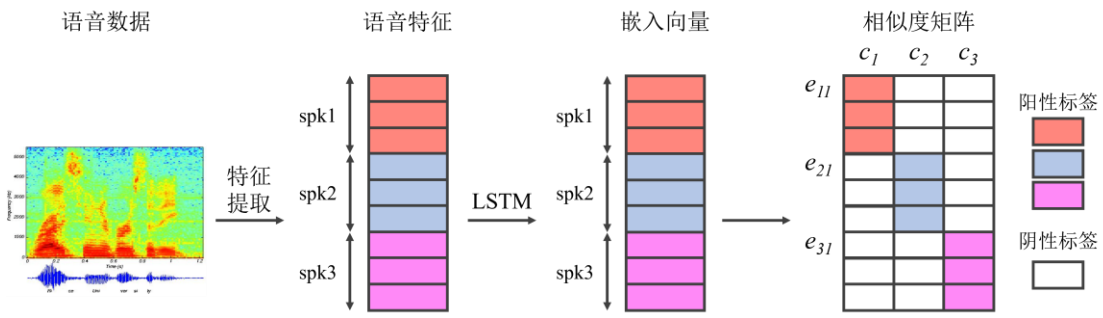


图 6-9 端到端声纹识别模型示例(GE2E[50]架构)

图 6-9 展示了一种经典的端到端声纹识别架构，即 generalized end-to-end(GE2E)。该模型首先分别提取不同说话人的多条语音的嵌入向量。然后，计算不同嵌入向量间的相似度，并进行线性变换，来构造相似度矩阵。在相似度矩阵中，同说话人语音嵌入间的相似度要尽可能大，不同说话人语音嵌入间的相似度要尽可能小。以此为依据可以设计损失函数来优化模型参数。GE2E 的损失可采用如下公式计算。其中，$\boldsymbol{e}_{ji}$表示第 j 个说话人的第 i 条语音$\boldsymbol{x}_{ji}$的嵌入向量，$\boldsymbol{S}_{ji,k}$表示相似度矩阵，$\boldsymbol{c}_k$表示第 k 个说话人的平均嵌入向量，$\boldsymbol{w}$ 和 $\boldsymbol{b}$ 表示线性变换的权重和偏移。

$$\boldsymbol{e}_{ji} = \frac{\boldsymbol{f}(\boldsymbol{x}_{ji};\boldsymbol{\omega})}{||\boldsymbol{f}(\boldsymbol{x}_{ji};\boldsymbol{\omega})||_2} \tag{6.19}$$

$$\boldsymbol{S}_{ji,k} = \boldsymbol{w} \cdot \cos(\boldsymbol{e}_{ji}, \boldsymbol{c}_k) + \boldsymbol{b} \tag{6.20}$$

$$L = \boldsymbol{S}_{ji,k} - \log \sum_{k=1}^{N} \exp(\boldsymbol{S}_{ji,k}) \tag{6.21}$$

6.3.4 迁移学习，多任务学习及多数据库联合学习

在一些广义副语言语音属性识别任务上，始终存在着训练数据少，测试数据与训练数据不匹配，不同语言有不用语言的数据库但不能通用，不同的语音属性(说话人，语种，年龄，性别)识别算法共用同一个数据库，同一个特征甚至同一个方法[51]等情况。目前在传统方法框架内已有一些研究开始关注这一研究方向[[52]-[55]]。对于说话人识别任务，如果训练数据和测试数据来自不用的语种，则使用训练数据语种对应的深度学习语音识别声学模型在测试数据上解码得到的后验概率用于 i-vector 系统并不会改善系统性能[5]。最近在传统方法分类器层面出现了几篇基于迁移学习的跨语种跨数据库的声纹识别和情感识别工作[[52]-[55]]，如图 6-10 所示，但基于统一端到端深度学习框架的包含多个副语言语音属性的迁移学习，多任务及跨语种多数据库学习工作还较少[56]。

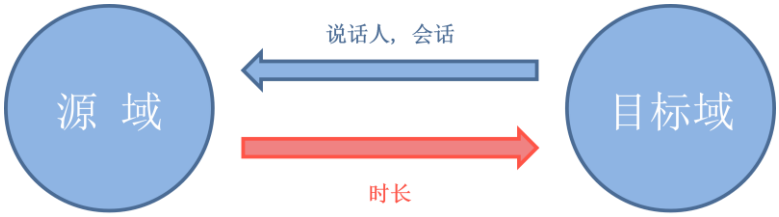


图 6-10 迁移学习在说话人识别中的应用(图引自[53])

6.4 小结与展望

本章描述了说话人识别的概念，并简要地说明了说话人识别的技术优势、应用前景及技术难点。说话人识别通过说话人的声纹特征判断说话人的身份，是一项重要的现代身份认证技术。但是，在现实应用场景下，说话人识别技术面临着远场噪声的干扰、跨设备识别等难点。因此，本章回顾了说话人识别技术的发展历程，全面地介绍了先前的工作中说话人识别系统的实现方案。具体的，本章重点介绍了传统的说话人识别算法以及基于深度学习的说话人识别算法。传统的说话人识别算法采用 GMM-UBM 克服了因语音数据少而无法训练高效模型的问题；采用 JFA 解决了信道干扰的问题；采用 i-vector 解决了说话人子空间与信道子空间相互掩盖的问题；采用 PLDA 解决了信道失配问题。在基于深度学习的说话人识别算法小节，本章详细讲解了 x-vector 说话人识别系统，介绍了使用深度神经网络取得说话人嵌入的过程，并描述了如何使用后端判别算法完成说话人身份的判断。为加深对说话人识别技术的理解，本章还介绍了如何应用 kaldi 工具实现说话人识别算法。目前，随着迁移学习、多任务学习等前沿技术的应用，声纹识别的性能在跨语种等挑战性问题上都有了明显的发展，相信随着技术的不断创新与突破，声纹识别可以为人类带来更加舒适、便捷的生活。以下是关于说话人识别领域今后需要重点关注的一些研究方向：

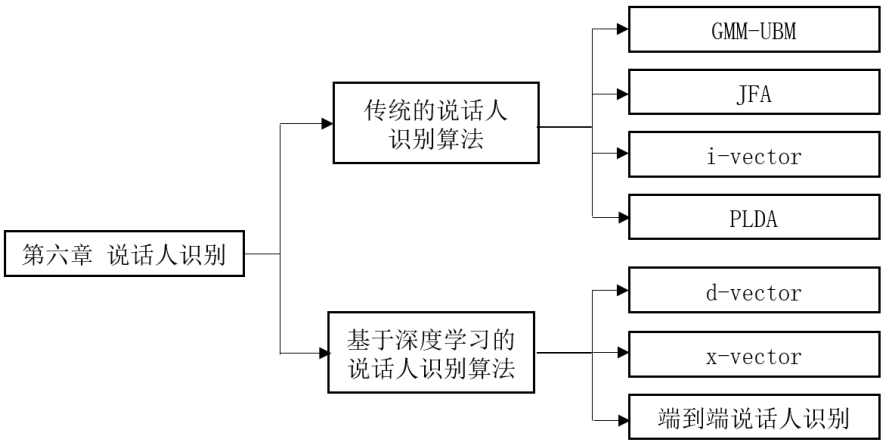
(1)针对复杂场景下的鲁棒性问题，如远场环境，短语音，跨信道，时变鲁棒性等；

(2)新兴的鲁棒性问题，防攻击，戴口罩等；

(3)领域知识的指引与神经网络的结合，可解释性；

(4)多模态身份认证。

本章知识点小结



6.5 说话人识别实践

6.5.1 所需环境

本次说话人识别程序实践主要基于 kaldi 平台，linux 系统。

6.5.2 数据库与评价指标

本节所用数据库为 AISHELL-1[57](数据库下载地址：http://www.openslr.org/33/)。AISHELL-1 录音时长 178 小时，录制过程在安静室内环境中，录制设备为高保真麦克风(44.1kHz，16-bit)。400 名来自中国不同口音区域的发言人参与录制。经过专业语音校对人员转写标注，并通过严格质量检验，此数据库文本正确率在 95%以上。

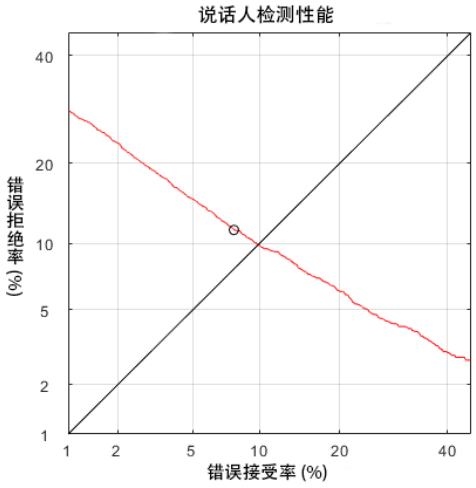


图 6-11 DET(Detection Error Tradeoff)曲线示意图

声纹识别的评价指标一般有 DET(Detection Error Tradeoff)曲线、EER(Equal Error Rate)、

DCF(Detection Cost Function)函数等。DET 曲线示例如图 6-11 所示。该曲线是对二元分类系统误码率的曲线图,绘制出错误拒绝率 FRR(False Reject Rate)与错误接受率 FAR(False Accept Rate)之间随着判断阈值的变化而变化的曲线图。DET 曲线相关计算公式如下所示。其中,n_{rc}表示正样本被判定为负样本的数目,n_{c}表示正样本总数,n_{ai}表示被错误接受的样本数,n_{i}表示负样本总数。

$$FRR = \frac{n_{\mathrm{rc}}}{n_{\mathrm{c}}} \times 100\% \tag{6.22}$$

$$FAR = \frac{n_{\mathrm{ai}}}{n_{\mathrm{i}}} \times 100\% \tag{6.23}$$

EER 是指在 DET 曲线中 FRR 与 FAR 相等的点,即在这个阈值条件下,错误拒绝的样本和错误接受的样本比例一样多,所以取这一点的值作为评价一个说话人验证系统的指标是比较合理的。如图 6-11 所示,两线交点即为 EER。

DCF 是美国国家标准技术局 NIST 在其组织的说话人识别评测中定义的一个检测代价函数,旨在更加有效评价说话人识别系统的性能。图 6-11 中标出的点即为该示例的 DCF。DCF 的计算如下所示。其中,C_{FRR}和C_{FAR}分别表示错误拒绝和错误接收的惩罚代价,P_{target}表示真实说话测试的先验概率。

$$DCF = C_{\mathrm{FRR}} \times FRR \times P_{\mathrm{target}} + C_{\mathrm{FAR}} \times FAR \times (1 - P_{\mathrm{target}}) \tag{6.24}$$

6.5.3 基于 i-vector 的说话人识别

在目录 kaldi/egs/aishell/v1 中即提供了使用 AISHELL-1 作为数据库的 i-vector 说话人识别模型。在 run.sh 文件夹中可以看到从数据下载到模型训练完成的全部过程。

(一) 首先对数据集进行下载:

```
# 该项目的获取路径为 kaldi/egs/aishell/v1
local/download_and_untar.sh $data_url data_aishell
local/download_and_untar.sh $data_url resource_aishell
```

(二) 接着进行数据准备,统计并计算数据信息:

```
# Data Preparation
local/aishell_data_prep.sh $data/data_aishell/wav $data/data_aishell/transcript
```

(三) 对音频数据进行前端段特征 MFCC 提取:

```
# Now make MFCC features.
# mfccdir should be some place with a largish disk where you
# want to store MFCC features.
mfccdir=mfcc
for x in train test; do
  steps/make_mfcc.sh --cmd "$train_cmd" --nj 10 data/$x exp/make_mfcc/$x $mfccdir
  sid/compute_vad_decision.sh --nj 10 --cmd "$train_cmd" data/$x exp/make_mfcc/$x \
    $mfccdir
```

```
    utils/fix_data_dir.sh data/$x
done
```

(四) 训练通用背景模型 UBM:

```
# train diag ubm
sid/train_diag_ubm.sh --nj 10 --cmd "$train_cmd" --num-threads 16 \
    data/train 1024 exp/diag_ubm_1024
# train full ubm
sid/train_full_ubm.sh --nj 10 --cmd "$train_cmd" data/train \
    exp/diag_ubm_1024 exp/full_ubm_1024
```

(五) 使用训练数据对 i-vector 提取器进行训练:

```
# train ivector
sid/train_ivector_extractor.sh --cmd "$train_cmd --mem 10G" \
    --num-iters 5 exp/full_ubm_1024/final.ubm data/train \
    exp/extractor_1024
```

(六) 利用训练好的 i-vector 提取器提取训练数据的 i-vector，为下一步的 PLDA 模型训练做准备:

```
# extract ivector
sid/extract_ivectors.sh --cmd "$train_cmd" --nj 10 \
    exp/extractor_1024 data/train exp/ivector_train_1024
```

(七) i-vector 模型训练:

```
# train plda
$train_cmd exp/ivector_train_1024/log/plda.log \
    ivector-compute-plda ark:data/train/spk2utt \
    'ark:ivector-normalize-length scp:exp/ivector_train_1024/ivector.scp ark:- |' \
    exp/ivector_train_1024/plda
```

(八) 使用 i-vector 提取器提取测试集的 i-vector:

```
# extract enroll ivector
sid/extract_ivectors.sh --cmd "$train_cmd" --nj 10 \
    exp/extractor_1024 data/test/enroll exp/ivector_enroll_1024

# extract eval ivector
sid/extract_ivectors.sh --cmd "$train_cmd" --nj 10 \
    exp/extractor_1024 data/test/eval exp/ivector_eval_1024
```

(九) 对测试数据进行 PLDA 后端打分:

```
# compute plda score
$train_cmd exp/ivector_eval_1024/log/plda_score.log \
    ivector-plda-scoring --num-utts=ark:exp/ivector_enroll_1024/num_utts.ark \
    exp/ivector_train_1024/plda \
    ark:exp/ivector_enroll_1024/spk_ivector.ark \
    "ark:ivector-normalize-length scp:exp/ivector_eval_1024/ivector.scp ark:- |" \
    "cat '$trials' | awk '{print \\\$2, \\\$1}' |" exp/trials_out
```

(十) 等错误率 EER 计算:

```
# compute eer
awk '{print $3}' exp/trials_out | paste - $trials | awk '{print $1, $4}' | compute-eer -
```

6.5.4 基于 x-vector 的说话人识别

在目录 egs/sre16/v2 中提供了 x-vector 模型的训练过程。但在此例中使用的数据集并非 AISHELL-1，但为了保持数据集的统一性，我们仍然可以根据 6.3 节中的数据下载及数据准备过程进行操作，而在与 i-vector 模型相比，x-vector 使用了神经网络提取 x-vector 特征，同时对训练数据进行了加噪等增广操作。在数据增广阶段，首先使用了 RIR 滤波器来模拟混响，以得到模拟混响的训练数据：

```
# 该项目的获取路径为 kaldi/egs/sre16/v2
# Make a reverberated version of the AISHELL list. Note that we don't add any
# additive noise here.
steps/data/reverberate_data_dir.py \
  "${rvb_opts[@]}" \
  --speech-rvb-probability 1 \
  --pointsource-noise-addition-probability 0 \
  --isotropic-noise-addition-probability 0 \
  --num-replications 1 \
  --source-sampling-rate 8000 \
  data/train data/train_reverb
cp data/train/vad.scp data/train_reverb/
utils/copy_data_dir.sh --utt-suffix "-reverb" data/train_reverb data/train_reverb.new
rm -rf data/train_reverb
mv data/train_reverb.new data/train_reverb
```

接着使用 MUSAN 数据集对训练数据进行"noise"、"music"、"babble"三种增广：

```
# Augment with musan_noise
steps/data/augment_data_dir.py --utt-suffix "noise" --fg-interval 1 --fg-snrs "15:10:5:0" \
 --fg-noise-dir "data/musan_noise" data/train data/train_noise
# Augment with musan_music
steps/data/augment_data_dir.py --utt-suffix "music" --bg-snrs "15:10:8:5" \
--num-bg-noises "1" --bg-noise-dir "data/musan_music" data/train data/train_music
# Augment with musan_speech
steps/data/augment_data_dir.py --utt-suffix "babble" --bg-snrs "20:17:15:13" \
--num-bg-noises "3:4:5:6:7" --bg-noise-dir "data/musan_speech" data/train data/train_babble
```

数据增广操作旨在通过增加不同环境下的训练数据使得训练出的模型更加鲁棒，在实践过程中数据增广能够显著提升说话人识别效果。在进行 x-vector 模型训练之前，需要对数据进行一些数据筛选操作，以去除对模型训练无作用甚至副作用的数据：包括剔除静音数据、过短数据、单个说话人包含的语音过少等。x-vector 的训练代码如下：

```
local/nnet3/xvector/run_xvector.sh --stage $stage --train-stage -1 \
  --data data/train_combined_no_sil --nnet-dir $nnet_dir \
  --egs-dir $nnet_dir/egs
```

x-vector 的模型结构如下：

```
# please note that it is important to have input layer with the name=input

# The frame-level layers
```

```
input dim=${feat_dim} name=input
relu-batchnorm-layer name=tdnn1 input=Append(-2,-1,0,1,2) dim=512
relu-batchnorm-layer name=tdnn2 input=Append(-2,0,2) dim=512
relu-batchnorm-layer name=tdnn3 input=Append(-3,0,3) dim=512
relu-batchnorm-layer name=tdnn4 dim=512
relu-batchnorm-layer name=tdnn5 dim=1500

# The stats pooling layer. Layers after this are segment-level.
# In the config below, the first and last argument(0, and ${max_chunk_size})
# means that we pool over an input segment starting at frame 0
# and ending at frame ${max_chunk_size} or earlier. The other arguments(1:1)
# mean that no subsampling is performed.
stats-layer name=stats config=mean+stddev(0:1:1:${max_chunk_size})

# This is where we usually extract the embedding(aka xvector) from.
relu-batchnorm-layer name=tdnn6 dim=512 input=stats

# This is where another layer the embedding could be extracted
# from, but usually the previous one works better.
relu-batchnorm-layer name=tdnn7 dim=512
output-layer name=output include-log-softmax=true dim=${num_targets}
```

此外，在进行后端 PLDA 打分之前，此模型还使用了 LDA 对提取出的 x-vector 进行降维操作:

```
# This script uses LDA to decrease the dimensionality prior to PLDA.
lda_dim=150
$train_cmd exp/xvectors/log/lda.log \
  ivector-compute-lda --total-covariance-factor=0.0 --dim=$lda_dim \
  "ark:ivector-subtract-global-mean scp:exp/xvectors/xvector.scp ark:- |" \
  ark:data/sre_combined/utt2spk exp/xvectors/transform.mat || exit 1;
```

之后再进行的 PLDA 后端打分和 EER 计算过程也与 6.3 节中相似。

6.5.5 常用声纹数据库及工具箱

NIST SRE[58]：是由美国国家标准与技术研究院主办的声纹识别技术评测和多媒体评测，为全球的研究机构提供了一个统一的测试平台。自 1996 年举办至今，NIST SRE 的重点任务是对于现阶段实用领域中，口语对话电话语音(CTS)的说话人检测。除了在各种手机上录制的 CTS 之外，SRE18 中的开发和测试材料还加入了 IP 语音(VOIP)数据，以及视频音频(AfV)数据。数据库环境的复杂程度更高、干扰因素更多，已远远超过一般的实际应用场景，具有很强的鲁棒性。

VoxCeleb[59]：是一个大型的语音识别数据集。它的音频来自 YouTube 视频。这些数据性别分布均衡，名人跨越不同的口音、职业和年龄，训练集和测试集之间没有重叠。(数据库下载链接：https://www.robots.ox.ac.uk/~vgg/data/voxceleb/)

SITW[60]：一个标准数据库，用于测试实际条件下的 ASV 性能。它是从开源媒体渠道

收集的，由 299 位知名人士的语音数据组成。

AISHELL[57]：希尔贝壳中文普通话语音数据库，录音文本涉及唤醒词、语音控制词、智能家居、无人驾驶、工业生产等 12 个领域。1991 名来自中国不同口音区域的发言人参与录制。

MUSAN[61]：MUSAN 数据库，常用语对语音增广的噪声数据库。(数据库下载链接：http://www.openslr.org/17/)

MSR Identity Toolkit[62]：微软开源的工具箱，MATLAB 版本，包含 GMM-UBM 和 I-vector 的基本示例，简单易用。(工具包下载地址如下：https://www.microsoft.com/en-us/download/details.aspx?id=52279)

Alize[63]：主要包括 GMM-UBM、I-vector and JFA 三种传统的方法，C++版，简单易用。(工具包下载地址如下：https://alize.univ-avignon.fr/)

Kaldi[64]：当下十分流行的语音识别工具包，也包括声纹识别：覆盖了主流的声纹识别算法(I-vector 、x-vector 等)。(工具包下载地址如下：https://kaldi-asr.org/)

Sidekit[65]：Python 工具包。(下载地址如下：https://projets-lium.univ-lemans.fr/sidekit/)

参考文献

[1] Tomi Kinnunen, Haizhou Li: An overview of text-independent speaker recognition: From features to supervectors. Speech Commun. 52(1): 12-40(2010).

[2] Schuller, Björn, et al. "The INTERSPEECH 2010 paralinguistic challenge." Eleventh Annual Conference of the International Speech Communication Association. 2010.

[3] Schuller B W. The computational paralinguistics challenge [social sciences][J]. IEEE Signal Processing Magazine, 2012, 29(4): 97-101.

[4] Dehak, Najim, et al. "Front-end factor analysis for speaker verification." IEEE Transactions on Audio, Speech, and Language Processing 19.4(2010): 788-798.

[5] 汤志远，李蓝天，王东，蔡云麒，石颖，郑方. 语音识别基本法[M]. 电子工业出版社. 2020:1-125.

[6] Variani E, Lei X, McDermott E, et al. Deep neural networks for small footprint text-dependent speaker verification[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2014: 4052-4056.

[7] Heigold G, Moreno I, Bengio S, et al. End-to-end text-dependent speaker verification[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2016: 5115-5119.

[8] Zhang S X, Chen Z, Zhao Y, et al. End-to-end attention based text-dependent speaker verification[C]//2016 IEEE Spoken Language Technology Workshop(SLT). IEEE, 2016: 171-178.

[9] Yu C, Zhang C, Kelly F, et al. Text-Available Speaker Recognition System for Forensic Applications[C]//INTERSPEECH. 2016: 1844-1847.

[10] Anand A, Labati R D, Hanmandlu M, et al. Text-independent speaker recognition for Ambient Intelligence applications by using information set features[C]//2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications(CIVEMSA). IEEE, 2017: 30-35.

[11] 杨琴. 与文本无关的说话人识别技术研究[D]. 电子科技大学, 2020.

[12] Snyder D, Garcia-Romero D, Povey D, et al. Deep Neural Network Embeddings for Text-Independent Speaker Verification[C]// Interspeech 2017. 2017.

[13] 王梦然. 声纹识别渐行渐近[J]. 发明与创新: 大科技, 2020(3): 34-35.

[14] 刘红星，刘山葆. 声纹识别和意图理解技术在电信诈骗检测中的应用研究[J]. 广东通信技术, 2020, 40(7): 33-39.

[15] 刘乐，陈伟，张济国，等. 声纹识别: 一种无需接触，不惧遮挡的身份认证方式[J]. 中国安全防范技术与应用, 2020(1): 32-40.

[16] 卢一男，单宝钰，关超. 声纹识别技术现状与发展应用[J]. 信息系统工程, 2017(2): 11-11.

[17] Berndt D J, Clifford J. Using dynamic time warping to find patterns in time series[C]//KDD workshop. 1994, 10(16): 359-370.

[18] Gray R. Vector quantization[J]. IEEE Assp Magazine, 1984, 1(2): 4-29.

[19] Krogh A, Larsson B, Von Heijne G, et al. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes[J]. Journal of molecular biology, 2001, 305(3): 567-580.

[20] Li, Ming, et al. "Speaker verification based on the fusion of speech acoustics and inverted articulatory signals." Computer speech & language 36(2016): 196-211.

[21] Guo, Jinxi, et al. "Speaker Verification Using Short Utterances with DNN-Based Estimation of Subglottal Acoustic Features." INTERSPEECH. 2016.

[22] Zeinali, Hossein, et al. "i-Vector/HMM Based Text-Dependent Speaker Verification System for RedDots Challenge." InterSpeech. 2016.

[23] Yun L, Scheffer N, Ferrer L, et al. A novel scheme for speaker recognition using a phonetically-aware deep neural network[C]// ICASSP 2014 - 2014 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2014.

[24] Snyder D, Garcia-Romero D, Sell G, et al. X-vectors: Robust dnn embeddings for speaker recognition[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2018: 5329-5333.

[25] Cai W, Chen J, Ming L. Exploring the Encoding Layer and Loss Function in End-to-End Speaker and Language Recognition System[C]// Odyssey 2018. 2018.

[26] Dehak N, Kenny P J, Dehak R, et al. Front-End Factor Analysis for Speaker Verification[J]. IEEE Transactions on Audio Speech & Language Processing, 2011, 19(4):788-798.

[27] Stafylakis T, Kenny P, Ouellet P, et al. Text-dependent speaker recognition using PLDA with uncertainty propagation[C]// Interspeech. 2013.

[28] https://www.nist.gov/itl/iad/mig/speaker-recognition.

[29] https://www.nist.gov/itl/iad/mig/language-recognition.

[30] https://www.robots.ox.ac.uk/~vgg/data/voxceleb.

[31] https://www.asvspoof.org.

[32] Kinnunen, Tomi, Kong-Aik Lee, and Haizhou Li. "Dimension reduction of the modulation spectrogram for speaker verification." Odyssey. 2008.

[33] Misra, Songhita, et al. "Comparison of MFCC and LPCC for a fixed phrase speaker verification system, time complexity and failure analysis." 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]. IEEE, 2015.

[34] Ganchev, Todor, Nikos Fakotakis, and George Kokkinakis. "Comparative evaluation of various MFCC implementations on the speaker verification task." Proceedings of the SPECOM. Vol. 1. No. 2005. 2005.

[35] Alam, Md Jahangir, et al. "Multitaper MFCC and PLP features for speaker verification using i-vectors." Speech communication 55.2(2013): 237-251.

[36] Todisco, M., Delgado, H., & Evans, N. W.(2016, June). A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients. In Odyssey(Vol. 2016, pp. 283-290).

[37] Wang, Longbiao, et al. "Relative phase information for detecting human speech and spoofed speech." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[38] Reynolds, Douglas A., Thomas F. Quatieri, and Robert B. Dunn. "Speaker verification using adapted Gaussian mixture models." Digital signal processing 10.1-3(2000): 19-41.

[39] Stolcke, Andreas, et al. "MLLR transforms as features in speaker recognition." Ninth European Conference on Speech Communication and Technology. 2005.

[40] Campbell, William M., et al. "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation." 2006 IEEE International conference on acoustics speech and signal processing proceedings. Vol. 1. IEEE, 2006.

[41] Reynolds D A, Quatieri T F, Dunn R B. Speaker verification using adapted Gaussian mixture models[J]. Digital signal processing, 2000, 10(1-3): 19-41.

[42] Kenny, Patrick, et al. "Joint factor analysis versus eigenchannels in speaker recognition." IEEE Transactions on Audio, Speech, and Language Processing 15.4(2007): 1435-1447.

[43] CSDN，声纹识别之 I-Vector. 2018.

[44] Senoussaoui M, Kenny P, Dehak N, et al. An i-vector Extractor Suitable for Speaker Recognition with both Microphone and Telephone Speech[J]. Odyssey, 2010.

[45] Ioffe, Sergey. "Probabilistic linear discriminant analysis." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2006.

[46] Kanagasundaram A, Vogt R, Dean D B, et al. I-vector based speaker recognition on short utterances[C]//Proceedings of the 12th Annual Conference of the International Speech Communication Association. International Speech Communication Association(ISCA), 2011: 2341-2344.

[47] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.

[48] Waibel A, Hanazawa T, Hinton G, et al. Phoneme recognition using time-delay neural networks[J]. IEEE transactions on acoustics, speech, and signal processing, 1989, 37(3): 328-339.

[49] Hermans A, Beyer L, Leibe B. In defense of the triplet loss for person re-identification[J]. arXiv preprint arXiv:1703.07737, 2017.

[50] Wan L, Wang Q, Papir A, et al. Generalized end-to-end loss for speaker verification[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2018: 4879-4883.

[51] Shivakumar P G, Li M, Dhandhania V, et al. Simplified and supervised i-vector modeling for speaker age regression[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2014: 4833-4837.

[52] Sholokhov A, Kinnunen T, Cumani S. Discriminative multi-domain PLDA for speaker verification[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2016: 5030-5034.

[53] Hong Q, Li L, Wan L, et al. Transfer Learning for Speaker Verification on Short Utterances[C]//Interspeech. 2016: 1848-1852.

[54] Aronowitz H. Compensating Inter-Dataset Variability in PLDA Hyper-Parameters for Robust Speaker Recognition[C]//Odyssey. 2014: 280-286.

[55] Wen Y, Liu W, Yang M, et al. Efficient misalignment-robust face recognition via locality-constrained representation[C]//ICIP. 2016: 3021-3025.

[56] Novotný O, Matějka P, Glembek O, et al. Analysis of the dnn-based sre systems in multi-language conditions[C]//2016 IEEE Spoken Language Technology Workshop(SLT). IEEE, 2016: 199-204.

[57] Bu H, Du J, Na X, et al. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline[C]//2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment(O-COCOSDA). IEEE, 2017: 1-5.

[58] Przybocki M A, Martin A F, Le A N. NIST speaker recognition evaluation utilizing the Mixer corpora—2004, 2005, 2006[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(7): 1951-1959.

[59] Nagrani A, Chung J S, Zisserman A. Voxceleb: a large-scale speaker identification dataset[J]. arXiv preprint arXiv:1706.08612, 2017.

[60] McLaren M, Ferrer L, Castan D, et al. The Speakers in the Wild(SITW) speaker recognition database[C]//Interspeech. 2016: 818-822.

[61] Snyder D, Chen G, Povey D. Musan: A music, speech, and noise corpus[J]. arXiv preprint arXiv:1510.08484, 2015.

[62] Sadjadi S O, Slaney M, Heck L. MSR identity toolbox v1. 0: A MATLAB toolbox for speaker-recognition research[J]. Speech and Language Processing Technical Committee Newsletter, 2013, 1(4): 1-32.

[63] Bonastre J F, Wils F, Meignier S. ALIZE, a free toolkit for speaker recognition[C]//Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. IEEE, 2005, 1: I/737-I/740 Vol. 1.

[64] Povey D, Ghoshal A, Boulianne G, et al. The Kaldi speech recognition toolkit[C]//IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011(CONF).

[65] Larcher A, Lee K A, Meignier S. An extensible speaker identification sidekit in python[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP). IEEE, 2016: 5095-5099.