摘 要

说话人识别(Speaker Recognition)是指通过语音自动鉴别说话人身份的技术。它是目前流行的生物认证技术之一，具有采集简单、实时识别、无接触、普适等优势。说话人辨识(Speaker Identification)是说话人识别的主要任务之一，即从一个说话人集合中辨识出当前说话人的身份。本文研究了基于高斯混合模型(Gaussian Mixture Model，GMM)的说话人辨识技术，在 Alize/LIA_RAL 工具包的基础上设计与实现了一个基于说话人辨识的声纹考勤系统。本文的主要工作和成果如下：

1. 实现了 Windows 环境下的音频控制。利用 WaveX APIs 实现了麦克风录音、声音回放、存储等功能。为后期声音数据采集和说话人辨识系统提供统一的音频控制接口。
2. 设计了实验语料库，并对采集到的语音数据进行切分和分类处理，为说话人识别方法的实验验证提供数据。
3. 基于 Alize/LIA_RAL 工具包，设计与实现一个基于 GMM 模型的说话人辨识模块。
4. 采用 CSLU 语料库和自建语料库对说话人辨识模块进行了实验测试。研究了高斯混合阶数、训练次数、训练数据量、模仿音等因素对识别性能的影响。
5. 使用 VC++实现了一个基于说话人辨识模块的声纹考勤系统。该系统的主要特点和优势有：采用语音录入进行人员考勤，具有便捷、响应速度快、识别率高、考勤信息可靠性高等优点；系统采用 C/S 架构的思想，客户端是用户操作区，简单方便，界面友好，“服务器”端完成耗时的模型加载和判分决策的任务；具有新用户注册与模型训练、声纹考勤登记、考勤记录查询等功能。

关键词：说话人识别，声纹识别，说话人辨识，高斯混合模型，声纹考勤

ABSTRACT

Speaker recognition (or voiceprint recognition) is the technology that recognizes the speaker's identity by the speech. It is one of the popular biometrics with a lot of advantages, such as easy speech data collection, real-time recognition and ubiquitous access. Speaker Identification is one of the major tasks of speaker recognition, which determines who is talking from a set of known voices. This thesis studies Gaussian mixture model (GMM) based speaker identification technoliges. We develop a voiced-based attendance checking system using the Alize/LIA-RAL Library. The main contributions of the thesis are summarized as follows:

Firstly, we develop an audio control interface on Windows platform, which can record, play and save sounds. It provides a unified audio interface for audio data collection speaker identification.

Then, we collect a speech corpus for speaker modeling and speaker identification evaluation.

Thirdly, we build a command-line speaker identification module based on the Alize/LIA_RAL library. We also study the influences of some factors to the speaker idenfitication system, which involve the number of the Gaussian mixtures, training times, the amount of the training data and mimic sound.

Lastly, we design and develop a voiced-based attendance checking system. The system has been used in our lab with a fast response, a low error rate with an IDER below 0.05 in the quiet recoding room and higher reliability of the attendance registration.

Keywords: speaker recognition, voiceprint recognition, speaker identification, Gaussian mixture model (GMM), voiced-based attendance checking system

目 录

第一章　绪论

1.1 课题研究的背景与意义

本文来源于“教育部新世纪优秀人才支持计划”。本次毕设的主要工作任务为：研究基于高斯混合模型(GMM)的说话人识别技术，利用 Alize 开源工具包，设计与实现一个完整的基于 GMM 的说话人辨识模块，搭建基于声纹识别的人员考勤系统。该系统的主要功能包括：录音与回放等音频控制、用户注册、用户模型训练、说话人识别与拒识、简单易用的使用界面等。

一千多年前阿里巴巴“芝麻开门”的故事，透露出人类想利用声音来实现自己要求的理想，这在当时只能算是一个美好的梦想。随着说话人识别技术的日趋成熟，今天足以让“芝麻开门”的梦想成为现实，说话人识别技术就是我们在智能轻松的数字化生活中的阿里巴巴魔咒！

人类语音的产生是人体语言中枢与发音器官之间的一个复杂的生理物理过程，人在讲话时使用的发音器官——舌、牙齿、喉头、肺、鼻腔等在尺寸和形态方面，每个人的差异很大，所以任何时候两个人的声音图谱都有差异。每个人的语音声学特征既有相对稳定性，又有变异性，不是绝对的的、一成不变的。这种变异来自生理、病理、心理、模拟、伪装，也与环境干扰有关。尽管如此，由于每个人的发音器官和发音习惯都不尽相同，因此在一般情况下，人们仍能区别不同的人的声音或判断是否是同一人的声音。

语音里面包含了丰富的信息，包括性别、情感、内容、身份、健康、语种等。我们可以很容易的根据这些特征来区分出不同的人。尽管是相同的语音内容，一个 90 岁的老太太说话，一听就知道这是个老年妇女的声音，一个很疲惫的人说话，一听便知道他很累。这些年龄、性别、身体状态等附加信息，对于传达语音内容是没有用的，但是却可以用来估计语音内容的可靠性、关联性以及一些其它的特性[1]。当我们将耳朵和大脑的分析任务交给麦克风和计算机的时候，语音中的这些特性就被分开了。由于计算机速度的不断提高，语音识别和、话人识别、情感识别、语种识别等方向的研究得到了极大的关注。其中语音识别和说话人识别是最热门的两个领域。语音识别(Speech Recognition)，是判断出说话人语音内容的技术，这种识别应该，也必须忽略方言、疾病以及年龄等因素的影响。如果

两个人语音中含有相同的字词，计算机应该能得到同样的结果。而说话人识别(Speaker Recognition)，是辨别出谁在说话的技术，语音的内容不重要，是可以被忽略的特征。但是，和语音识别一样，疾病、年龄以及短暂的环境噪音将会对计算机的识别结果造成很大的影响，基于这样原因，系统必须要有较好的鲁棒性。

说话人识别(Speaker Recognition)，又称声纹识别(Voiceprint Recognition)，是生物特征识别(Biometrics)技术中的一种。按照识别的任务划分，主要可分为说话人辨识(Speaker Identification)任务和说话人确认(Speaker Verification)任务。前者用来确认某段语音是若干人中的哪个所说的，是“多选一”的问题；而后者用以确认某段语音是否是指定的某个人所说的，是“一对一判别”的问题。不同的任务和应用会使用不同的说话人识别技术，如缩小刑侦范围时可能需要辨识技术，而银行交易时则需要确认技术。

与其他生物识别技术，诸如指纹、掌纹、虹膜等相比较，说话人识别除了具有不会遗失和忘记、不须记忆、使用方便等优点外，还具有以下的特性：

(1) 用户接受程度高，由于不涉及隐私问题，用户无任何心理障碍。

(2) 利用语音进行身份识别可能是最自然和最经济的方法之一。声音输入设备低廉，甚至无费用(如电话、手机)，而其他生物识别技术的输入设备往往造价昂贵。

(3) 在基于电信网络的身份识别应用中，如电话银行、电话炒股、电子购物等，与其他生物识别技术相比，说话人识别更为擅长与自然，得天独厚。

声纹识别独特的优点使该项技术具有广泛的应用前景，近年来越来越受到人们的重视[2]。目前的主要应用领域如表 1-1 所示。

据 IBG(International Biometric Group，国际生物识别集团)发布的 2007-2012 年全球生物识别市场报告预测，到 2012 年全球生物识别市场的规模将超过 74 亿美元。比尔·盖茨认为：以人类生物特征(指纹、语音、脸像等)进行身份验证的生物识别技术，在今后数年内将成为 IT 产业最为重要的技术革命。在世界生物识别技术市场上，从具体产品结构上来看，指纹识别占据生物技术门类的主导地位，掌形识别居于第二位，其中指纹识别占 52.1%，掌形识别占 30%，虹膜识别占 7.3%，说话人识别占 4.5%，笔迹识别占 2.4%，其它占 3.7%[4]。虽然说话人识别仅排名第四，但是随着计算机运算速度的提高、电信网络的发展、说话人识别技术的不断成熟，说话人识别技术会以它特有的优势，在世界范围内广泛应用于诸多领域。目前，我国说话人识别市场尚属启动阶段，其发展空间更为广阔，在金融、证券、社保、公安、军队及其他民用安全认证等行业和部门有着广泛的需求。

表 1-1 说话人识别的应用领域举例

应用领域	应用举例
信息领域	自动总机系统中，识别出主叫方的身份。
银行、证券	应用说话人确认系统到电话银行、远程炒股、信用卡交易等业务的用户身份确认，与传统密码一起形成双保险。
访问控制	如机场和小区的门禁系统，也可用于门和车的钥匙卡，远程访问计算机网络等。
军队和国防安全	作战中监听是否有关键人说话出现，对军事指令的发令者进行身份确认，如 2001 年中美撞机事件中的美侦察机就有说话人识别模块。
公安司法办案	主要用于一些勒索事件中，可以缩小侦察范围；在法庭上可以利用说话人身份确认技术。
多媒体信息管理	对含有多个说话人的音频文件进行分类归档、索引、语音邮件浏览等。
当代个性化服务	用语音操控的智能家庭系统。
与二维条形码技术相结合的防伪应用	在 PDF417 二维条形码中加入声音信息进行编码，声纹二维条形码可以应用生活的很多领域，如物流配送。

1.2 目前国内外研究进展和发展趋势

1.2.1 研究进展和应用

说话人识别的研究始于 20 世纪 30 年代，早期的工作主要集中在用人耳进行听辨语音的实验和探讨听音识别的可能性方面。随着电子技术和计算机技术的发展，通过机器自动识别人的语音称为可能。Bell 实验室的 Pruzansky 提出了一种基于模式匹配和概率统计方差分析的说话人识别方法[3]，从而引起信号处理领域许多学者的注意，形成了说话人识别领域的一个热潮。这期间主要工作集中在各种识别参数的提取、选择和实验上。20 世纪 70 年代至今，说话人识别的研究重点转向对各种声学特征参数的线性或者非线性处理以及新的模式匹配方法上。如今，说话人识别已经逐渐走向世纪应用，AT&T 应用说话人识别技术制造出的智慧卡(Smart Card)，已经应用于自动提款机上。欧洲电信联盟于 1998 年完成了 CAVE(Caller Verification in Banking and Telecommunication)计划，并于同年启动

了 PICASSO Pioneering Call Authentication for Secure Service Operation)计划，在电信网上完成了说话人识别。其他一些商用系统还包括：ITT 公司的 SpeakerKey、Keyware 公司的 VoiceGuardian、T-NETIX 公司的 SpeakEZ 等。此外，国内许多高科技公司， 如中科模识科技公司、 中科信利技术有限公司等，也都专门开发了许多说话人识别方面的应用产品；另外，还有厦门天聪智能软件有限公司所研发的智能声纹识别系统，集成了国际先进的说话人模型技术，能够依据较短的语音，自动鉴别说话人身份，目前用于中国科技馆人类声纹展项，体现了声纹识别最高科技成果；得意音通(d-Ear)公司开发的“得意”身份证将声纹辨认与“得意”关键词检出器结合起来，可以在自动总机系统中使用。

目前国际上许多著名大学、 研究机构以及很多大公司的实验室都在进行说话人识别方面的研究，如麻省理工学院林肯实验室(Lincoln Laboratory) 、美国的 ICSI(International Computer Science Institute) 、美国的 SRI 公司的语音技术与研究实验室(STAR) 、法国的 LIA(Laboratoire Informatique Avigonon) 、加拿大的 CRIM(Centre de recherche informatique de Montréal)实验室等。 在国内，许多大学和研究机构也在这一领域开展了大量的研究工作，并在说话人识别方面取得了丰硕的研究成果，如中科院声学所、中科院自动化研究所、北京大学、中国科技大学、北京邮电大学、北京交通大学、北京理工大学、上海交通大学、哈尔滨工业大学、西北工业大学等。

说话人识别技术也开始逐渐应用于一些产品上。2004 年开始，得意音通公司的声纹识别技术就被公安部分认可和采用。目前国内市场上利用声纹识别技术做出产品主要有：上海优浪科技有限公司的“优浪声纹识别门禁 LYM-01”，安徽科大讯飞信息科技股份有限公司的“interVeri 系列”专业声纹识别软件，北京得意音通公司做出的“得意声纹识别 VPR4.0”，厦门天聪智能软件有限公司的“智能声纹识别系统”等。其中，科大讯飞语音实验室在 2008 年国际说话人识别评测大赛上荣获综合指标第一名。2009 年，得意音通公司中标中国建设银行电话银行 95533 交易整合及业务管理项目，在该项目中，语音识别技术将用于呼叫中心的自动导航，而声纹识别技术将用于电话银行的用户身份确认。这说明，随着说话人识别技术的不断提高，其应用将离我们越来越近。

1.2.2 研究热点与发展趋势

国际上有一些定期的说话人识别评测的活动，如从 1996 年开始，为了评估说话人识别的研究水平，NIST(National Institute of Standards and Technology，美国国家标准及技术署)开始举办一年一度的世界性的说话人评测(Speaker Recognition Evaluation，简称 SRE)，NIST 说话人识别评测在说话人识别领域被

认为是与文本无关的说话人确认领域的最高水平。每年 NIST 会公布 SRE 测评计划，参赛者报名参加。NIST 为所有的参赛者提供统一的数据、评测条件和结果评估方法，因此各个参加者的说话人识别系统之间可以进行比较。NIST 的评测标准是不断提升的，如今基于文本有关的、干净环境下的说话人识别的性能已经很高，因而 NIST 的测评重点转移到传输通道环境下的与文本无关的说话人检测技术。语料库通常为 SwitchBoard 或 SwitchBoardIII的子集。评测以识别等错误率(EER)、最小检测代价(minDCF)、检测代价(DCF)为三大核心测试指标。目前 NIST 2010 年的 SRE 的评测计划已经发布。

在 NIST 的 SRE 测评中，出现了很多优秀的说话人识别技术，使得说话人识别技术在近几年得到了大幅度的提高，如 GMM-UBM 结构、说话人模型合成(SMS)、以及针对电话话筒失真的评分规整技术 HNORM[6] 等。

现在用于说话人识别技术的研究热点与发展趋势有：

(1) **特征提取方面** 说话人个性特征的差异可以从生理和行为两个方面来体现，特征的选取和前端、后端处理是一个很重要的环节。通常使用的 MFCC、PLP 以及 LPCC 特征体现的是比较低层的特征，也是目前做说话人识别使用的主流特征。基于高层特征(如说话人用语习惯、发音特点)主要作为前者的补充与前者的识别结果进行融合。据 Reynolds[7]的研究表明，在说话人识别任务中，MFCC 特征比 LPCC 和 PLP 具有更加优越的识别性能，因此大多数系统使用的都是 MFCC 特征。目前的研究主要有 LPCC 特征参数与其他特征参数结合，在 MFCC 特征参数的基础上引入一些相位参数和高层特征参数等，使用新的特征和设计新的特征提取简化算法等。这些特征参数在一定程度上提高了系统的性能，今后进一步的研究方向是：一方面寻找新的更有效的特征参数，另一方面是在已有特征参数的基础上，将多个参数结合起来，分析权重，达到更好的识别效果。

(2) **建模方面** 传统的基于 GMM 的模型属于生成性模型，适用于语音数据建模；支持向量机(SVM)则是属于区分性模型，适用于数据分类。2005 年由 Lincoln 实验室的 Campbell 等人将 SVM 思想引入到说话人识别领域，取得不错的效果[8]，将高斯均值超向量作为 SVM 系统的输入构成 GSV-SVM(Gaussian Mean Supervecotrs - Support Vector Machine)系统，在跨信道性能方面取得了很大提高，甚至超过 GMM-UBM 系统。目前，GSV-SVM 系统已是文本无关说话人识别系统的主流系统之一。另外，一些匹配算法的研究也达到了提高识别效率或加快识别速度的目的。如使用 PCA 变换矩阵、在 GMM 模型中引入 MCE 准则、在 SVM 模型中引入最小二乘准则(Least Square，LS)等。

(3) **判分识别方面** 判分识别时说话人识别系统的后端数据处理部分的作用

也是非常重要。目前的一些好的判分处理算法对系统的性能提高取得一定的效果，尤其是在信道失配方面较显著。常用的判分规整技术主要有 ZNorm、TNorm、HNorm、HTNorm、DNorm、ATNorm 等。现在又有一些新的判分规整算法，如多个系统分数融合算法、分数非监督自适应算法等。另外，现在对于信道失配的研究热点，还有潜伏因子分析(latent factor analysis，LFA)、基于 GMM-supervector 进行分析、干扰属性投影(nuisance attribute projection， NAP)、以及在前两者的基础上提出的基于信道子空间投影(channel-based subspace project，CSP)等，这些模型补偿的算法都取得了不错的效果。

说话人识别技术发展至今，尽管已经取得了不错的进展，要寻找更加优良的研究方法仍然有相当艰巨的路要走。由于技术条件的限制，目前所采用的抽样建模方法还存在着不足。对说话人识别最有影响的因素是在不同实验室中声音特性信号的变更，包括说话人生理上的变动性以及实验条件的不稳定性等，这些都对说话人识别系统构成严峻的挑战。此外，说话人识别技术还应解决提取声音长期稳定的特征参数的问题。在多个人进行交谈时，自动实时的从中提取每个人的声音特性并加以区分的技术是值得研究的方向。

1.3 本文的工作

本文研究基于高斯混合模型(GMM)的说话人识别技术，利用 Alize 开源工具包，设计与实现一个完整的基于 GMM 的说话人辨识模块，搭建基于声纹识别的人员考勤系统。该系统的主要功能包括：录音与回放等音频控制、用户注册、用户模型训练、说话人识别与拒识、简单易用的使用界面等。

本文的工作与成果主要有：

(1) 实现了 Windows 环境下的音频控制

利用 Windows 提供的 Waveform Audio APIs 以及 Multimedia File I/O APIs 实现一个 Windows 环境下的具有麦克风录音、声音回放、数据存储等功能的音频控制模块。该模块有相应的波形显示区，以提示用户是否有数据被采集。该模块为后期声音数据采集和说话人辨识系统提供统一的音频控制接口。

(2) 设计了实验语料库，并对语音数据进行切割和分类处理

根据本文实验的实际需求，采集了一批语音数据，组成自己的实验语料库，并对采集到的连续语音数句进行切割和分类处理，为说话人识别方法的实验验证部分提供数据。

(3) 设计与实现了一个基于 GMM 的说话人识别模型

基于 Alize/LIA_RAL 工具包[9]，进行 GMM 模型训练、说话人识别模型练、对数似然计算、识别结果统计等模块程序的编写。本文分 GMM 和 GMM-UBM 两种方案对说话人进行建模。

(4) 实验测试

采用 CSLU 语料库和自建语料库对说话人辨识模块进行了实验测试。通过一系列的实验研究了高斯混合阶数、训练次数、训练数据量、模仿音等因素都对识别性能的影响。根据实验结果，选取一组较优的参数组合应用到声纹考勤系统中。

(5) 声纹考勤系统的设计与实现

使用 VC++实现了一个基于说话人辨识模块的声纹考勤系统。该系统主要是采用语音录入的方式对系统已经建模的用户进行考勤。系统具有新用户注册于模型训练、声纹考勤登记、考勤记录查询等功能。采用 C/S 架构的思想，客户端作为用户操作区，简单方便、界面友好；“服务器”端完成耗时的模型加载和判分识别的任务。

(6) 声纹考勤系统的运行和维护

系统完成之后，让教研室的所有成员试用该系统，工作日进行考勤。所有使用该系统的用户对系统进行多方面的评估，最终得出本系统的综合性能测试结果。

1.4 本文的组织结构安排

本文共分六章，具体安排如下：

第一章即本章是绪论部分。首先介绍了本课题的来源，研究背景和意义；然后，分析了国内外研究的进展情况，介绍了国内一些利用说话人识别技术做成的产品；阐述了本文的工作和应用创新点；最后介绍了本文的组织结构安排。

第二章是基于 GMM 的说话人识别原理。简述说话人识别方法和系统结构，介绍本文所使用的音频特征，分阶段介绍本文说话人识别的技术。最后给出系统流程图。本章将重点讨论 GMM 和 GMM-UBM 算法的思想和应用原理。

第三章是说话人辨识模块的设计与实现。首先介绍本文所使用的 Alize/LIA_RAL 软件包，包括 Alize 和 LIA_RAL 的关系、开发背景、开发框架和使用方法等，还介绍了如何使用这两个软件包完成本文的说话人辨识模块的功能。最后，介绍了利用进程间的通信和同步技术来解决实时说话人识别的问题。

第四章是声纹考勤系统的设计与实现部分。这一章主要介绍 Windows 下的音频控制技术、声纹考勤系统的架构、各个模块的功能、以及用户界面设计等。

另外，对系统的使用给出了要求。

第五章是实验验证和系统评估。介绍本次毕设所涉及到的几个语料库，并给出自己采集的语料库的概况。在 CSLU 语料库上进行基本测试，然后在自建语料库上进行具体测试，并对实验的结果进行分析。最后给出声纹识别系统的整体评估。

第六章是总结与展望。总结了本次毕业设计的主要工作，同时提出了不足和改进方式，对未来的工作进行了展望。

第二章 基于 GMM 的说话人识别原理

2.1 说话人识别基本原理

2.1.1 说话人识别任务分类

说话人识别是从说话人的语音中提取说话人个性特征，并根据个性特征识别出说话人身份的技术。说话人识别任务根据识别方式的不同可以分为三类[10,11](如图 2-1 所示)。

1) 说话人辨识(Speaker identification，也称说话人鉴别)，是指从指定用户集中把测试语音所属的说话人区分出来，是一个“多选一”的问题。

2) 说话人确认(Speaker Verification，也称说话人检测)，主要是通过用户测试语音来判断其是否是所声称的用户身份，是一个“一对一”判别的问题。

3) 说话人切分与聚类(Speaker Segmentation and Clustering，也称说话人探测跟踪)，主要指对一段包含多个说话人的语音，正确的切分出这段语音中说话人转换的时刻，并对每个切分出的说话人进行身份的标注。

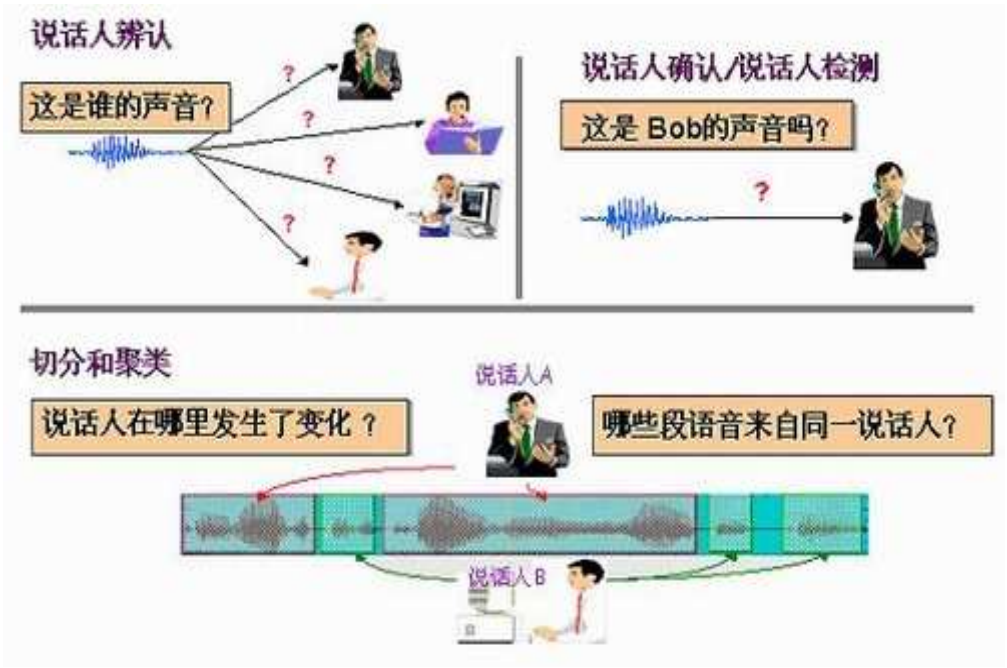


图 2-1 说话人识别任务分类[11]

从语音的内容上来讲，说话人识别又可分为与文本有关的(text-dependent)和与文本无关的(text-independent)两种说话人识别。

(1) 与文本有关的说话人识别，训练模型的语料是由参加录音的用户按照规定的内容发音，每个人的声纹模型逐个被精确地建立，而识别时也必须按规定的内容发音，因此可以达到较好的识别效果，但系统需要用户配合，如果用户的发音与规定的内容不符合，则无法正确识别该用户。

(2) 与文本无关的说话人识别，不规定说话人的发音内容，模型建立相对困难，但用户使用方便，可应用范围较宽。

根据特定的任务和应用，两种是有不同的应用范围的。比如，在银行交易时可以使用文本相关的声纹识别，因为用户自己进行交易时是愿意配合的；而在刑侦或侦听应用中则无法使用文本相关的声纹识别，因为你无法要求犯罪嫌疑人或被侦听的人配合。

本文主要研究方向是说话人辨识，根据待识别的说话人是否在注册的说话人集合内，可以分为开集(open-set)辨识和闭集(close-set)辨识两种。前者假设待识别的说话人可以在集合外，而后者假定待识别说话人在集合内。明显的，开集辨识需要一个对集合外的说话人的“拒绝辨识”的问题，这就需要通过一个先验知识设定一个阈值来判断说话人是否在集合中，或者通过训练一个“仿冒者(impostor)”模型来与集合外的说话人相匹配。

如果技术达到一定的水平，可以把文本相关识别并入文本无关识别，把闭集辨识并入开集辨识，从而提供方便的使用方法，比如北京得意音通技术有限公司的“得意”身份证就是文本无关的、开集方式的说话人辨认和确认[12]。

2.1.2 说话人识别基本原理

无论对于说话人识别的哪个任务、文本相关或无关，都需要先对说话人的声纹进行建模，这就是所谓的“训练”或者“学习”的过程。测试时，根据测试语音特征，通过判断逻辑来判定该语句的归属类别。图 2-2 显示了说话人识别的原理图。因此，若要实现对说话人的识别，需要解决以下几个问题：对语音信号的预处理和特征提取、说话人模型的建立和模型参数的调配、测试语音与说话人模型的判分计算。其中说话人模型的建立是问题的关键。

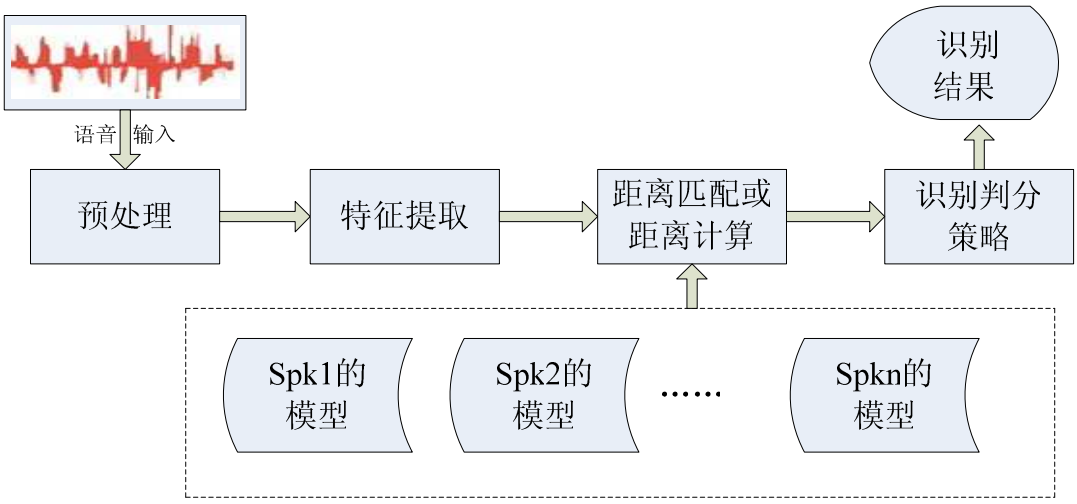


图 2-2 说话人识别基本原理

说话人识别本质上是一种模式识别，目前说话人识别的方法主要有以下几种：

- 基于模板匹配的方法。为每个说话人发的每一个字音建立这个语音的特征序列的模板。识别时，测试音特征序列与每一说话人的每一个字音的模板进行比较和匹配。通常用于文本相关的任务。

- 基于 VQ 的方法。比如 LBG 方法，效果比较好，算法的复杂度也不高，和 HMM 方法配合使用，可以收到较好的效果。

- 基于 GMM 的方法。高斯混合模型(GMM)是用多个高斯分布的线性组合近似说话人的特征分布，识别时将最能够产生测试音特征的说话人分布模型对应的说话人作为识别的结果。现在比较流行的是使用 GMM-UBM 模型的方法。

- 基于 HMM 的方法。每一个说话人的语音特征都是随着时间的变化而变化，如果将这种特征的变化过程用状态时间的转移来描述，则构成了说话人的隐马尔科夫模型(HMM)。识别时，令测试音由每一说话人的 HMM 模型产生，但由于不同的说话人 HMM 模型产生测试音的概率不同，将产生测试音概率最大的那个说话人 HMM 所对应的说话人作为识别的结果。

- 最近邻方法的方法：训练时保留所有特征矢量，识别时对每个矢量都找到训练矢量中最近的 K 个，据此进行识别，通常模型存储和相似计算的量都很大。

- 基于人工神经网络(NN)的方法：用人工神经网络进行说话人识别的三种基本结构有：1、为说话人集合中的每一说话人建立一个人工神经网络，以将这个说话人与其它说话人区分开；2、用一个神经网络实现对说话人的分类；3、为

每一对说话人建立一个神经网络，以将每一对说话人区分开。这种方法训练量很大，且模型的可推广性不好。

2.1.3 说话人识别的系统框架

典型说话人识别系统的基本结构如图 2-3 所示，其基本的组成部分如下：

(1) 语音数据预处理部分

由于说话人的声音中可能会含有背景噪音、线路噪音、以及无用的静音数据等，为了便于我们提取到有用的音频特征，我们需要对录取的语音数据进行预处理。预处理首先要对模拟的语音信号进行数字化，然后对输入计算机的数字化的语音数据进行端点检测、预加重、加窗、分帧等。

(2) 特征提取部分

在说话人识别系统中，特征提取是最重要的一个环节。特征提取就是从说话人的语音信号中提取出表示说话人个性的基本特征，一般有低级别特征(即与发音器官有关的特征)和高级别的特征(基于说话习惯和风格等有关的特征)两种。

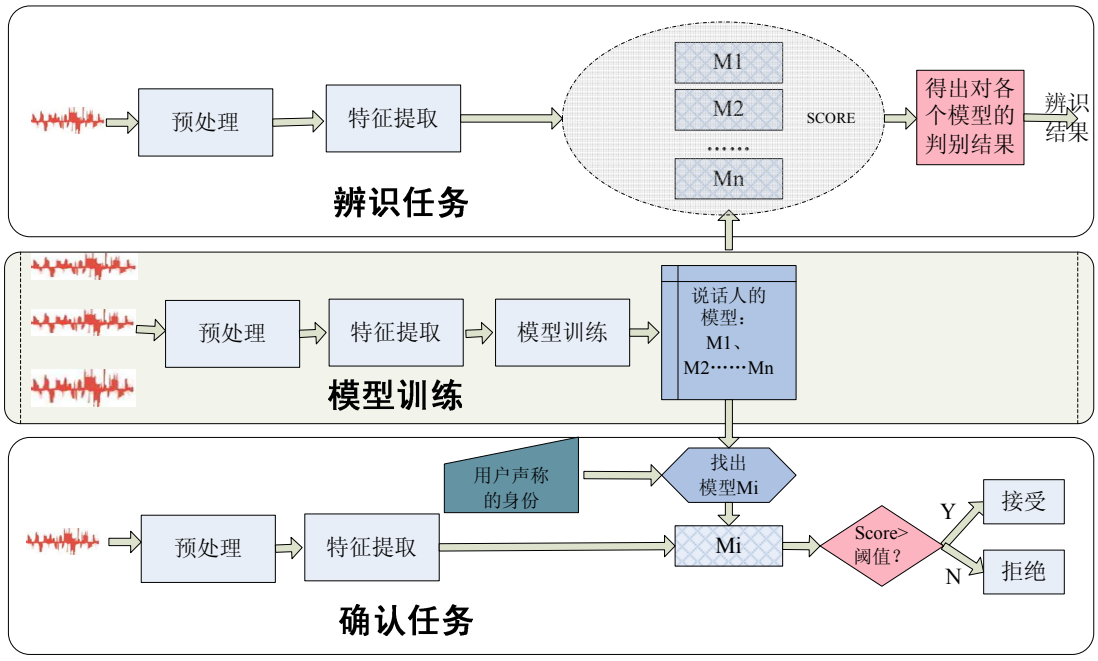


图 2-3 说话人识别典型的系统结构

(3) 模型训练部分

这一部分主要是针对提出的音频特征，采用模式匹配的方法为每个说话人建立声音模型。在目前语音特征与说话人个性特征还未很好的从语音特征中分离出

来的情况下，为每一说话人建立的说话人模型实际上都是说话人的语音特征模型。一般而言，说话人模型有产生式模型和区分性判别式模型两种。

(4) 计算分数和决策部分

将测试语音的特征和已经建好的一个模型或多个模型进行相似性比较，得出分数，并决策出用户最终的身份。

在以下的章节中，我们将针对于基于 GMM 模型的说话人辨识系统的设计与实现，给出以上四个步骤的详细介绍。

2.2 语音特征提取

语音数据经过预处理之后，得到的语音帧进行时域或者频域分析，并用相应的特征参数(向量)来描述，整个语音信号的特征则用各帧语音的特征参数所组成的参数序列描述。

经实验分析得知，能够表征说话人个性的特征有：短时能量、短时平均幅度、短时过零率、短时基音周期和基音频率、线性预测系数(LPC)、部分相关系数(PARCOR)特征、线谱对(LSP)特征、短时频谱、共振峰频率及带宽、倒谱特征、美尔倒谱系数(MFCC)和 2 维美尔倒谱系数等[13]。现在说话人识别系统使用的主流特征是 MFCC 特征和 LPC 特征，以及包含时序信息的 delta 特征。由于本文所描述的系统使用的是最常用的 MFCC(Mel Frequency Cepstral Coefficient，美尔频率倒谱系数)特征，所以本小节将详细介绍 MFCC 特征。

Mel 频率是基于人耳听觉特性提出来的，它与 Hz 频率成非线性对应关系，MFCC 则是利用他们之间的这种关系计算得到的 Hz 频谱特征。由于充分模拟了人的听觉特性，而且没有任何前提假设，MFCC 特征参数具有较好的识别性能和抗噪能力。MFCC 特征参数的计算是以“bark”为其频率基准的，它和 Hz 频率对应关系如公式 2-1 所示。由于 Mel 频率与 Hz 频率之间的非线性的对应关系，使得 MFCC 随着频率的提高，其计算精度随之下降。因此，在应用中常常只使用低频 MFCC，而丢弃高频 MFCC。

$$F_{Mel} = 2595lg(1+ f_{Hz}/700) \tag{2-1}$$

MFCC 特征提取的过程如图 2-4 所示。具体的步骤如下：

(1) 原始语音信号 S(n)经过预加重、分帧、加窗等处理得到每个语音帧的时域信号 $X'(n)$。预加重目的是为了对语音的高频部分进行加重增加其高频分辨率，一般使用的传递函数为 $H(z)=1-\alpha Z^{-1}$，α 取 0.97。根据语音信号短时平稳的特性，通过分帧操作提取语音短时特性便于建模，本文实验的帧长是 20ms。

为了平滑信号以减少每帧信号两端的预检测误差，避免频谱出现“破碎”现象，一般在说话人识别中采用汉明(Hamming)窗进行加窗处理，本文实验中采用汉明窗，窗长设为 35ms。有时还会对语音信号进行端点检测，减少数据的存储量和处理时间等。最终得到语音帧的时域信号 $X(n)$，并由此可以计算它的短时能量谱 P(f)。

(2) 将每一帧的时域信号 X(n)进行离散 FFT 或者 DFT 变换，则第 i 帧语音的频谱为：

$$X(k,i)=\sum_{n=0}^{N-1}X(n)e^{\frac{-j2\pi nk}{N}} \quad \text{(N 表示汉明窗的点数)。} \tag{2-2}$$

(3) 将上述线性频谱 X(k, i)通过在频率范围内设置的若干个具有三角滤波器特性的带通滤波器 $H_m(k)$ 的 Mel 滤波器组得到 Mel 频率。为了使结果对噪声和谱估计误差有更好的鲁棒性，将得到的 Mel 频谱取对数能量得到能量谱再经滤波器输出，总传递函数如公式 2-3 所示。

$$S(k,i)=\ln(\sum_{n=0}^{N-1}\left|X(k,i)\right|^2 H_m(k)) \tag{2-3}$$

(4) 将对数能量谱 $S(k,i)$ 经过离散余弦变换(DCT)到倒谱域即可得到 Mel 倒谱系数 $C_{Mel}(n)$，M 为滤波器的个数。

$$C_{Mel}(n)=\sum_{k=0}^{M-1}S(k,i)\cos(\frac{\pi n(m+1/2)}{M})(0\le k\le M) \tag{2-4}$$

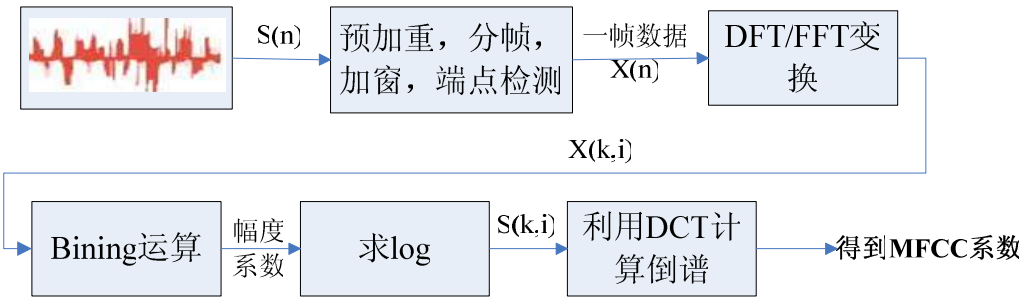


图 2-4 MFCC 特征提取过程

当有加性噪声存在的情况下，MFCC 特征的值就不具有较好的鲁棒性，所以在做识别前，使用归一化技术来对特征值进行规整来减少噪声的影响。一些研究人员提出修正的 MFCC 特征来提高鲁棒性，例如在 DCT 变换前通过提高对数

Mel 幅度(log-mel-amplitudes)到一个适当能量(2 或 3 左右)，来减少低能量(low-power)的影响[14]。

由于 MFCC 特征仅仅描述了静态的特征，而人在说话的时候，语音能量是不断变化的，因此，也可以利用语音的短时能量这个特征参数。大量实验表明，在语音特征中加入表征语音动态特征的差分参数，能够提高系统的识别性能。在本文实验中，我们采用 24 个 Mel 滤波器组，提取 12 阶的 MFCC 特征，用到了 MFCC 参数的一阶差分参数和二阶差分参数。另外，我们也采用了语音的短时归一化对数能量及其一阶、二阶差分参数。所以总的 MFCC 特征的维数是 39。

2.3 GMM 模型

2.1 节中列出了目前说话人识别领域中应用比较多的一些建模方法，但是高斯混合模型(GMM)[15]依然是当今用于文本无关的说话人识别的主流技术。GMM 是语音信号处理中的一种常用的统计模型，该模型的一个基本理论前提是只要高斯混合的数目足够多，一个任意的分布就可以在任意的精度下用这些高斯混合的加权平均来逼近。GMM 用于说话人识别，通过对大量训练语音特征数据集的统计分布进行描述，可以较好的刻画说话人不同情况下的特点，使系统更具有鲁棒性。

在以后的章节中，为了减少叙述的冗余和与下一章说明 Alize/LIA_RAL 工具包的风格保持一致，我们约定：

- Mixture：表示一个 M 阶的高斯混合模型，它是由 M 个单高斯分布的线性组合而成，用来描述语音帧特征在特征空间的分布。
- Distribution：表示高斯混合模型 Mixture 里面的一个单高斯，可以说一个 M 阶的 Mixture，就是一个由 M 个 Distribution 线性组合的 Mixture。

2.3.1 GMM 模型的基本概念

高斯混合模型(GMM)主要是用多个高斯分布分线性组合来近似描述说话人的特征分布，识别时将最能够产生测试语音特征的说话人模型对应的说话人作为识别的结果。每一个说话人的概率密度函数都是相同的，所不同的仅仅是函数中的参数。一个 M 阶的 Mixture 的概率密度函数是由 M 个高斯概率密度函数加权求和得到的，所示如下：

$$p(x)=\sum_{i=1}^{M}\omega_i b_i(x) \tag{2-4}$$

其中

$$b_i(x) = N(x, \mu_i, R_{x_i})$$

$$= \frac{1}{(2\pi)^{p/2}\left|p_i\right|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_{x_i})^T R_{x_i}^{-1}(x-\mu_{x_i})\right\} \tag{2-5}$$

这里的 ω_i 是各个 Distribution 的加权值；p 是特征的维数；$b_i(x)$ 称为核函数，是均值向量为 μ_i，协方差矩阵为 R_{x_i} 的高斯分布函数；M 称为 GMM 模型的阶数，在建立说话人模型之前设定为一常量。我们记 $\lambda \overset{\Delta}{=} \{\omega_i, \mu_i, R_{x_i} \mid i = 1,2,...M\}$ 为说话人特征分布 Mixture 中的参数。作为高斯混合分布 Mixture 的加权系数，ω_i 应满足：

$$\int_{-\infty}^{+\infty} \omega(x/\lambda) d_x = 1 \tag{2-6}$$

即

$$\sum_{i=1}^{M} \omega_i = 1 \tag{2-7}$$

图 2-5 是一个说话人的 GMM 模型，当 M=1 时，特征的分布只有一个分布中心 μ，这时的分布就是高斯分布。对于 M≠1的情况，GMM 可以有多个分布中心 μ_i (i=1,2,⋯,M)，可以描述更广的非高斯分布。由于计算 GMM 中 p(x)的时候需要求 $p*p$ 维方阵 R_{x_i} (i=1,2,⋯,M)的逆，运算量较大。为此，常常将求逆运算转化为求倒数运算以提高运算速度。

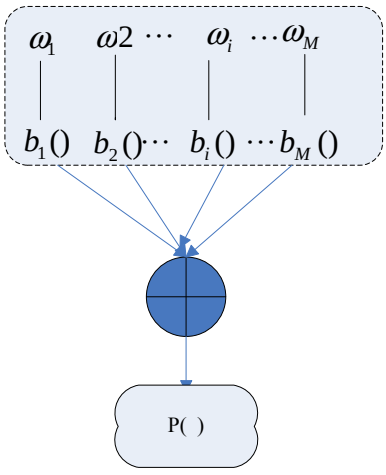


图 2-5 一个说话人的 GMM 模型

设某说话人训练语音的特征为 $\{z_t \mid t = 1,2,\cdots T\}$，按照最大似然估计(Maximum Likelihood，ML)准则，找到 GMM 模型的参数，使得这一说话人的 Mixture 产生训练语音特征 $\{z_t \mid t = 1,2,\cdots T\}$ 的概率为最大，即满足

$$\lambda = \arg_{\lambda} \max(p(z/\lambda)) \tag{2-8}$$

如果说话人的各语音帧特征统计独立，那么

$$p(z/\lambda) = \prod p(z_t/\lambda) \tag{2-9}$$

利用迭代的方法可以获得λ。事实上，几乎所有的系统都假设各帧是独立的，现在还没有任何一种方法可以描述帧与帧之间的内在联系。

由于式 2-9 是参数λ的非线性函数，所以我们很难直接求出上式的最大值。因此，常采用 EM(Expectation Maximization)算法进行参数λ的估计。EM 算法的计算式从参数λ的一个初值开始，采用 EM 算法估计出新的参数$\hat{\lambda}$，使得新的模型参数下的似然度满足

$$p(z/\hat{\lambda}) \geq p(z/\lambda) \tag{2-10}$$

新的模型参数再作为当前的参数进行训练，这样迭代运算直到模型收敛。每一次迭代运算，下面的重估公式保证了模型似然度的单调递增。

a) 混合权值的重估公式：

$$\hat{w}_i = \frac{1}{T}\sum_{t=1}^{T} p(i/z_t,\lambda) \tag{2-11}$$

b) 均值的重估公式：

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} p(i/z_t,\lambda) z_t}{\sum_{t=1}^{T} p(i/z_t,\lambda)} \tag{2-12}$$

c) 方差的重估公式：

$$\hat{\sigma}^2 = \frac{\sum_{t=1}^{T} p(i/z_t,\lambda)(z_t-\hat{\mu}_i)^2}{\sum_{t=1}^{T} p(i/z_t,\lambda)} \tag{2-13-1}$$

也可写成

$$\hat{\sigma}^2 = \frac{\sum_{t=1}^{T} p(i/z_t,\lambda) z_t^{\,2}}{\sum_{t=1}^{T} p(i/z_t,\lambda)} - \hat{\mu}^2 \tag{2-13-2}$$

其中，$p(i/z_t,\lambda)$表示z_t是分量 i 的后验概率，如公式 2-14 所示。而ω_i，μ_i，σ_i^2分别表示上一步迭代中模型的权重、均值、协方差矩阵，而$\hat{\omega}_i$，$\hat{\mu}_i$，$\hat{\sigma}_i^2$

则是更新后对应的参数。

$$p(i/z_t,\lambda)=\frac{\omega_i b_i(z_t)}{\sum_{k=1}^{M}\omega_k b_k(z_t)} \tag{2-14}$$

总体来讲，迭代估计 GMM 模型参数的过程可分为两步，根据式 2-14 计算训练数据在第 i 阶时的概率，这一步称为求数学期望值(expectation)；然后以局部最大准则用公式 2-11，2-12，2-13 估计 GMM 参数，这一步称为极大化(maximization)。

前面我们假设随机矢量的各维间是独立的，所以在计算时，我们可以采用对角协方差矩阵，亦即仅估计方差。这种方法能够极大减少模型繁琐，让模型训练更加充分。同时需要注意的是，在某些情况下，对角协方差矩阵可能会出现非常小的方差值，从而使得协方差矩阵奇异。因此在训练对角协方差阵的时候必须采用最小方差约束。亦即当新估计出来的某维方差$\hat{\sigma}_i^2$小于设定的$\sigma^2_{\min}$时，让$\hat{\sigma}_i^2$等于$\sigma^2_{\min}$，然后再进行下一次迭代估计。

本文使用的 Alize/LIA_RAL 开源软件包里面不仅对方差的下限做了限定，对其上限也做了限定。即在开始根据先验知识，设定特征方差的 VarianceFlooring(下限)和 VarianceCeiling(上限)。在每一次 EM 迭代中，都采用下面的方法进行方差限定：

$$\hat{\sigma}_i^2=\begin{cases}\text{VarianceCeiling} & \sigma^2>\text{VarianceCeiling}\\ \text{VarianceFlooring} & \sigma^2<\text{VarianceFlooring}\end{cases} \tag{2-15}$$

2.3.2 模型初始化参数设置

当给定了某个说话人的训练语音后，需要通过训练来建立说话人模型，在训练阶段就是采用最大似然准则训练出每个说话人的 GMM 参数，训练的结果就是使得训练序列能够在这种模型下得到最大的概率值。而识别阶段则是用某个人的测试序列分别代入到训练好的模型 K 中，来求取此序列的最大后验概率 $p(\lambda_i/z)$，一般用对数似然函数。由于 EM 算法是一个极大值搜索算法，它收敛于极大值。所以，参数初始化的取值对整个系统的性能至关重要。

一般而言，可以采用两种模型初始化的方法[2]：

(1) 使用一个与说话人无关的 HMM 模型对训练数据进行自动分段。训练语音帧根据其特征分到 M 个不同的类中，与初始的 M 个高斯分量相对应。每个类的均值和方差作为初始化参数。

(2) 从训练数据序列中随机选择 M 个矢量作为模型的初始化参数。尽管有实

验证明 EM 算法对于初始化参数的选择并不敏感，但是显然第一种方法要优于第二种方法。也可以首先采用聚类的方法将特征矢量归于与 Mixture 中 Distribution 个数相等的各个类中，然后分别计算各个类的方差和均值，作为初始矩阵和均值，权值是各个类中所包含的特征矢量的个数占总的特征矢量的百分比。

高斯混合参数初值包括分布系数、均值向量、协方差矩阵。在使用 EM 算法训练 GMM 模型时，GMM 模型的高斯分量的个数 M 和模型的初始参数必须首先确定。这是一个困难而又困难的问题。如果 M 选取的值太小，则训练出来 GMM 模型不能有效的刻画说话人的语音特征，从而使整个系统的性能下降；如果 M 的取值太大，则模型的参数会很多，从有效的训练数据中可能得不到收敛的模型参数，同时，训练得到的模型参数误差会很大。而且，如果有太多的模型参数，那么模型训练和判分识别的时间复杂度和空间复杂度将大大增加。高斯分量 M 的大小，很难从理论上推导出来，可以根据不同的识别系统，由实验确定。

2.3.3 说话人识别判定

假设系统中有 N 个说话人的模型，为每一个说话人 $Spk_i(i=1,2,\cdots,N)$ 建立的模型参数分别为 $\lambda_i(i=1,2,\cdots,N)$。识别则是对测试语音的特征 $\{z_t|t=1,2,\cdots,T\}$，按照最大后验概率(Maximum a Posteriori，MAP)准则找到满足

$$\lambda_k = \arg_{\lambda} \max\, p(\lambda / z) \tag{2-16}$$

的 λ_k ($\lambda_k \in \{\lambda_i|i=1,2,\cdots,N\}$)。在 $p(\lambda_i)$ 未知的情况下，可以通过找到满足式 2-16 的 λ_k 对应的说话人 Spk_k 作为识别的结果。

一般在实验中，我们都会将判分结果取对数，此时，说话人识别的结果就可变为计算

$$\lambda_k = \arg\max_{1\le k\le N} \sum_{t=1}^{T} \log(p(z_t / \lambda_k)) \tag{2-17}$$

2.3.4 训练数据不充分的问题

在实际应用中，往往很难得到大量的充分的数据进行模型的训练，尤其做实时说话人识别系统的情况下。此时，已有的说话人识别的语料库会因为信道不匹配、背景噪音、应用环境等原因的影响而不能作为说话人识别系统的训练数据时，采集大量的语音数据就会成为一个很大的门槛。“巧妇难为无米之炊”！没有适配的数据，一切的后续工作都不可能顺利的进行下去！

由于训练数据的不充分，GMM 模型的协方差矩阵的一些分量就可能会很小，

这些很小的值对模型参数的似然度函数影响很大，严重影响系统的性能。为了避免小的值对系统性能的影响，一种方法如 2.3.1 节所述，在 EM 迭代中设置协方差的门限值。另外，在说话人训练数据不充分的情况下，我们可以事先建立一个所有说话人通用的模型，在给特定的说话人建立模型的时候，可以只在这个通用的模型上修改有效的 GMM 模型参数即可，这也就是下一节即将讲到的 GMM-UBM 模型训练方法。

2.4 基于 GMM-UBM 的说话人模型训练

2.4.1 引言

在开始这一节之前，我们首先来了解一下说话人确认识别的机理，然后再引入 GMM-UBM 模型。基于似然率(Likelihood Ratio，LR)的说话人确认系统有两个基本假设：

H0：语音信号 z 是来自假定的说话人 S；

H1：语音信号 z 不是假定的说话人 S(即是一个仿冒者)

对数似然比测试如下：

$$\wedge(z)=\frac{p(z/H0)}{p(z/H1)}\begin{cases}>\theta, Accept H0\\<\theta, Accept H1\end{cases} \tag{2-18}$$

其中，$p(z/Hi)$，i=0，1，是假设语音 z 是属于声称身份 Hi 的概率密度函数。θ 是接受或拒绝的阈值。说话人确认的基本目标就是计算两个似然值 $p(z/H0)$ 和 $p(z/H1)$。图 2-6 是基于似然率的说话人确认系统的基本框架。

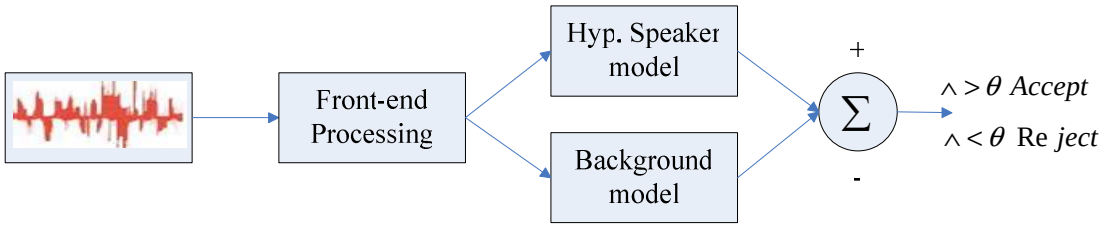


图 2-6 基于似然率的说话人确认系统[16]

假设在 H0 用 GMM 模型 λ_{hyp} 表示；H1，即仿冒者模型，用 $\lambda_{\overline{hyp}}$ 表示，则式 2-18 用对数似然比表示为

$$\wedge(z)=\log\left(\frac{p(z/\lambda_{hyp})}{p(z/\lambda_{\overline{hyp}})}\right)=\log p(z/\lambda_{hyp})-\log p(z/\lambda_{\overline{hyp}}) \tag{2-19}$$

模型 H0 能够通过训练语音 z 有效的估计出来，由于模型 $\lambda_{\overline{hyp}}$ 需要表示所有可能的候选说话人的特征空间，有可能没有很好的训练，这样便会影响系统的性能。有两种方法可以用来训练假定模型(alternative hypothesis models)，也称仿冒者模型(impostor models)，如图 2-7 所示。一种方法，如图 2-7(a)，给定一个含有 N 个说话人(背景)模型的集合 $\{\lambda_1,\lambda_2,\cdots,\lambda_N\}$，这样，仿冒者模型的似然值就是这 N 个模型集合的判分函数，计算 $p(z/\lambda_{\overline{hyp}})$ 为

$$p(z/\lambda_{\overline{hyp}})=F(p(z/\lambda_1),\cdots,p(z/\lambda_N)) \tag{2-20-1}$$

函数 $F()$ 是取背景说话人集合中 N 个似然值的均值或最大值的函数。

另一种方法，如图 2-7(b)，就是建立一个说话人无关的、通用的背景模型(Background Model，或者 general model ，或 world model) λ_{ubm}，即在与说话人无关的语音集中训练模型以表示一般的语音特性，这种模型对于 GMM 模型来讲比较容易做自适应，则计算 $p(z/\lambda_{\overline{hyp}})$ 简化为计算

$$p(z/\lambda_{\overline{hyp}})=p(z/\lambda_{ubm}) \tag{2-20-2}$$

这种方法的好处就是一个简单的说话人无关的模型可以针对于一个特定的任务训练一次，然后可以在以后所有的确认任务中使用这个模型。在说话人确认任务中，所有的仿冒者模型 $\lambda_{\overline{hyp}}$ 使用这个通用的背景模型 λ_{ubm}，我们将这个模型称为 UBM 模型[16]。基于 GMM-UBM 的说话人识别的系统结构流程如图 2-8 所示。在下面的几个小节中，我们将给出每一步的详细说明。

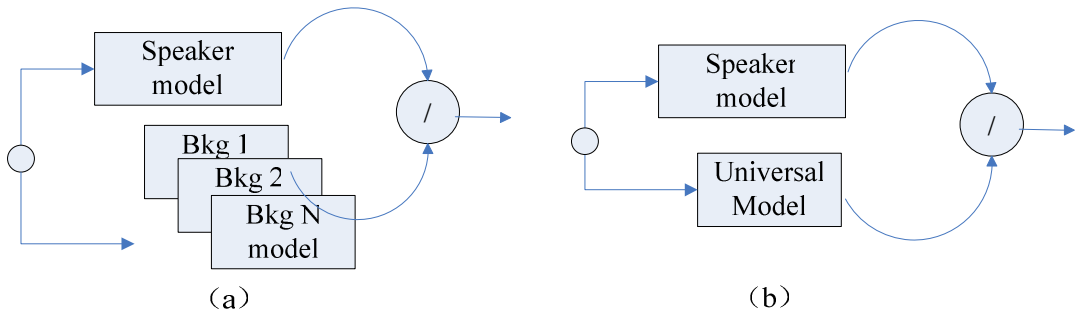


图 2-7 基于仿冒者模型的说话人辨识

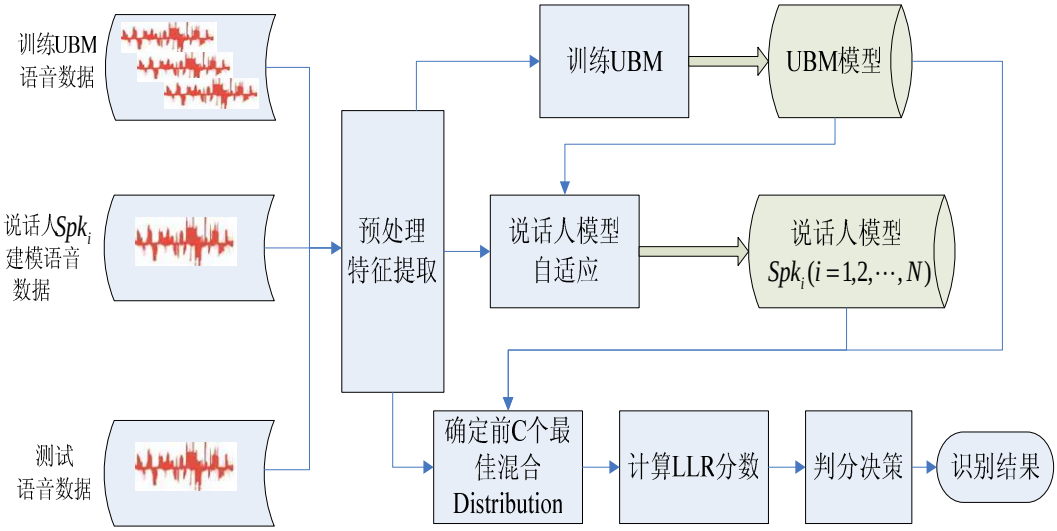


图 2-8 GMM-UBM 说话人识别系统流程

2.4.2 UBM 模型

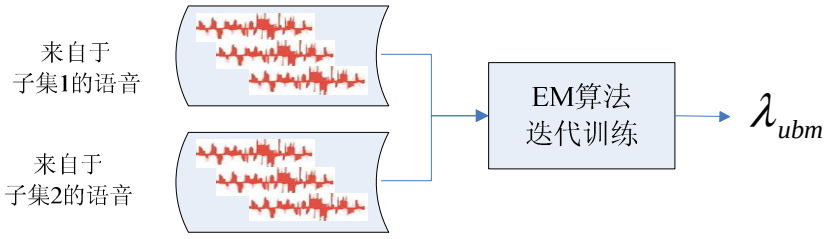
GMM-UBM 模型[16]最早就是用在上述的说话人确认系统中，它是用一个简单的说话人无关的背景模型来表示 $p(z/\lambda_{\overline{hyp}})$。UBM 本质上就是一个用来表示说话人无关的特征分布的一个“很大的”GMM 模型。该模型通常由数百人甚至上千人、男女平衡的数小时的语音训练得到，用于表示说话人的统计平均发音特性。基于 GMM-UBM 模型有两个好处：一是，说话人模型是在 UBM 上根据说话人的训练语音自适应得到的，这样，对于说话人训练语音覆盖到的发音，可以用该说话人自身的语音建模；而说话人训练语音没有覆盖到的发音，可以用 UBM 里的发音分布近似，从而减少测试语音与训练语音在声学空间上由于分布不同所带来的影响。二是，UBM 可以被看成是一个“标准参考者”模型，这样在做身份确认时，可以用测试语音在 UBM 上的得分作为一种参考阈值。

一般 UBM 用于说话人确认的时候，会根据一些先验知识来训练 UBM。例如，在 NIST SRE 中，如果测试语音数据是来自于本地男性的长途电话语音，在这种情况下，测试男性语音时，应仅使用男性的电话语音来训练 UBM 模型。当没有性别的先验知识时，我们应使用性别无关的，即前文所说的男女均衡的语音数据训练。除了性别上的差异，现在没有客观的方法来评价训练语音中说话人个数或者语音的数量对训练 UBM 的影响。根据 NIST SRE 的经验数据，发现用一个小时的语音数据训练的 UBM 并不比 6 个小时的语音数据训练出来的效果差，但这必须是基于训练数据来自于相同的说话人群体的前提下。

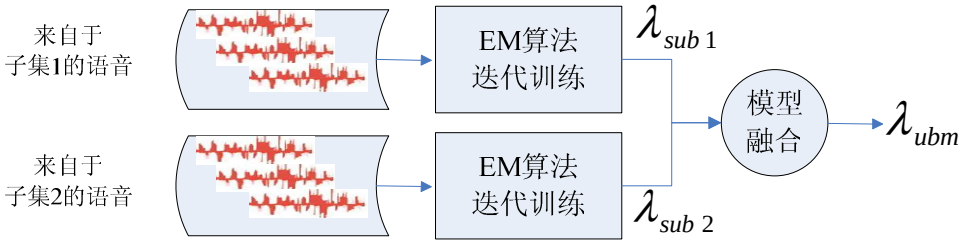
一般我们会将训练数据按照语音数据的特点分成不同的子集，如说话人群

体、性别、麦克风、环境等。

训练模型，得到最终的 UBM，最简单的一种方法就是仅仅使用所有的训练数据通过 EM 算法得到 UBM，如图 2-8(a)所示。但是需要注意的是，要保证这些数据集合中的每个子集数据的均衡性；否则，最终的模型会偏向于占优势的数据子集。另一种方法就是在训练集上训练不同的 UBM，例如一个男性的 UBM 和一个女性的 UBM，然后将这些子集的模型融合在一起，如图 2-8(b)所示。这种方法的好处就是可以有效的利用不均衡的训练数据、控制最终 UBM 的各个构成成分。另外在一些论文中还提到其它的方法[17,18]。



（a）所有子集的训练数据都用来训练UBM



（b）先训练子集的模型，然后再融合成UBM

图 2-9 训练 UBM 模型策略

2.4.3 说话人模型自适应

UBM 模型代表了所有说话人的语音通性，如何利用某个说话人较短的语音来找到这个说话人的语音个性呢？这就是如何利用少量的训练数据快速的得到各个目标说话人模型的问题。一般通过自适应的方法来得到目标说话人模型。所谓自适应，即利用目标说话人的少量语音数据，调整系统参数，最终得到适合该目标说话人的模型。现在主要有两种自适应算法：基于最大后验概率的(Maximum a Posterior，MAP)和基于变换的(如 Maximum Likelihood Linear Regression，MLLR)。本文系统是通过贝叶斯(Bayes)自适应方法来获得每个目标说话人模型。贝叶斯自适应方法的最大特点是，及时在数据量相当大的情况下，也能迅速有效的获得模型。

贝叶斯自适应算法分为两个步骤：

(1) 与 EM 算法的 E-Step 相同，对 UBM 的每个 Distribution，计算目标说话人的模型参数的估计。

(2) 为了自适应，这些新的模型参数与旧的模型参数通过一个数据有关的混合系数进行合并。这个数据有关的混合系数可以这样假定：使得目标说话人的高数据量的 Distribution 在最后的参数估计中更依赖新的模型参数，而目标说话人的低数据量的 Distribution 更依赖旧的模型参数。这里的模型参数即权重、方差和均值。

贝叶斯自适应的具体算法如下：给定一个 UBM 模型和目标说话人的语音序列 $z=\{z_1,z_2,\cdots,z_T\}$，则 UBM 中第 i 个 Distribution，可以计算语音序列在各个 Distribution 中概率分布：

$$P(i\mid z_t)=\frac{\omega_i p_i(z_t)}{\sum_{j=1}^{M}\omega_j p_j(z_t)} \tag{2-21}$$

然后利用 $P(i\mid z_t)$ 和 z_t 计算充分统计：权重、均值和方差参数，用下列公式计算，这一步与 EM 算法的 E-Step 相同。

$$n_i=\sum_{t=1}^{T}P(i\mid z_t) \tag{2-22}$$

$$E_i[z]=\frac{1}{n_i}\sum_{t=1}^{T}P(i\mid z_t)z_t \tag{2-23}$$

$$E_i[z^2]=\frac{1}{n_i}\sum_{t=1}^{T}P(i\mid z_t)z_t^2 \tag{2-24}$$

最后，这些由训练数据产生的新的充分统计量用来更新旧的 UBM 的第 i 个 Distribution 的充分统计量，产生第 i 个混合变量的自适应的参数。用下列的公式进行计算[19]

$$\hat{w}_i=[\alpha_i^{w}n_i/T+(1-\alpha_i^{w})w_i]\gamma \tag{2-25}$$

$$\hat{\mu}_i=\alpha_i^{m}E_i[x]+(1-\alpha_i^{m})\mu_i \tag{2-26}$$

$$\hat{\sigma}_i^{2}=\alpha_i^{v}E_i[x^2]+(1-\alpha_i^{v})(\mu_i^{2}+\sigma_i^{2})-\hat{\mu}_i^{2} \tag{2-27}$$

$\{\alpha_i^{w},\alpha_i^{m},\alpha_i^{v}\}$是自适应系数，控制新旧估计之间的均衡，分别控制权值、均值和方差。尺度因子γ作用于所有的自适应混合权值，确保和为 1。

对于每个 Distribution 和每个参数，一个与数据有关的自适应系数 α^{δ}，$\delta \in \{w,m,v\}$用在上面的公式中，其定义为：

$$\alpha_i^{\delta} = n_i /(n_i + \gamma^{\delta}) \tag{2-28}$$

其中，γ^{δ} 是参数 δ 的一个固定关系因子。通常在 GMM-UBM 系统中，采用一种简单的自适应系数，$\alpha_i^w = \alpha_i^m = \alpha_i^v = n_i /(n_i + \gamma^{\delta})$，一般来讲，$\gamma$ 的取值范围为 8~20，通常取为 γ=16。另外，由先前的研究可知，只有均值向量做自适应会得到较好的识别率，因此本文的说话人辨识系统在使用 GMM-UBM 模型只是对模型的均值做了自适应。

2.4.4 对数似然比计算

基于 GMM-UBM 模型的说话人判分决策是基于对数似然比的计算，假设测试语音特征向量为 $z = \{z_1, z_2, \cdots, z_T\}$，系统共有 N 个说话人模型 $\lambda_{Spk} = \{\lambda_{Spk_1}, \lambda_{Spk_2}, \cdots, \lambda_{Spk_N}\}$和一个 UBM 模型 λ_{ubm}，针对于说话人辨识的任务，我们采用 2-19 公式计算对数似然比，此时的公式我们采用另一种符号意义来标识

$$\wedge_i (z) = \log p(z / \lambda_{Spk_i}) - \log p(z / \lambda_{ubm}) \tag{2-29}$$

$\wedge_i (z)$ 表示在第 i 个说话人模型 λ_{Spk_i} 上的对数似然比。最后，我们选择 $\max(\wedge_i (z))$ 的模型对应的说话人作为辨识的结果。

由于目标说话人模型是从 UBM 模型中自适应计算得到的，所以对数似然比的计算可以采用下面的快速算法得到。快速算法是基于两个方面的事实：一是当一个大型的 GMM 对一个特征矢量进行测试时，只有少数的混合成员对概率值的贡献较大，这是因为 GMM 表示了一个大空间的分布情况，而一个单一的向量只是接近 GMM 中的少数几个混合成员。这样，仅仅利用前 C 个最佳混合分量就可以很好地近似概率值。另一个事实是，自适应后的 GMM 模型中仍然保留了与 UBM 对应的混合成员，因此这个变量接近于 UBM 中的某个特定的混合成员时，也将接近于目标说话人模型的相应成员。鉴于以上两个方面，快速算法具体如下：

(1) 对于每个特征向量 z_i，确定 UBM 中最佳的 C 个混合成员，并利用这最佳的 C 个混合成员计算 UBM 概率值。

(2) 仅利用目标说话人模型中相对应的 C 个 Distribution 对向量进行打分来计算测试语音的似然值。对于具有 M 个 Distribution 的 UBM 而言，对每个向量仅需要进行 M+C 次高斯运算，这比通常的计算 2M 个高斯运算，节省的较大的运算量，提高了识别的速度。

第三章 说话人辨识模块的设计与实现

3.1 概述

Alize 是由法国 ELISA 联盟发起，法国科研部资助的一个 Technolanguage 框架[9]。2007 年 3 月，Alize 被纳入法国 Mistral 项目。Mistral 还召集一些著名实验室(如，IRTT、LIUM、CLIPS、LIA 等)和一些私有公司(如，Thales 公司、Calistel 等)等合作伙伴来设计一些辅助工具。LIA_RAL 软件包是 Alize 的一个补充库，在 Alize 的基础上增加了一些功能(如特征分析)。

Alize 是自动说话人识别在“小冰山”时期开始的一个项目，最初是作为一个库来开发的，被纳入 Mistral 项目后，又补充了一些功能。目前，Alize 已经被世界各地的许多大学、科研机构和企业使用。Alize 在一些重大的评估活动，如 NIST SRE，RT 和法国 ESTER 等，都取得了优秀的成绩。目前，Mistral 的性能也将通过定期参加各种国际活动来进行测评。

Alize 主要是基于 GMM-UBM 算法的，但也包括说话人识别领域的最新技术，如潜在因素分析(Latent Factor Analysis)、无监督自适应(Unsupervised Adaptation)、SVM 超向量一类的区别性分类器等。它主要是采用 UML 面向对象的设计思想，搭建一种基于服务器的框架，而每个服务器就是一个相同或相近的功能函数的集合。

本章将着眼于描述使用软件包，Alize 和 LIA_RAL，进行说话人辨识模块的设计与实现。本文使用的是 Alize 软件包 1.31 版本，LIA_RAL 软件包 1.03 版本。

3.2 Alize 和 LIA_RAL

Alize 和 LIA_RAL 都是法国 Mistral 工程公布的开源软件包，两者之间既有区别也有联系。首先，Alize 是一个数学、统计学和 I/O 的框架，而 LIA_RAL 则是在这个框架的基础上建立起来的专门用于说话人识别的软件包。

Alize 并没有包含针对说话人(或语音)识别的具体算法。在一定程度上，可以看成是一个偏向于语音研究领域中数据结构和算法的一般性的统计框架。这主要是由于它并不包含一些训练算法和对高斯分布的特殊处理。LIA_RAL[1]是针对于

说话人识别的软件包，主要包含基于 GMM-UBM 说话人识别的每个步骤的可执行文件和源码。本节具体的描述将从两者之间的关系、组成和使用特性等这几个方面进行。

3.2.1 Alize 库简介

Alize 库实际上是由很多用标准 C++编写的类组成的，这些类主要分成两个级别：基本的级别是对各个功能类的技术复杂性的封装(数据获取，计算，存储，为用户提供数据等)，主要是不让用户自己管理内存分配；高层的级别包括由用户操作的公用工程和算法(列表管理，模型初始化，MAP 算法，……)。

Alize 基本架构是建立在一些数据和计算服务器(servers)的基础上的：

- 数据音频服务器(Data audio server)，可以存储来自于麦克风、文件或其它音频资源的数据，旨在为用户提供一个无限大的缓冲区作为用户的音频资源。目前，这类服务器尚未实现。
- 特征服务器(Feature Server)，存储来自文件或音频数据计算结果的特征，它允许用户有无限大的缓冲区。
- 混合/分布服务器(Mixture/Distribution Server)，目的是存储说话人语音模型(GMM 模型)。根据特征或者是从文件中加载的特征来计算，这个服务器既没有缓冲区的概念，也没有持续时间的存在。
- 统计服务器(Statistic Server)，集中了最常用的算法(似然性计算，EM 算法……)，由存储的结果或迭代计算的结果得到一个特征集的全局均值。

Alize 可以分成更小的单元，其中本文使用到的单元主要有：

(1) 配置文件解析程序

配置文件的解析和管理程序，能够处理一个有稍许严格语法要求的 java 风格的属性文件和 XML 格式的文件。Alize 里，通常将配置文件称为 config 文件，一个 config 文件就是一个包含一些“键-值”对的文件，它是整个程序运行的 “地图”，一些关键的参数都是通过配置文件来传递给程序的。config 属性值可以写成文件的格式，可以在程序中设置，也可以在命令行中动态输入。

(2) 文件 I/O

Alize 里面用两个类来处理文件的读和写的问题，前台使用 C++的文件流、后台则用 C 的 API 来处理。这样可以提高处理批量文件的效率，提高系统的性能。

(3) 特征文件

为了处理磁盘上的特征向量，Alize 支持一些特征文件格式(大都是 HTK 和 Apro4[20]的文件格式)。它也支持将一些特征文件的列表组成一个单一的文件形

式，这种文件称为列表文件(list files)。这主要是通过 XLine 类来管理的。有了这个方法，程序就可以在一次运行中执行多个文件。另一个用处体现在：当训练模型的时候，可以将这些子集的文件名放在 list 文件中，执行时只需要输入 list 文件名就可以了，这样大大降低了参数输入的工作量。

(4) 统计函数

Alize 能够做简单的迭代、统计运算。通过对向量的一系列计算，返回均值和方差等。同时，也能够把数据合成直方图，显示出来。

(5) 高斯分布(Distribution)

Alize 能够处理全方差和协方差矩阵，由于全方差矩阵可以由对角矩阵估计出来，所以它几乎不会在 Mixture 中使用。

(6) 高斯混合(Mixture)

Alize 支持的计算单位是 Distribution，但只有 Mixture 才能被程序执行，所有的代码最终的目标是合成高斯模型 Mixture。所有对模型的磁盘 I/O 的单位是 Mixture，每个 Mixture 里的 Distribution 是不能单独处理的。

(7) 管理

Alize 里面一些主要的类，被称为服务器(Servers)，用来处理命名实体在内存中的存储和计算。具体的 Server 的使用特性如前文所述。

根据 Alize 的使用特性，就可以调用里面的一些单元来帮助我们设计和实现说话人辨识模块。由于 Alize 并未提供 EM 迭代、MAP 自适应等算法，所以在本文的实验中，采用 Alize 库实现的是一个基于 GMM 模型的说话人识别模块。也就是，由说话人的训练数据为每个说话人建立一个 GMM 模型，识别时则是找到有最大后验概率的模型所对应的说话人标识作为识别的结果。使用 Alize 实现基于 GMM 模型的说话人辨识模块的主要流程如下：

第一步，模型初始化。利用每个说话人的训练数据，进行模型的初始化。模型初始化，就是利用训练特征文件，得到全局的均值和方差，而每个 Distribution 的权重则设成相同的值。然后利用模型计算器来进行模型的初始化，保存成 **_init.xml 的文件，待下一步模型训练使用。

第二步，模型训练。根据先验知识设置训练模型的重要参数，如方差的上下限阈值、模型训练次数和 EM 迭代的次数等；将第一步得到的初始化模型作为说话人的第一次训练之初的模型，初始化模型训练器，开始进行 nbTrainIt 次的模型训练。在每次训练中，实验进行 nbEmIt 次的 EM 迭代运算，每次 EM 迭代运算都要对最小/最大方差进行限制，具体的方法见 2.3.1 小节所述。 实验中，nbTrainIt 和 nbEmIt 这两个值的选取是非常重要的。如果设置的过小，就会导致模型训练的不充分，不能充分表示说话人的语音特性；如果，设置的过大，会存在“过训练(over-training)”的问题，使训练模型对训练数据过分依赖，降低识别

效果。模型初始化和训练的流程如图 3-1 所示。训练后的说话人模型会放在模型数据库中，每个说话人模型的标识符将写入到模型列表文件中，以便下一步判分识别时加载所有的说话人模型。

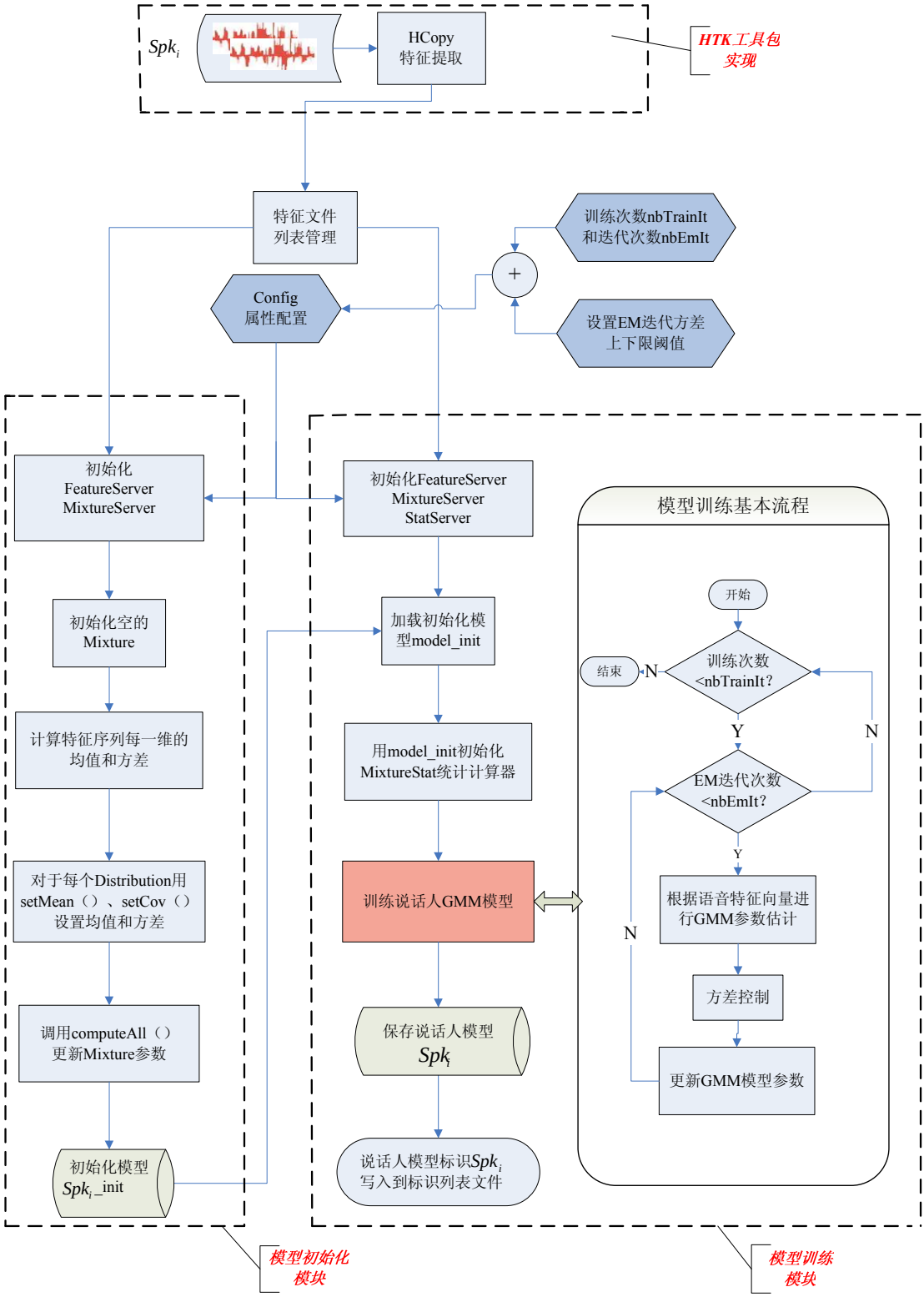


图 3-1 GMM 模型训练流程

第三步，辨识说话人。这一步主要是根据说话人的测试语音特征序列，在说话人模型数据库中找到一个对数似然率(LLR)最高的模型的标识作为识别的结果。我们采用一种“闭集”的说话人辨识技术，集合内说话人的测试语音会出现识别错误和识别正确这两种情况，但是如果是集合外的说话人，系统只会出现识别错误的结果。辨识说话人的处理流程如图 3-2 所示。

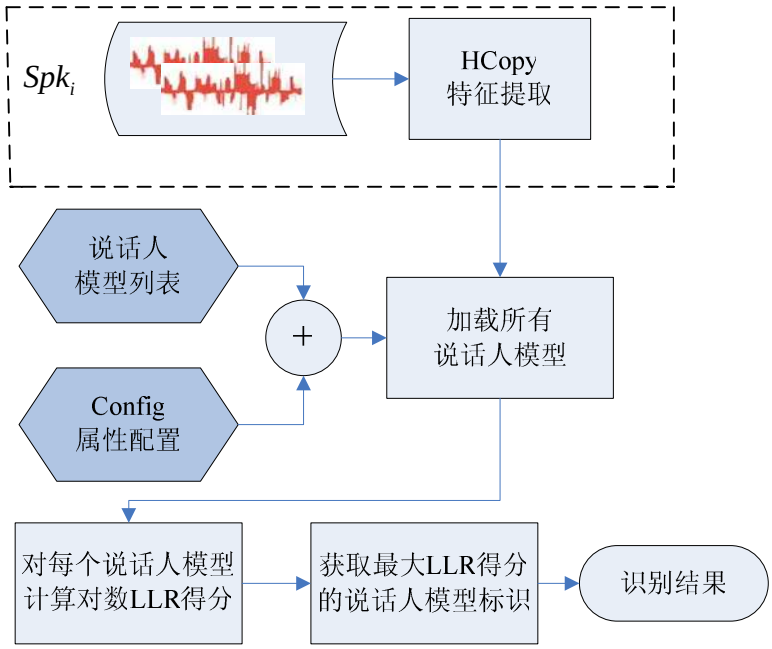


图 3-2 辨识说话人流程

3.2.2 LIA_RAL 库使用简介

LIA_RAL 包设计的主要目标是使所有程序中的算法都能在命令行下使用，另外，这个工具包还提供各个算法的源码。LIA_RAL 没有提供特征提取的方法，它假设所有的输入都是“可用的”特征向量。一般我们使用 SPro、HTK 等工具包来提取说话人特征。LIA_RAL 不支持同时对同一个文件读且写的操作，文件的处理形式是：输入、处理、保存成其它的文件。在本文的实验中，我们将使用 HTK 工具包中的 HCopy 来提取语音的 MFCC 特征。LIA_RAL 中主要的模块单元如图 3-3 所示。每个模块的具体功能如表 3-1 所示。

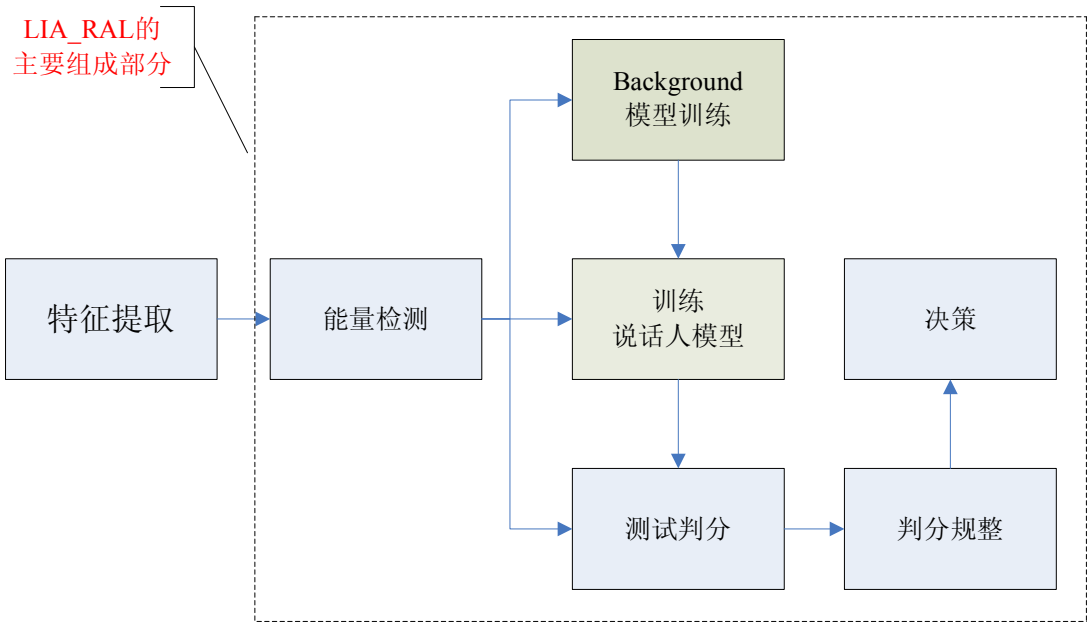


图 3-3 LIA_RAL 的主要模块

表 3-1 LIA_RAL 各模块功能表

模块	功能说明
能量检测 (EnergyDetector)	分析语音特征的能量组成部分，产生一个特征文件的标签文件，标出能量最高的特征区间，是一个典型的语音/非语音的检测。
特征规整 (NormFeat)	使用一些特征归一化的方法，对输入语音特征(如 MFCC，LFCC，……)进行规整，主要的功能就是将输入特征的分布改成另外一种形式，比如，将特征改成一种“0-均值和 1-方差”的高斯分布。
训练 UBM 模型 (TrainWorld)	是一个通过 EM 迭代算法训练 GMM 模型的应用模块，从输入的特征文件中随机选取一些特征来进行模型的初始化。输入特征的维数和目标 GMM 模型的不一定要相同。EM 迭代训练可以首先根据给定的一个随机系数，从输入特征文件中随机选取一些语音特征训练，然后将所有的语音特征用来做最后的迭代训练。输出是一个 UBM 模型(通常称为 world 模型)。
训练目标 说话人模型 (TrainTarget)	实现说话人模型的自适应模块，在语音识别领域，它主要是通过 MAP 算法在 world 模型上自适应出说话人模型。 它支持以下两种 MAP 算法： (1) MAPConst：通过计算语音特征参数在 world 模型上经过 EM 算法得到的一些参数的线性组合来得到模型的参数估计(如公式 2-22，2-23，2-24)。这种组合的权重通过 config 里面的 alpha 给定(α 对应于 world 模型，1-α 对应于目标模型)。这种方法实际上就是由语

	音特征参数在 UBM 模型上的均值做自适应来得到目标模型。没有提供方差和权重的自适应方法。本文采用的则是这种自适应方法。 2)MAPOccDep：通过计算语音特征参数在 world 模型上经过 EM 算法得到的一些参数的线性组合来得到模型的参数估计。这种方法考虑到每个高斯的后验概率 n，这种组合的权重由参数 γ 来给定(对于 world 模型采用 $\frac{n}{n+\gamma}$，而目标模型采用 1-$\frac{n}{n+\gamma}$)。
计算判分 (ComputeTest)	主要是在给定的模型上，得出测试语音特征参数在模型上的得分。支持多个测试句在多个模型上的测试模式。
判分规整 (ComputeNorm)	利用一些归一化技术对对数似然分数进行规整。主要用到的规整技术有 ZNorm，TNorm 和 ZTNorm。但是，这些归一化技术需要进行冒充者测试及测试的似然分数。

LIA_RAL 工具包里所有的可执行文件都可以在 Win32 的命令行里直接执行，由前文所述，LIA_RAL 是在 Alize 的基础上开发的一个开源包，所以它采用与 Alize 一样的参数输入方法—config 配置文件。使用每个模块时，都可以将输入参数写成 config 文件，作为程序的输入。config 文件相当于程序的眼睛，他将告诉程序往哪个方向执行、采用什么样的算法、训练什么样的模型等。所以，每个模块 config 文件的设置是相当重要的。本文将在附录Ⅲ中给出每个模块 config 文件的内容。至于 LIA_RAL 使用的文件类型主要可以分成五种：特征文件、模型 Mixture 文件、标签文件、列表文件和索引文件。每种类型的文件是通过前缀和后缀来区分的。Alize 里面不能自动的找到特征文件的格式，所以每个程序都要指明加载和保存的特征文件的类型。同样的，每一个 Mixture 的格式也要事先指定(指定成存储压缩格式，或是 XML 格式)。另外，Alize 并不能确定文件的字节序列，所以在 config 文件中需要事先指定，所有关于 Mixture 的计算都有一个 LLK 的阈值，避免除零的问题。

3.2.3 小结

由于 LIA_RAL 是在 Alize 的基础上开发出来的，所以仅使用 LIA_RAL 包是不够的。另外，Alize 是一个典型的用于说话人识别的工具包，而 LIA_RAL 是对 Alize 性能的一个评估手段。虽然评估 Alize 的使用性能并不是我们感兴趣的部分，但是 LIA_RAL 里面包含了基于 GMM-UBM 的说话人识别的算法框架。所以，这两个软件包必须结合在一起使用。既然，我们已经明白了这两个软件包的关系，所以在下面的章节中，我们将称 Alize/LIA_RAL 为本文使用的工具包。

3.3 设计基于 Alize/LIA_RAL 的说话人辨识模块

本文的工作就是搭建一个完整的基于 Alize/LIA_RAL 的文本无关的说话人辨识系统，系统能够根据说话人的语音判断出说话人的身份，并能以友好的界面将结果展示给说话人。在系统面前，说话人要做的就是：点一下鼠标，张开嘴巴说话！

在介绍系统之前，我们再来明确一下说话人辨识的含义。所谓“辨识”就是给出一段语音，系统能够辨别出是谁在说话。我们知道，说话人辨识任务存在一个“开集”与“闭集”的问题。如果系统中说话人模型个数是一定的，那么这个系统是一个“闭集”系统，辨识任务就简单的转化为在这个集合中找到与说话人语音最匹配的模型的标识即可；但是，若系统是“开集”的，也就是说说话人模型的个数是可变的，那么系统还要能判断出说话人是不是在这个集合中。它的典型的结构如第二章的图 2-3 所示。本文搭建的说话人辨识系统是一个面向实际应用的系统，可以认为是“开集”的，也可以看成是“闭集”的，说它是“开集”的，是因为系统允许增加说话人模型；说它是“闭集”的，是因为它的辨识原理是基于“闭集”系统的辨识原理。具体的原因，将在下面的章节中说明。此外，本文的说话人辨识模块采用 GMM 和 GMM-UBM 两种建模方法。当我们能够获取目标说话人充足的训练数据时，采用 GMM 的方法为说话人建模；当目标说话人的训练数据比较少的情况下，而又想将说话人模型迅速的加入到模型库中时，则采用在 UBM 做自适应的方法建模。

本文说话人辨识系统的功能主要有音频控制、说话人建模、说话人模型自适应、说话人身份识别、语音提示等，3-4 示出了说话人辨识系统的物理结构图。

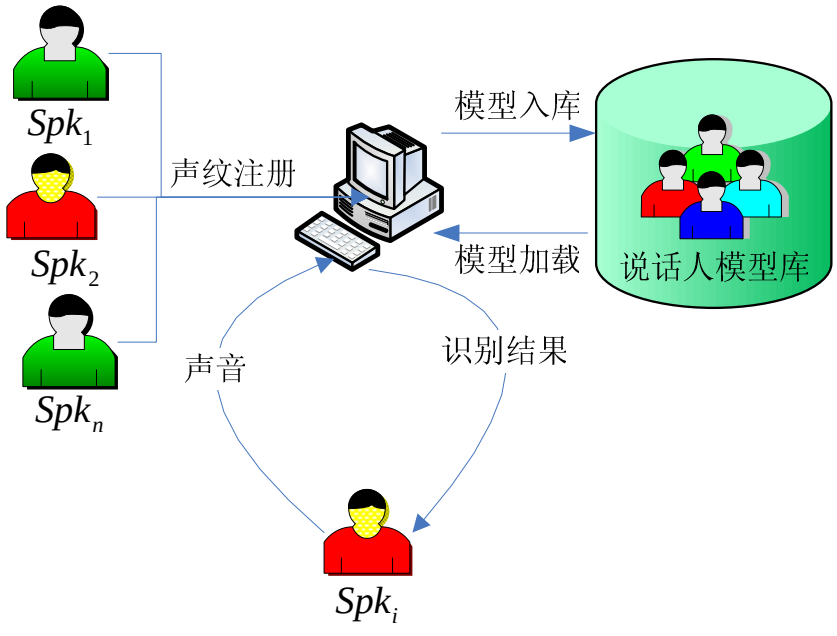


图 3-4 说话人辨识系统架物理结构图

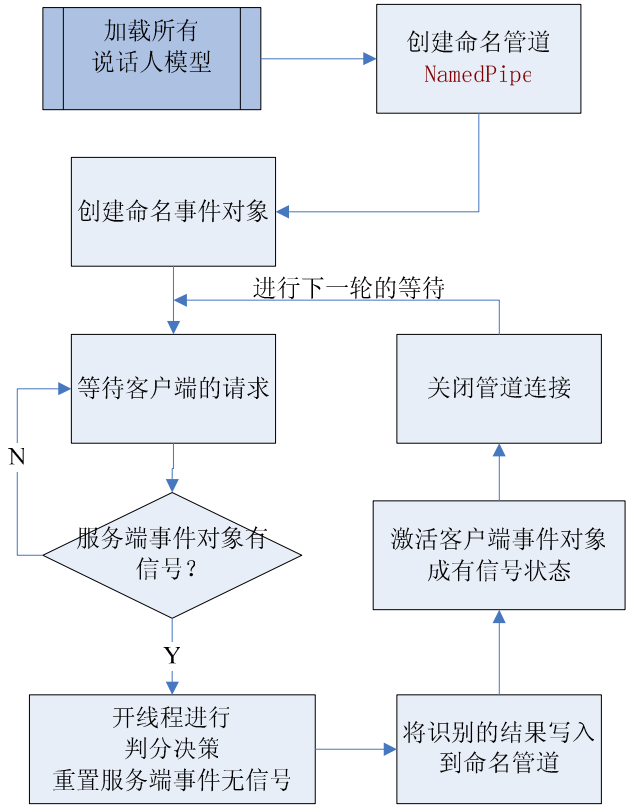
3.4 使用进程间的通信和同步技术实现实时识别

从上文中 Alize/LIA_RAL 的使用特性可以知道，识别判分模块要先加载说话人模型。但是当模型库的说话人模型比较多时，加载模型便需要较长的时间，根据本文的实验经验，加载一个 256 个 Distribution 的 Mixture 需要 50 多秒的时间。这样，说话人模型较多的情况下，加载模型将会花费相当长的时间。若是直接调用 computeTest 模块，根据得到的判分文件求最大值，然后得到识别结果。这样，每一次识别都要加载模型，这在时间上和空间上的代价是我们所不能容忍的，更不符合实时识别的要求。所以，在真正搭建辨识系统的时候，我们并不能直接调用工具包里面编译好的工具。这些工具的作用就是，搭建一个完整的“离线”辨识系统，以便利用语料库中的语音数据进行建模和实验。我们可以根据这些实验结果，调整实验参数，得到最优的参数组合，以便在工程应用中使用。

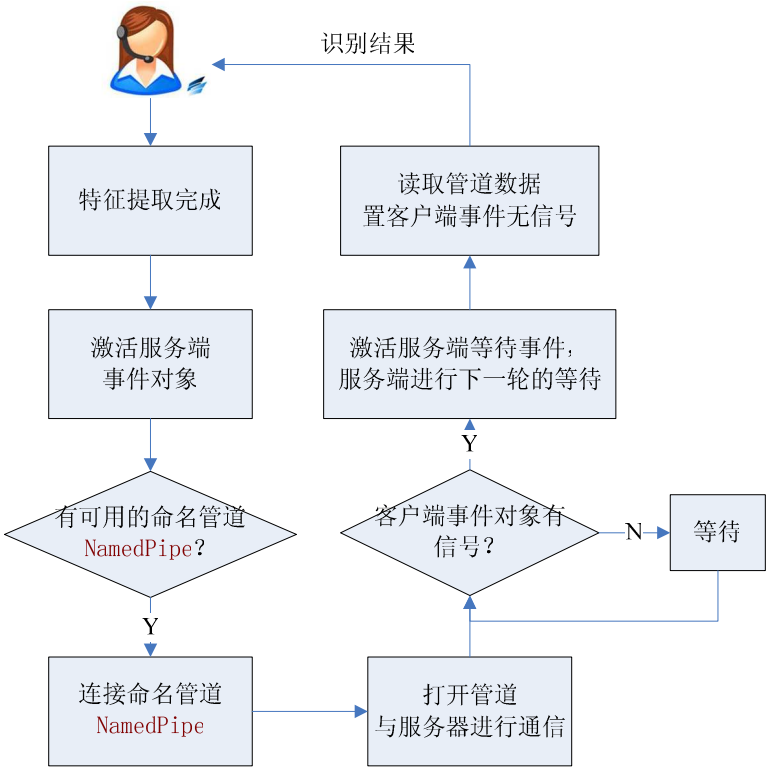
Alize/LIA_RAL 是采用标准 C++语言开发的，一般用于 Microsoft Visual Studio 2005 环境下的 Win32 的控制台应用程序[20]。而本文的系统是 VC6.0 MFC 平台下的项目。VC 在 console 下和标准 C++是兼容的，但是在 MFC 中微软采用了很多自己特有的东西，这些和标准 C++是不兼容的，微软对 MFC 有很多扩展属性，并且不提倡原来的函数声明风格。这样 Alize/LIA_RAL 的源码就不能在 VC 下直接使用。如何能既使用工具包的源码，而又能达到实时辨识的要求呢？

本文采用了进程间的同步与通信技术来解决这个问题。进程间的通信技术主要有内存映射、消息、管道剪贴板、邮槽等。本文采用命名管道来实现进程通信，采用一种基于 C/S（客户端—服务器）架构的思想，让加载模型、判分计算等数据处理部分作为后台的“服务端”，前台界面作为客户端是人机交互界面，两者通过命名管道来传递信息，进程之间采用事件对象来同步。客户端的程序处理流程如图 3-5(a)所示，服务器端的程序处理流程如图 3-5(b)所示。

“服务器”端，首先完成模型的加载部分，然后创建管道，等待客户端的连接，当客户端有请求时，则进行判分识别操作，并将辨识结果通过管道传递给客户端，然后等待客户端的指示。当用户说话结束，客户端完成特征提取后，会给“服务器”端发信号，告诉“服务器”端，可以进行判分计算，而此时客户端处于等待状态，当接收到辨识结果后，则将辨识结果显示给用户，同时给“服务器”端发信号，指示“服务器”端可以进行下一轮的判分识别。



(a)服务器端通信处理流程图



(b)客户端通信处理流程

图 3-5 客户端和“服务器”端进程通信与同步

第四章 声纹考勤系统的设计与实现

说话人识别具有广泛的应用前景，目前国内的一些公司已经开发出一系列基于说话人识别技术的产品，如“得意身份证”。针对于教学管理中人工点名签到工作的复杂繁重、浪费时间和不准确性，我们将说话人辨识应用于考勤中，开发声纹考勤系统，使实验室的管理跟上现代信息社会的步伐。

声纹考勤系统是通过分析语音的特征，从中抽取语音的特征，从而确认个人身份。然后利用计算机对该学生的出勤状况等信息进行登记。另外，系统还能迅速准确的完成出勤的统计计算和汇总工作。系统的主要目的就是帮助实验室的导师提高工作效率，实现学生管理的系统化、规范化和自动化。和以往人工签到考勤相比，声纹考勤系统的主要优势在于：

- 不可伪造性。与传统的手工签到相比，语音指令的可变性和说话人声纹的稳定性组成双重身份认证依据。这样就会极大可能的避免考勤中代签、替签的现象，提高考勤表的真实度和可信度。同时，系统会保存所有考勤记录的语音，以备不时之需。
- 管理便捷性。所有的考勤信息通过数据库技术进行管理，可以方便的查询所有的出勤信息，这样大大节约了管理人员的时间。
- 使用方便、人性化。用户仅用简单的语音便可完成签到，由于人说话与一系列发音器官有关，且与发音的心态有较大关系，这样彰显了人性化运用的特点。
- 识别速度快、适应性广。每次识别的时间不超过 1 秒，达到实时识别的要求，同时说话人不受语种的限制。
- 应用新颖性。给用户树立智能化、现代化的形象。

本章的主要内容安排：首先将介绍基于 Windows APIs 的音频控制，包括录音、放音、WAVE 文件的保存等内容。然后，详细的介绍声纹考勤系统的设计和实现部分。最后将对系统的使用特性给予说明。

4.1 Windows 下麦克风录音模块

为了保证录音接口的统一性，本文所有音频数据的采集都不依赖于第三方录音软件，而是使用由 Waveform Audio APIs 以及 Multimedia File I/O 编写的一个能自由控制的音频控制接口。

4.1.1 数字音频基础

麦克风的录音过程，其实就是通过声卡将输入的模拟信号转化为数字信号的过程。而声卡就是实现声波/数字信号相互转换的硬件。声卡的基本功能是把来自话筒、磁带、录音机的原始声音信号加以转换，输出到耳机、扬声器、扩音机、录音机等声响设备，或通过音乐设备数字接口(MIDI)使乐器发出美妙的声音。

其中，我们使用到一些概念，如下：

- 采样率(Sampling Rate) ：即声卡在一秒之中对声音(波形)作记录的次数，根据研究，声音播出时的质量常常只能达到采样率的一半，因此必须采用双倍的采样率才能将声音标准重现。采样率越高所记录下来的音质就越清晰，当然采样所记录的文件就越大，本文实验采用的采样率为 16KHz。
- 采样位数(Sampling Byte)：可以理解为声卡处理声音的解析度。这个值越大，解析度就越高，录制和回放的声音就越真实。为了较精细的描述声音的特征，本文采样位数为设定为 16 位。
- 量化(Quantization)及量化误差(Quantization error)：量化是将幅值连续的模拟信号转化为幅值离散的离散信号；量化误差是指量化结果和被量化模拟量的差值,显然量化级数越多,量化的相对误差越小。
- 声音强度(Sound Intensity)：是指波形振幅的平方，两个声音强度上的差分，常以分贝(db)为单位来度量，计算公式为：20*log(A1/A2)分贝，其中 A1 和 A2 为两个声音的振幅。

4.1.2 WAVE 文件格式

WAVE 文件作为多媒体中使用的声波文件格式之一，它是以 RIFF 格式为标准的，每个 WAVE 文件的头四个字便是“RIFF”。WAVE 文件由文件头和数据体两大部分组成。其中文件头又分为 RIFF/WAV 文件标识段和声音数据格式说明两部分。RIFF 块包含两个子块，这两个子块的 ID 分别是“fmt ”和“data”，其中“fmt”子块由结构 PCMWAVEFORMAT 所组成，其子块的大小就是 sizeof(PCMWAVEFORMAT)，数据组成就是 PCMWAVEFORMAT 结构中的数据,

实验中，我们采用的 WAVE 的文件头的格式如下图所示。

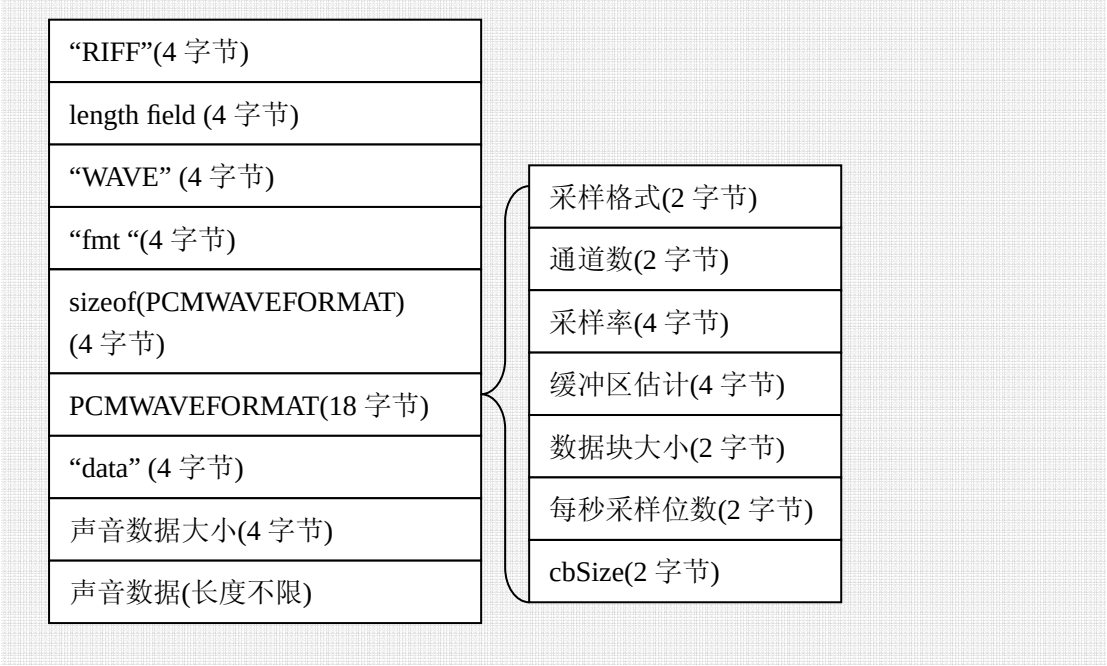
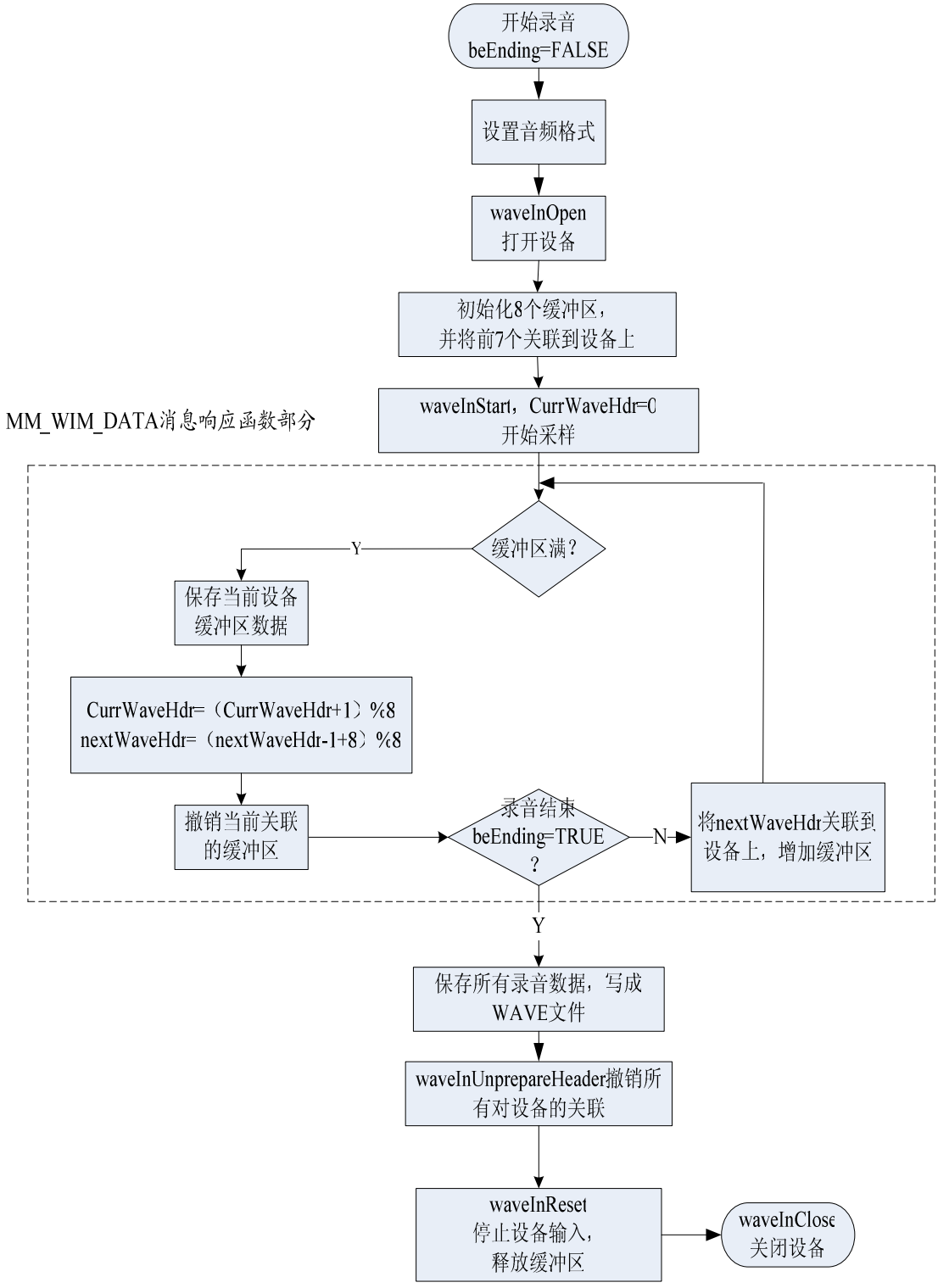


图 4-1 WAVE 文件格式

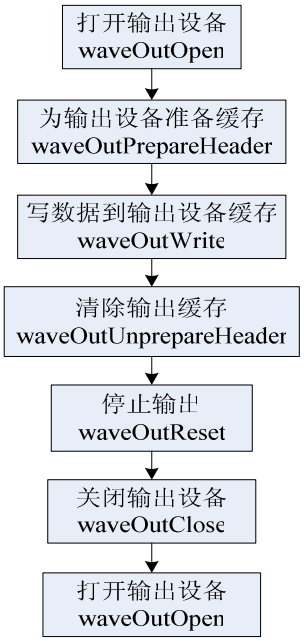
4.1.3 Windows 下的音频控制

Windows 下录音和回放主要有三种模式：通过高级音频函数和媒体控制接口 MCI 设备驱动程序；利用低级音频函数 MIDI Mapper 和低级音频设备驱动(WaveX APIs 见附录Ⅰ)；利用 DirectX 中的 DirectSound。使用 MCI 的方法及其简便，但是灵活性较差，使用低级音频函数的方法相对来说难一点，但是能够灵活的进行操控；而采用 DirectSound 方法，控制声音数据灵活，效果比前两者都好，但是实现起来比较困难。本文根据实际需求采用低级音频函数的方法。

MCI 按照打开设备、配置设备、实现功能、撤销配置、关闭设备的标准组织 APIs。对于录音编程而言，其要点在于根据音频格式打开对应的设备、配置录音所需要的参数、按一定的次序发送命令给设备、接受数据并配置参数继续录音、停止录音释放资源、关闭设备等几个步骤。主要的录音和放音流程如图 4-2。多/双缓冲技术可以很好的实现声音的快速采集和实时顺畅播放，一般录音程序都是采用双缓冲技术，但是会出现录音不连续的问题，所以本文采用 8 缓冲循环的技术，不仅不会丢失数据，而且可以实现无限时长的录音。声音回放的操作就相对简单，采用双缓冲就可以满足要求。


开始录音
beEnding=FALSE
设置音频格式
waveInOpen
打开设备
初始化8个缓冲区，
并将前7个关联到设备上
waveInStart，CurrWaveHdr=0
开始采样
MM_WIM_DATA消息响应函数部分
缓冲区满？
Y
保存当前设备
缓冲区数据
CurrWaveHdr=（CurrWaveHdr+1）%8
nextWaveHdr=（nextWaveHdr-1+8）%8
撤销当前关联
的缓冲区
录音结束
beEnding=TRUE
？
N
将nextWaveHdr关联到
设备上，增加缓冲区
Y
保存所有录音数据，写成
WAVE文件
waveInUnprepareHeader撤销所
有对设备的关联
waveInReset
停止设备输入，
释放缓冲区
waveInClose
关闭设备

(a)录音流程



(b)声音回放主要流程

图 4-2 基于 WaveX APIs 的录音和回放

4.2 系统组成

4.2.1 系统架构

系统采用 C/S 的架构，主要包括音频控制模块、语音预处理模块、声纹注册模块、声纹(说话人)模型加载模块、识别模块、考勤信息管理模块等。系统主要框架如图 4-1 所示。

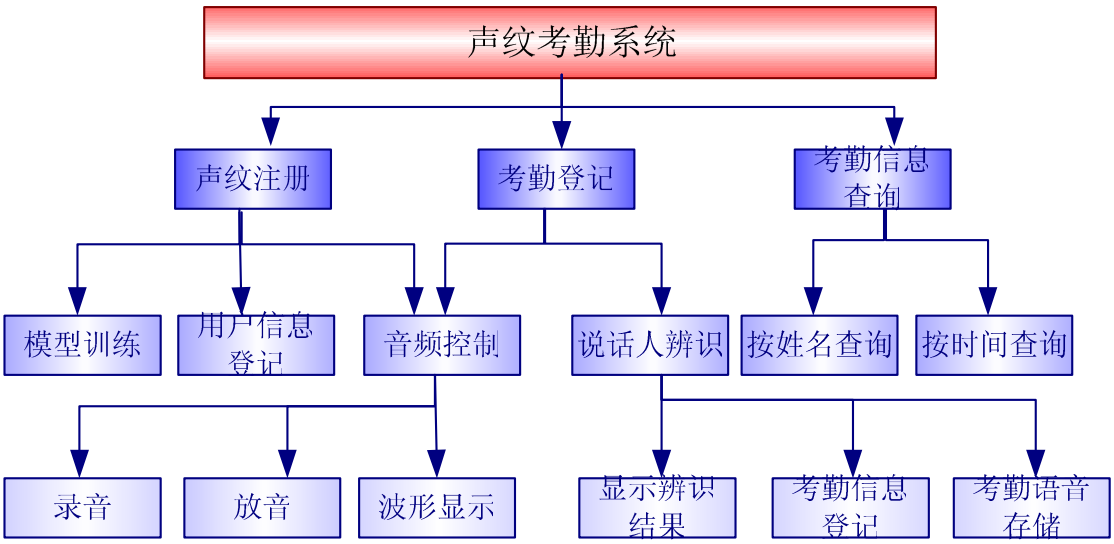


图 4-3 声纹考勤系统功能模块图

(1) 声纹注册模块

这一模块实现的功能主要是对到考勤对象训练说话人模型。如果说话人首次使用系统，则首先注册信息，之后再根据“抄本”(见附录Ⅱ)录音，训练说话人模型，说话人模型的训练可以分为单纯的 GMM 模型训练和自适应模型训练两种方式。

(2) 考勤登记模块

这个模块主要的功能就是根据说话人的语音辨识出身份，如果与用户选择的身份吻合，则将说话人的出勤时间写入到考勤数据库中，否则考勤信息登记失败，用户再次尝试。3 次失败后系统用户本次考勤登记失败，找管理员帮助登记信息。每次通过考勤的语音文件将会被保存到语音数据库中，以便以后实验使用。此模块还有语音提示和播放识别结果的功能，主要是通过 TTS 来完成的。本模块的处理流程如图 4-2 所示。

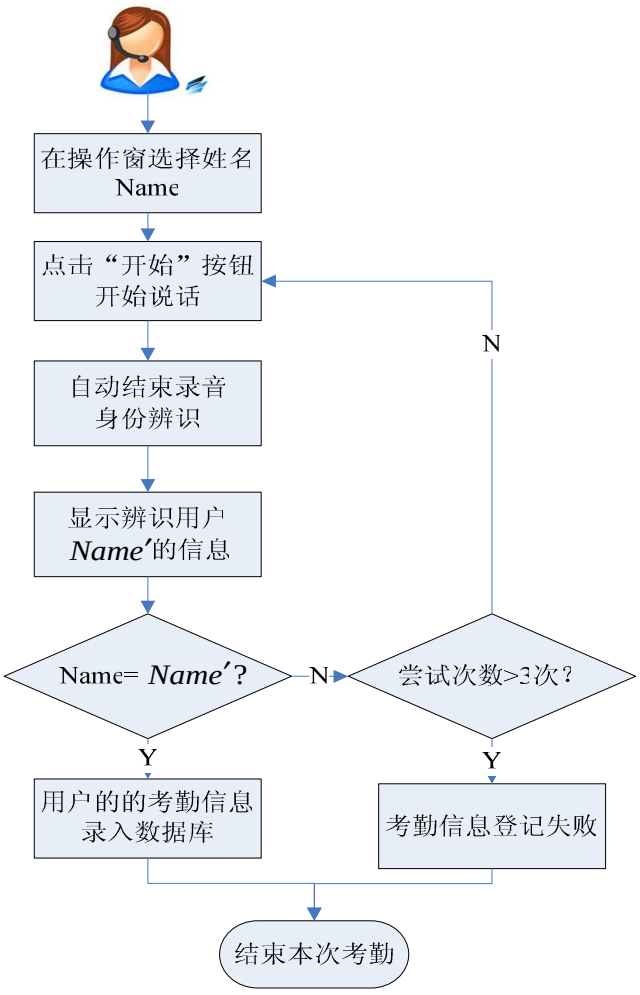


图 4-4 考勤信息登记流程图

(3) 考勤信息查询模块

每个用户的信息和考勤信息将通过 Access 数据库来管理，授权的用户可以查看考勤信息(按照姓名或时间段的条件来查询，也可两者综合查询)，但是一般的用户只有查询的权限，增、删、改的操作是不允许的。

4.2.2 声音采集

为了保持 UBM 训练的音频数据和系统实时测试的音频数据的一致性，本文在前期采集语料库时采用的录音软件也是采用统一的音频控制接口。被采集声音的人员首先登记信息，将登记表上的编号填写到“编号”一栏，如图 4-5 所示。点击“开始录音”就可以根据抄本来说话。录音结束后，点击“保存文件”即可将录音数据保存成 WAVE 格式的文件。

图 4-5 声音采集程序界面

4.2.3 系统主要界面设计和使用说明

本文系统的客户端是一个 VC6.0 的 MFC 工程。基于人性化、界面友好、操作简单的原则来设计系统的界面。图 4-6 展示了系统的主界面。

图 4-6 声纹考勤系统主界面

(1) 声纹考勤

主界面展示的是声纹考勤的功能模块。最左边的树形框是用户名列表，显示了系统现有的模型标识列表。考勤前用户需要选择自己的姓名，然后点击“开始”按钮开始说话。考勤模块通过自有的录音接口来接收语音数据，系统采用能量检测的方法实现简单的端点检测，以便自动检测说话人是否在说话。具体的检测原则如下：

说话时长阈值设置为 T (可调参数，本文系统设置成 T=6s)，说话人有效说话时长为 t0，从点击“开始”到开口说话的时长为 t1，说话结束但总时长没有达到 T 的这段时间记为 t2，系统总的录音时长 t。系统保存的语音文件的有效时长 t 为：

$$t=\begin{cases} =0, & t1>=2 \\ =t1+t0, & t1<2 \ \&\ t2>2 \\ =6, & t1<2 \ \&\ t1+t0>=T-2 \\ =6, & t1+t0>=6 \end{cases}$$

=0, t1>=2 → 重新录音

=t1+t0, t1<2 & t2>2 → 发出录音时长不够警告

=6, t1<2 & t1+t0>=T-2 / =6, t1+t0>=6 → 有效录音，开始辨识身份

当发出有效时长不够的警告时，用户可以选择忽略警告，则系统自动的开始辨识身份，可以想象，用户的总有效时长不过 2~3 秒，辨识率自然会很低。

系统会将辨识的结果在界面的右面显示出来，当左侧用户选择的姓名和声纹辨识出来的姓名相同时，系统才会自动记录考勤信息到考勤表中，同时将考勤语音放到语音文件数据库中。否则，系统将会提示两者不匹配，再次尝试；如果用户有事先注册自己的 GMM 模型，则可以不用再次说话，点击“GMM 确认”按钮，采用另一种模型来识别，此时可能就会识别正确。

(2) 声纹注册

图 4-7(a)是用户的声纹注册部分，首先用户需要注册信息，然后登录进行模型训练。若是选择“训练 GMM 模型”，进入界面如图 4-7(b)：这个部分采用逐句录音的形式，用户的录音工作可以分成几个阶段，每个阶段录音 40 句，每个阶段有一定的时间间隔，当所有阶段的录音工作完成后，系统才会为用户建立模型。若是选择“自适应模型”，进入界面如图 4-7(c)：在 UBM 上自适应模型讲究的是“快速”，所以用户一次性完成录音、训练模型的工作。录音抄本见附录 II 。

(a)声纹注册

(b)训练 GMM 模型　　(c) 根据 UBM 自适应说话人模型

图 4-7 用户声纹注册界面

(3) 记录管理

最后一部分是考勤信息管理模块，用 Access 数据库的 ADO 编程的方法来实现。目前实现的功能是查询模块，日后可以根据实际需要增加更多的信息管理功能，比如学生信息管理和请假管理等。

图 4-8 考勤信息管理界面

在一般情况下，界面默认的是显示考勤表中的所有记录，用户考勤登记完毕后，可以点击“刷新”查看记录项。

4.3 系统使用

任何一个系统只有投入到使用时，才能体现它的价值。目前，声纹考勤系统已经在陕西省语音与图像信息处理重点实验室“音频、语音与语言处理研究组”投入使用。本系统将实验室的所有同学按照年级分成不同的组(大四、研一、研二等)。考勤时，只有用户选择的身份和辨识的身份相同时，当前用户的出勤时间才会被存入到数据库中，否则再次尝试。由于是在实验室内使用，而实验室学生数量比较少，所以，系统的整体识别率较高，用户一般能一次性完成考勤任务。

由前文所述可知，本系统是分成客户端(即系统的用户端)和“服务器”端组成。“服务器”端主要是完成模型加载、模板匹配计算和决策的功能，客户端则是系统与用户的交流区，根据用户的操作给予相应的响应。

使用本系统只需要 PC 机一台、麦克风一个、音响一个(可选)。这是一般实验室都有的设备，因此本系统具有较好的通用性。操作时，首先启动服务端，加载所有说话人模型，加载模型的时间与模型的个数和每个高斯模型的 Distribution 的个数成线性关系；模型加载完毕后就可以进行考勤登记；系统中每增添一个说话人模型，都要重新启动服务端加载模型。

4.4 使用要求

由于人的语音会随着时间的变化而变化，而且会受到健康和感情等因素的影响，所以在使用声纹考勤系统时，需要注意一些使用条件。

4.4.1 对用户的使用要求

任何系统的使用都需要用户的配合，本系统对用户的要求主要有：

(1) 用户要按照训练模型时的讲话方式在麦克风前讲话；

(2) 讲话的时长要保持在 6 秒左右，否则会影响识别效果；

(3) 讲话时，嘴部离话筒的距离保持在 10cm 的距离；使用距离和使用人的说话音量有关，音量大的，距离可适当放宽。

(4) 对说话人的语种无要求，但是其语音内容尽量能体现说话人的声纹特性，若用户一直说“啊……呀呀”，这分明就是对系统的一种挑战！

(5) 系统具有较高的敏感性，当用户故意仿冒或者模他人时；由于模仿音还带有说话人的特有的声纹特性，所以被识别成被模仿者的可能性极低。这样就提高了考勤登记表的可信度。

4.4.2 使用环境要求

由于训练 UBM 模型的数据是在静音实验室的环境下采集的，所以需要在安静的实验室环境下才能发挥出其优秀的性能，但是一般的实验室下的噪音也在系统的容忍范围内；超出应用环境要求的情况下使用，用户可能需要重复多次才能成功登记考勤信息。

下图是一个使用声纹考勤系统进行考勤的图片。

图 4-8 声纹考勤

第五章 实验验证和系统评估

实验中，我们通常会将语料库划分成三个数据集(通常是不能分离的)，分别用来训练 UBM、训练目标说话人模型和测试。在数据不足的情况下，训练 UBM 的数据和训练目标模型的数据是不能有交集的，否则会产生过训练(over-training)的现象。当然，训练数据和测试数据是不允许有交集的。说话人识别实验就是要进行多次的人工测试、改进和分析，所以实验部分就是在不断的重复下面的步骤：

(1) 搭建系统；
(2) 定义系统参数；
(3) 设置参数；
(4) 在开发集上运行系统；
(5) 分析、重新定义参数；
(6) 重复第(4)步；
(7) 用评估集运行系统；
(8) 评估的结果作为系统的评估结果。

本文的实验测评分为两个部分，一部分是在标准语料库上的评估，另一部分是在自录预料库上的测评结果。本文系统的应用对象是实验室，所以我们的实验评估焦点在第二部分。

首先，介绍一些本文的数据库使用安排情况。

5.1 实验准备

本文所涉及到的四个语料库分别是绪论中所提到的 NIST SRE2004 的语料库，CCC 的 CCC-VPR2C2006-10000 语料库，OGI CSLU 中心 2005 年的语音辨识语料库和西北工业大学陕西省语音与图像信息处理重点实验室“音频、语音与语言处理研究组”的语料库(即自建语料库)。其中 NIST 语料库是目前说话人识别研究领域中最权威、使用最多的语料库；最后一个语料库是我们根据实验需求自己采集的语料库。

5.1.1 公用语料库

本文使用标准语料库对 Alize/LIA_RAL 软件包的识别性能做出基本把握，

根据实验结果，再决定能否将其用在本文的说话人辨识系统中。这三个语料库的特性如表 5-1 所示：

表 5-1 语料库简况

名称 / 特性	NIST SRE2004 语料库	CSLU 语料库	CCC 语料库
录音设备	电话	电话	电话和手机
录音抄本	对话	短句、数字、单词、口令、模仿音和对话语音	专业设计的 40 句话(见附录Ⅱ)和少许对话语音
语种	英语为主	英语为主	普通话
录音人员成分	不同国籍(310 人) 男 186，女 124	不同国籍(91 人) 男 46，女 45	中国籍(6000 人) 20 岁左右的男性
音频文件格式	CCIT μ-Law sph 音频格式	PCM 编码 WAVE 文件格式	PCM 编码 WAVE 文件格式
延展性	较好	较好	差
采样率	8KHz	8KHz	8KHz
采样位数	16 位	16 位	16 位
录音环境	家里 办公室	家里 办公室	环境不固定
录音间隔	数天 持续数月	数月 两年内完成 12 次录音	每人两次：电话录音一次、手机录音一次
用途	语音识别领域	用于说话人辨识的研究	说话人识别领域

5.1.2 自建语料库

声纹考勤系统是将说话人辨识系统实际应用于工程中，所以要求说话人识别系统的训练和测试数据要符合工程应用的环境。明显的，上面几个语料库都是针对于电话信道的说话人识别的研究，采样率受到限制、录音环境也与本文系统的使用环境不同。基于以上几个原因，我们需要按照自己的需要采集我们自己的语音数据。

(1) 语料库采集录音方案

语料库是根据现有实验条件和系统工程的实际应用而收集的，由于本系统是在 PC 机上的应用，完全可以采用较高的采样率，将声音描述的更精细，所以本

语料库的采样率设为 16KHz，具体情况如下：

- 时间：2010 年 4 月初至今
- 生源：西北工业大学在校学生，20~30 岁之间，讲普通话；
  录音环境与设备
- 录音地点：计算机学院大楼 静音实验室
- 录音设备：硕美科(Somic) SM-010 麦克风(全指向性)，计算机(瑞昱 ALC662 英特尔 82801G(ICH7)声卡)
- 录音间隔：25 人录音 2 次，间隔 2 个星期；其他人录音一次
- 数据集合：目前数据库共有 86 人的录音数据，录音工作一直持续中……
- 录音抄本：CCC 的 40 句中文语句，几乎囊括的所有中文音素发音
- 录音要求：说普通话，讲话自然、清晰、语速适中，录音时长 2min 左右
- 音频文件格式：WAVE 文件格式、标准 PCM 编码、16KHz 采样率、16 位采样位数

为了下文论述方便，以上含有 40 句固定语音内容的语音称为 U 集。另外，该语料库还包括一些特定人(共 23 人)的短语音数据，这些数据是由不超过 8 秒、语音内容不限、说话方式不限的短句组成，标记为 T 集，主要作为系统测试集。23 个特定人是我们的主要测试对象，设标识分别为 Spk1，Spk2，……，Spk23。

(2) 语料库的存储与标注

建立语料库的最终目的在于方便我们实验使用。为了达到这个目的，我们必须首先对采集到的语音数据库进行后处理。现今我们根据实验需求，将每个说话人的语音切分成 40 个小句子，按类别存储，供以后实验使用。

目前收集的语料库不仅用在本人毕业设计开发的声纹考勤系统中，在另外的两个项目：实时语音驱动的面部动画系统及关键词检索中都发挥了作用，提高了其系统的性能。

(3)本文对语料库的使用分配情况

训练 UBM 的数据集是 U′集，为了避免“过训练”的现象，U′集是 U 集减去 23 个特定人语音数据的数据集。训练 23 个说话人模型数据集是$\{U-U'\}$，T 集为测试集。则训练 UBM 的数据量个数据分配情况如表 5-2 所示。各说话人模型的训练数据量和数据时长的分配情况如表 5-3 所示：

表 5-2 UBM 模型的训练数据分配情况表

女生语音的数据量(MB)	女生的数据时长(s)	男生语音的数据量(MB)	男生数据的时长(s)	总语音数据量(MB)	总时长(s)
44.5	1540	197	6969	241.5	8509

表 5-3 模型的训练数据量和数据时长的分配情况表

Speaker ID	GMM 训练数据量(MB)	GMM 训练数据时长(s)	模型自适应数据量(MB)	模型自适应数据时长(s)	测试文件数
Spk1	6.33	202	3.20	100	99
Spk2	6.88	215	3.25	106	34
Spk3	6.04	236	4.45	145	41
Spk4	6.00	190	3.04	99	40
Spk5	4.95	163	2.61	85	37
Spk6	6.69	218	3.36	110	29
Spk7	6.27	204	3.10	101	33
Spk8	6.57	214	3.25	106	41
Spk9	6.44	210	3.38	110	36
Spk10	5.99	172	2.87	94	30
Spk11	7.89	258	2.84	93	24
Spk12	6.28	237	3.40	111	30
Spk13	7.45	197	3.68	120	31
Spk14	4.85	189	2.50	81	43
Spk15	6.52	204	3.19	104	31
Spk16	5.95	189	2.89	95	33
Spk17	5.60	179	2.54	83	41
Spk18	5.83	195	2.95	97	52
Spk19	5.92	196	2.94	96	28
Spk20	6.50	223	3.48	114	34
Spk21	6.42	209	3.08	101	28
Spk22	7.09	233	3.75	123	50
Spk23	6.26	183	2.99	98	30

注：其中女生五人，分别是 Spk1、Spk14、Spk15、Spk16 和 Spk18。

5.1.3 性能评价指标

对于说话人辨识来说，识别的结果之可能是正确或错误两种情况，且正确识别的概率和错误识别的概率之和为 1，因此可以简单的用正确识别率(常称为正确

率)或者错误识别率(常称为错误率)作为评价识别系统性能的指标。本文使用辨识错误率(Identification Error Rate，IDER)作为系统性能评价的一个重要指标：

$$IDER = \frac{\text{错误识别的次数}}{\text{总的辨识试验次数}} * 100\%$$

对于说话人识别系统的评价，识别率是最重要的评价指标，但是对于实际工程应用，还应将可识别人数、说话方式、训练时间、识别响应时间、噪声处理等多项因素考虑进来。本文对声纹考勤系统的评价也将从以上几个方面进行。

5.2 CSLU 语料库上的实验

根据表 5-1 可知，CSLU 语料库是我们实验的理想语料库，每个人的录音间隔较长、录音次数较多，且每个人有充足的数据以便我们实验，由于本文的主要任务是实现一个系统，而不是与其它的说话人识别系统的性能做比较，所有没有必要使用比较复杂的 NIST 2004 数据库，由于 CCC 数据库全部是男声，所以也不适合实验。

在 CSLU 数据库上的实验，主要是测试 Alize/LIA_RAL 包的说话人识别性能，并研究了高斯混合中分布的个数对系统识别性能的影响。实验采用 81 个人的音频文件为训练数据(男 41 人，女 40 人)，训练的音频文件数 91699；另外 10 个人(男女各 5 个)的音频文件都分成训练集和测试集两个部分，每个人的模型自适应的实验数据分配如表 5-4 所示。

表 5-4 实验参数设置

说话人模型自适应训练次数	音频文件个数	有效词个数	音频总时长	测试音频文件总数
5	30	约 100 个	45s	10989

实验结果如图 5-1 所示，系统的整体识别错误率在 7%以上，这与当前的说话人识别水平还有较大差距，与 Alize 在 NIST SRE2004 年的测评结果也有很大的差异。经过分析可知，这主要跟 CSLU 语料库的特性有关系。一方面，由于测试数据是以文件为单位的，而每个音频文件的有效语音数据是不定的，有些文件只有一个单词的有效时长，这样的测试数据很难反映说话人的特性，系统的性能自然就不会高。另一方面，在实验中我们没有采用特征归一化和判分规整等技术。若是我们忽略这些因素，可以认为 Alize/LIA_RAL 的识别性能，足以满足我们声纹考勤系统的识别率高的要求。

同时，从这组实验中还可以看出，随着高斯模型分布个数的增加，系统的识别效率不断提高，但是提高的幅度会不断减少。

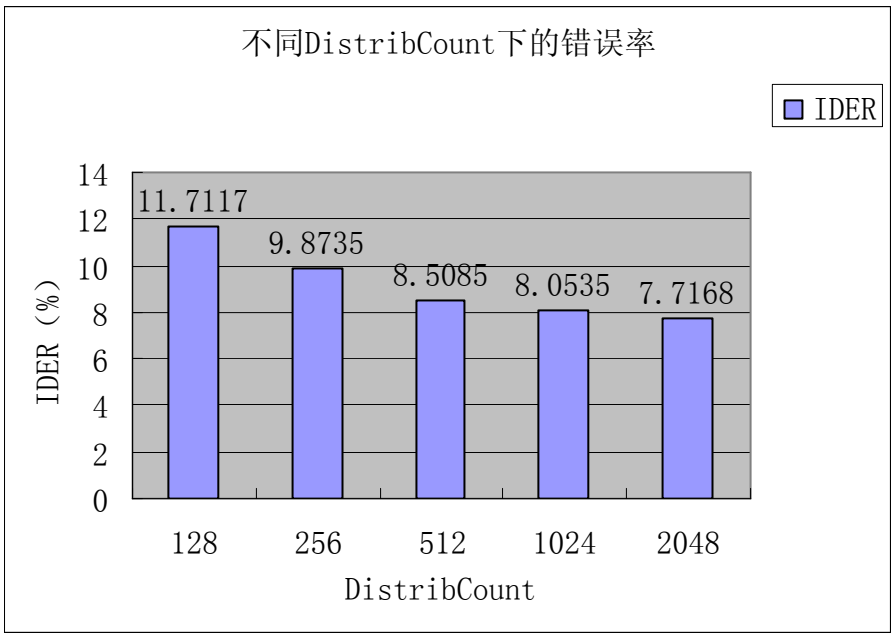


图 5-1 不同高斯分布情况下的错误率

5.3 自建语料库上的实验

实验的目的主要是从系统架构的角度，研究 GMM 识别中高斯混合模型的分布个数、训练次数、归一化技术等对系统识别率的影响。另外实验还比较了 GMM 说话人辨识系统和 GMM-UBM 说话人辨识系统的识别性能，以便在应用工程中设置最优的参数组合，让系统以最高的识别率、最快的建模和响应速度展现给用户。此外，针对于模仿的现象本文也做了一组实验，以证明系统具有较高的敏感性。

以下的实验中，都采用如下相同的实验环境：

系统中说话人模型个数为 23 个，测试语句以文件为单位，每个文件都是这 23 个人之中的一个人的语音文件，测试语句的长度小于 6 秒，说话方式不限、语种不限、语音内容不限。

5.3.1 GMM 模型混合阶数对应用系统的影响

根据实际工程应用中说话人辨识系统的识别性能高、训练速度快、训练时间短和空间代价小这几个要求，本文进行下面的一组统计实验，研究不同高斯分布

情况下，系统的性能，实验结果如表 5-5 所示。

表 5-5 不同高斯分布个数的实验结果

高斯分布个数	错误率 IDER (%)	每个说话人模型自适应训练时间(s)	每个说话人模型所占空间(KB)
128	10.40	43	443
256	9.1429	85	886
512	8.2286	170	1771
1024	7.4286	349	3543
2048	6.6286	698	7090

经过实验分析，得到如下结论：

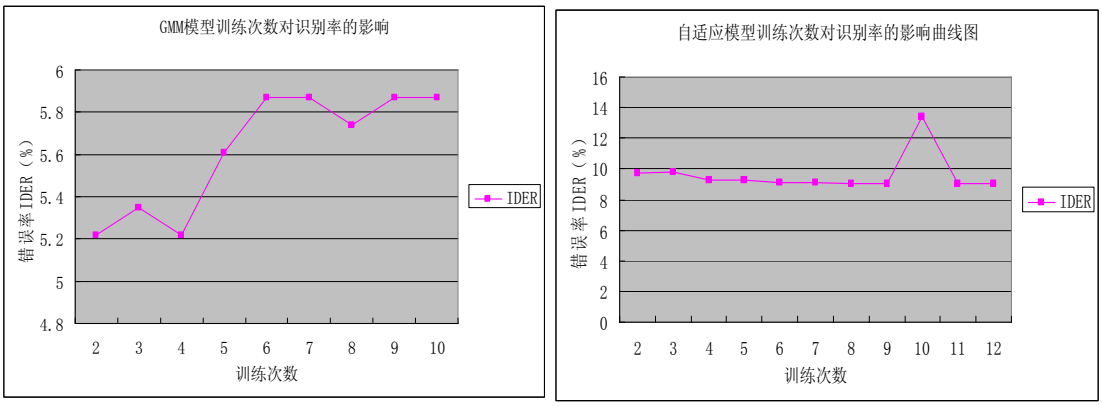
- 在高斯混合度较小时，每阶含有大量差异性较大的数据，模型与实际的说话人模型差异很大，不能很好的反应说话人的个性特征，这时系统的识别率较低。
- 当高斯混合度比较大时，模型表示说话人个性特征的误差比较小，系统识别率就会比较高。理论上，在混合度过大时，由于训练语音长度的不足，在每阶的数据量较少，每个高斯模型的估计是由很少的数据统计得到，这样的模型比较尖锐，不能很好的反应说话人的个性特征。由于本文实验的训练数据较多，所以当分布数增加到 2048 个时，也没有出现这样的实验结果。
- 另外，随着混合度的增加，训练模型的时间和模型所占空间均呈线性增加。高斯混合度较大时，模型训练耗时过长，模型存储空间较大，如果应用于工程中，会给系统带来一定的负担。
- 混合度在 256 以后，倍数提高模型分布个数后，系统的识别率提高的效果就不太显著(不到一个百分点)，但是时间和空间的耗费却倍数增加。

根据以上的结论，综合系统在识别率、时间和空间上的需求，本文声纹考勤系统采用的 256 阶的高斯混合模型，在以下的对比实验中，如果没有特别说明，高斯混合分布的个数 DistribCount 取值 256。

5.3.2 训练次数对模型的影响

由于 GMM 模型的统计特性，在训练数据训练说话人模型时，存在欠训练(under-training)和过训练(over-training)的现象。前者得到的模型不能充分体现说话人的语音特性，而后者使模型与训练数据过分的拟合，这两种情况都会降低系

统的识别效率。本组实验是在所有其他参数都相同的情况下的一组实验，目的是得到最优的训练次数。GMM 说话人辨识系统实验结果如图 5-2(a),GMM-UBM 说话人辨识系统的实验结果如图 5-2(b)所示。



(a) (b)

图 5-2 训练次数对识别率的影响

实验表明，无论是基于 GMM 的说话人辨识还是基于 GMM-UBM 的说话人辨识，都存在上述欠训练和过训练的现象，相对于 GMM 模型，GMM-UBM 模型表现的不明显。在本文语料库上，我们得到最优的训练次数：GMM 模型训练次数：

$$nbTrainIt_{GMM} = 4 \tag{5-1}$$

GMM-UBM 自适应模型训练的次数：

$$nbTrainIt_{GMM-UBM} = 5 \tag{5-2}$$

在本文以后的实验中，如无特别说明，说话人模型训练次数则按式 5-1 和 5-2 取值。

5.3.3 训练数据量对系统的影响

GMM-UBM 说话人辨识系统的优势在于它能根据较少的训练数据、较快的得到说话人模型，这就弥补了 GMM 模型训练数据不足而识别率低的问题。但是训练 UBM 模型却需要大量的、均衡的实验数据，否则训练出的 UBM 不具有普遍性，或者模型偏向于训练数据多的子集。由于语料库收集进展较慢，得不到充分的数据。所以本文实验的 UBM 只有 60 多个人的训练数据，而且男女比例失衡，出现了上面的两个问题是预料之中的事情。解决办法之一就是在本文所述的前三个语料库上训练出UBM'，然后再用自建语料库中的数据做简单自适应，得

到的模型作为系统的 UBM ；但是，实验结果显示这种方法不可行。经分析可知，前三个库与自建语料库存在信道、采样率、录音环境、男女比例、语种等差异，这就导致模型不能很好的显示当前自建语音数据库下的语音共性。另一种方法，就是继续扩充语料库。

对于 GMM 系统，由于不需要 UBM，但是它需要每个说话人要有充足的训练数据，例如采用说话人的 40 句话为其建立模型，错误率为 6.65%，当采用说话人的 2 次录音的 2 个 40 句话的音频数据建模时，错误率降到 5.21%。GMM 模型的训练数据较多时，GMM 系统有它相应的优势。

综合以上两个原因，本文声纹考勤系统中依然保留两种模型训练方法(即直接训练模型和自适应模型)和两种识别方案。

5.3.4 系统对模仿音的敏感性实验

声纹考勤系统不同于以往人工考勤的一个鲜明的特点就是考勤表的真实可靠性，企图通过模仿他人的声音来达到顶替签到的可能性是比较低的。为了测试系统对模仿音的敏感性，本文选取大家很熟悉的赵本山的声音为模仿对象。测试前，反复播放赵本山说的一句话，让受试者熟悉赵本山的说话方式；然后，让受试者模仿赵本山的说话方式说这句话。本次测试的对象是 11 个男生，标识为 $Spk_1, Spk_2, \cdots, Spk_{11}$。本次实验的结果如下：

如图 5-3 所示，绿色线条(imitate)是模仿音在 11 个说话人模型上的似然分分布情况。可知，当说话人按正常的方式讲话时，测试语音在所有说话人模型上的对数似然分数将保持在稳定值左右；但刻意模仿时，测试语音的对数似然率普遍下降。这说明模仿音与说话人正常的声音已有所不同，至于能不能模仿成功呢？

图 5-4 列出了其中 6 个测试者的实验结果。由实验结果可知：说话人 Spk_i 的模仿音在本人的模型得分远大于在赵本山模型上的得分。这说明，即使人耳听出模仿的声音很像，但系统仍能捕捉到说话人本文的声纹特性，区分出是本人的声音还是模仿他人的声音。

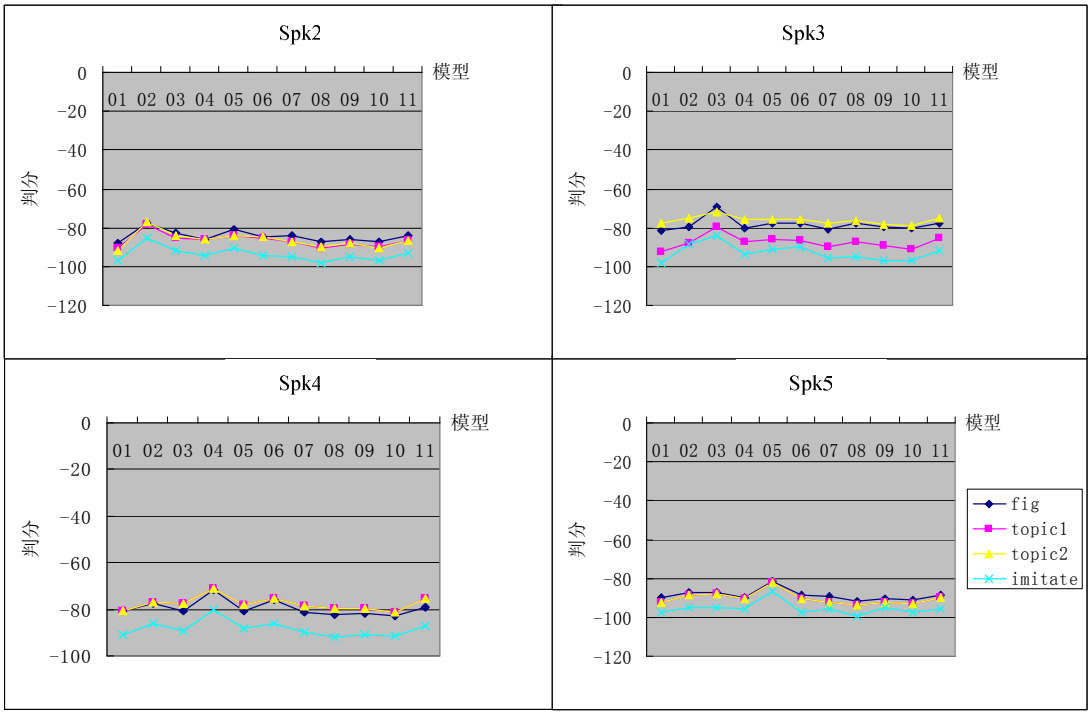


图 5-3 正常说话与故意模仿的似然分对比图

(fig 表示说话人说的是数字串；topic1 和 topic2 都表示正常交谈语句；imitate 表示模仿音)

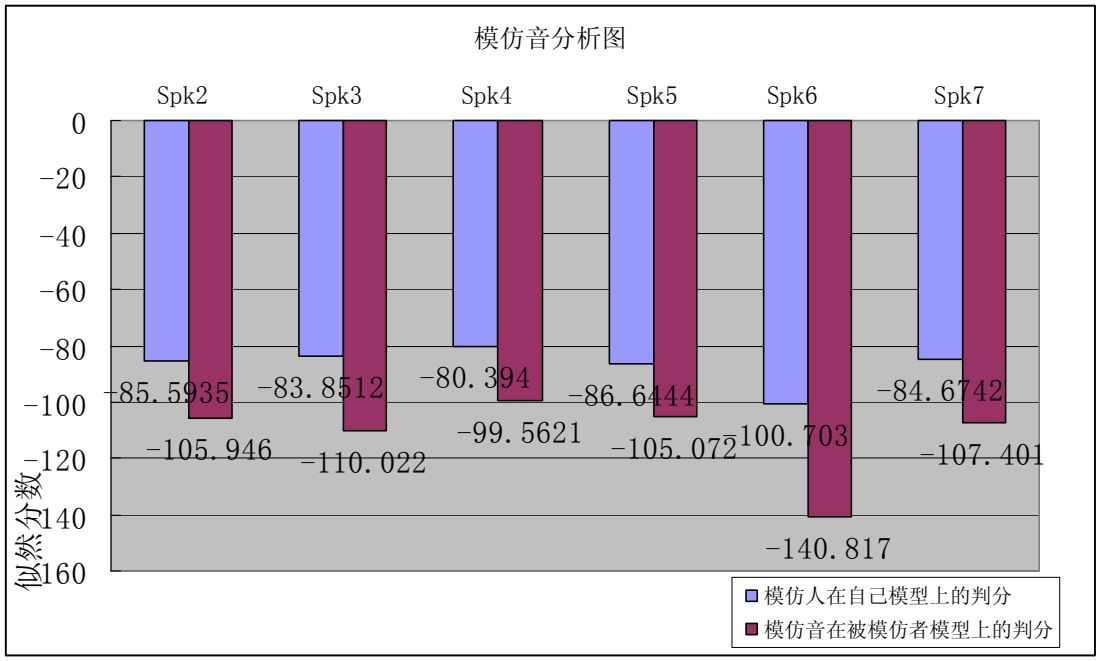


图 5-4 模仿音在说话人模型和被模仿者模型的分数直方图

5.3.5 系统抗噪性能测试

由于说话人模型训练和测试都是在静音实验室进行的，而实际应用中系统性能必然到背景噪声的影响，抗噪性是系统性能的一个重要的评价标准。静音实验室环境下的声音可以看成是没有任何噪声的数据，下面我们对 Spk_1(女)和 Spk_6(男)的纯净的测试数据做了人工加噪(白噪声)处理，得出在不同信噪比下IDER，如表 5-6 所示。

表 5-6 不同信噪比下的识别性能

信噪比 SNR(db)	测试句子数	Spk_1 错误率 IDER(%)	Spk_6 错误率 IDER(%)
纯净语音	50	0.00	0.00
31	50	0.00	58.00
23	50	50.00	80.00

由上表可知：系统具有一定的抗噪功能，但在在信噪比较低的时候，系统的性能会急剧下降。实际应用中的背景噪音不可能只是白噪声，系统的抗噪音功能只有经过长时间的测试和检测才能得出结论。 Spk_1 和 Spk_6 测试的测试结果有一定的差距，这跟男女声音中加入白噪后的声音特性的改变不同有关。

5.3.6 实验总结

本小节主要从模型阶数、训练次数、建模方式、环境噪音、模仿说话这几个方面做了一系列的实验，并通过不断的实验，找到适合本文系统的最佳模型参数组合。综合以上的实验结果，我们在声纹识别系统中使用的一些参数如表 5-6 所示：

表 5-6 声纹考勤系统相关参数设置

参数	基于 GMM-UBM 的说话人模型	基于 GMM 的说话人模型
高斯分布个数	256	256
训练次数	5	4
模型训练语句	40 句话	至少 80 句话
建模录音次数	1 次	至少 2 次(不同时期)
录音形式	一次性录完 40 句话	逐句录音、中间可暂停

5.4 声纹考勤系统测试

前文说过本文描述的系统既可以看成是“开集”的，也可以看成是“闭集”的，由于系统允许新用户注册声纹数据，成为被考勤对象之一，所以从这方面讲这是个“开集”的声纹考勤系统。但是系统在实验室使用，而实验室的学生变动很小，所以一定程度上，可以认为说话人集合是“闭集”的。下面的软件综合测试是以“闭集”的系统测试方法进行测试的。目前系统还处于α测试阶段，即只在开发实验室内部测试，主要是检查系统是否存在缺陷、错误以及功能是否完善等。目前参加测试人数是陕西省语音与图像信息处理重点实验室“音频、语音与语言处理组”的全体同学，一共 23 人。

5.4.1 静音实验室环境下的测试

5.3 节的实验数据都是静音实验室的录音数据，所以在不考虑系统的抗噪性的前提下，测试系统的实时识别率、响应速度、考勤信息管理情况等。下面是受试者的测试要求：

- 按照训练模型的说话方式测试，音量适中，离话筒 10cm 左右，说话声音大的同学可以距离远些。
- 测试语句保持在 6 秒左右，测试分两次，每次测试语句 25 句以上
- 语种不限，可以是普通话、英语、日语、方言等。
- 测试环境为静音实验室。
- 测试分两次完成，两次测试时间间隔为一周左右，且两次测试不能在同一时间段(分为上午、下午和晚上三个时间段)。

经过测试，用户信息注册、声纹模型训练、考勤信息登记、考勤记录管理等模块功能完整，自适应模型训练时间保持在 1 分钟之内，系统响应时间在 1 秒左右，系统的识别率如下：

表 5-7 声纹考勤系统的识别率统计表

Speaker ID	测试句子总数	GMM-UBM 模型系统 IDER(%)	GMM 系统 IDER(%)
Spk1(女)	60	3.33	0.00
Spk2	60	5.00	1.67
Spk3	60	3.33	6.67
Spk4	60	0.00	0.00
Spk5	60	3.33	0.00

Spk6	70	1.43	0.00
Spk7	60	5.00	0.00
Spk8	60	1.67	3.33
Spk9	60	3.33	0.00
Spk10	60	0.00	0.00
Spk11	90	8.89	0.00
Spk12	60	0.00	0.00
Spk13	60	0.00	0.00
Spk14(女)	65	0.00	0.00
Spk15(女)	58	3.45	0.00
Spk16(女)	60	3.33	0.00
Spk17	60	0.00	3.33
Spk18(女)	60	6.67	0.00
Spk19	60	0.00	0.00
Spk20	60	6.67	0.00
Spk21	60	0.00	0.00
Spk22	60	0.00	0.00
Spk23	60	1.67	1.63
合计	1303	2.84	0.77

静音实验室下的实验结果表明系统有相当好的识别性能，由于 GMM-UBM 系统中，UBM 模型的训练数据量较少，造成 UBM 不足以代表所有说话人的共有特性，这是 GMM-UBM 性能低于 GMM 的主要原由。另外测试时发现，来自四川省的同学容易被误识成另一个来自四川的同学，并且两个说话方式有些许相似的同学也容易被误识成对方。这要是由于他们的声音特性比较相似，模型间的差距较小。

5.4.2 声纹考勤系统的试用

到本篇论文完成之前，该系统已经在实验室运行半个多月，虽然识别效果没有在静音实验室的好，但是已经达到了预期的效果。

考勤系统要求说话人首先选择自己的身份，然后再说话，待系统辨识身份，只有两者吻合，考勤信息才登记到考勤表中。这样，代签、替签的成功率极低。声纹考勤系统的运行起到了一种自动督促学生学习的作用，无形中成为指导老师

教学管理的一个小助手。

据考察，正常情况下，大多数同学只需要说一句话便可考勤成功，只有极少数人要重复两遍或以上。这些极少数的同学就是前文所说的来自四川的(如 Spk_{13} 和 Spk_{23})和说话方式有些相似的(如 Spk_3 和 Spk_{20})同学。

根据系统的使用情况，我们做了问卷调查，调查主要从模型训练时长的忍受程度、系统响应速度、对说话方式的限制、抗仿冒性、替签成功率、识别正确率、抗噪性、对学习的影响等 10 个方面进行，参加调查的同学一共 20 人，综合调查的结果如下：

(1) 训练模型需要 50s 左右的时间，60%的同学认为这个时长一般，有少部分人认为较短，另外一小部分认为较长，但能忍受。

(2) 90%的同学认为系统的响应速度很快或较快；另外 10%的同学认为响应速度一般。

(3) 在系统对说话方式的限制方面，大多数同学认为对说话方式的限制较少；在处理噪声方面，普遍反映，有点噪音不影响识别效果。

(4) 对于抗仿冒性和替签成功率这两个方面，一些同学通过自己的实验得出，系统抗仿冒性较好，替签成功率也很低，几乎不能提别人签到。但是也有同学认为系统的替签成功率较高，多试几次就行。经过分析可知，认为替签成功率较高的同学就是在测试时系统容易混淆的人，他们已经知道系统会把他们误识成谁，所以较易替被误识的人签到。

(5) 60%的人认为系统的识别正确率很高，只要说一次就能成功登记考勤信息，但 35%的人认为识别率较好，重复一次就行。

(6) 在 GMM 模型和自适应的模型的比较上，还是大部分认为 GMM 模型较好，这与本文的实验统计结果也是相吻合的。

(7) 对于影响方面，除了 60%的人认为没有影响外，30%的人认为考勤系统能督促他们早上不睡懒觉，另外 10%的人认为本系统的使用在一定程度上提高了他们学习的积极性。

以上统计结果显示：声纹考勤系统的性能基本上达到了实际使用的需求，并且对实验室中 40%的同学的学习起到了积极的影响。他们对系统的总体评价为：感觉很新鲜，挺有意思，督促大家按时到教研室，很实用。

他们还对声纹考勤系统提出了一些很好的意见和建议：希望识别率能更高一些，更好的人性化设计，考勤系统上再增加一些提醒功能，做成全屏界面，加入语音识别和关键词检出功能等。这些建议为我的后期工作提出了努力的方向。

5.4.3 存在的问题分析

系统的识别率对说话人的说话方式依赖性很大，若是说话人能照训练模型的说话方式说话，系统的性能必然能达到最佳。但是，由于人的语音会随着时间的变化而变化，而且会受到健康和感情等因素的影响，所以随着训练时间与使用时间间隔的加长，系统的性能肯定会有所下降。为了维持系统性能，一种解决办法是在训练时所取的语音样本来自不同的时间，比如相隔几天或几周。这样加长训练时间往往难以做到，因为很难要求用户这样安排。另一种解决办法就是在使用过程中不断更新参考模型，比如说，在每次成功的识别以后，即把当时说话人的语音提取到的特征按一定的比例加入到原来的参考模版中去，以保证对使用者说话状态的跟踪。

另外，在系统中，随着说话人数目的增多，集合中的两人和两人以上的说话人的数据分布互相非常接近的可能性也变大。因此，系统中的说话人数目不能无限增大，需要有一定的限制。

5.5 实验小结

本章是整个毕业设计中最重要的一部分，也是最繁琐的一部分。需要通过不断的重复实验，以期得到一组合理的参数，应用于声纹考勤系统中，使其的综合性能达到最优。

明显的，GMM 模型阶数越高，GMM 模型的声学分辨率就越高，能模拟的分布就越复杂，但是太高的阶数，无论从 GMM 参数的估计处理上，还是从说话人模型的存储容量上都是不易接受的。因此，需要通过实验找出能保证声纹考勤系统性能的较小的 GMM 阶数，实验后，系统选取的阶数为 256。

用 EM 算法进行模型训练，从对训练过程的跟踪可以看到，EM 算法在 3~6 次迭代后产生训练特征的概率最大，系统采用的迭代次数分别如公式 5-1 和 5-2 所示。

由于 GMM-UBM 方案需要大量的数据作为 UBM 的训练数据，当训练数据不足时，会给系统的识别性能带来影响；我们在系统中仍然保留 GMM 方案的训练模型方法，是因为在实验室环境下，每个人多次录音是条件允许的。

无噪声环境下训练在有噪声环境下识别时，由于噪声的引入，识别性能有所下降。因此，噪声环境下的说话人辨识是声纹考勤系统面向实际应用的一大挑战。

由于人说话，不仅受发音器官影响，跟环境、感情、时间等有很大的关系，而训练模型的语音数据不可能包括了所有因素下的语音，这就要求说话人尽量按照训练模型时的状态说话，这时，系统才能发挥出其理想的性能。因此，模型能

对说话人个性变化具有一定的补偿能力，也是考勤系统面临的一个实际问题。

第六章 总结与展望

6.1 本文工作总结

本文主要是利用 Alize/LIA_RAL 软件包搭建说话人辨识系统，包括音频控制、说话人声纹注册、身份识别、界面设计等。针对于实际需求，开发了一个简单的声纹考勤系统，以声音作为身份的标识对用户进行出勤信息的登记，并且能对信息进行简单的管理，服务于实验室指导老师的教学管理工作。

本文所做的工作总结如下：

(1) 利用 Windows APIs 开发麦克风录音系统，实现音频控制接口。具有录音、声音回放、数据存储等功能，为后期语音库采集和说话人辨识系统提供了统一的音频控制接口。

(2) 采集了实验语料库，并对采集到的语音数据进行切分和分类处理，为后期的说话人辨识模块的模型训练和测试实验部分提供了重要的数据来源。

(3) 基于 Alize/LIA_RAL 工具包设计与实现了一个基于 GMM 模型的说话人辨识模块。该模块作为应用系统的一个重要的模块，被设计成两种建模方案。一种是用说话人的所有训练数据之间训练说话人模型，另一种方案是在 UBM 模型的基础上自适应出说话人模型。这两种方案在应用工程中都有使用。对于说话人，可以根据实际需求训练自己的模型。

(4) 在已有的说话人辨识模块的基础上，使用 VC++实现了一个声纹考勤系统。具有新用户信息注册于模型训练、声纹考勤信息登记、考勤记录查询等功能。该系统采用 C/S 架构，客户端是用户操作区，界面友好，用户操作方便，“服务器”端完成耗时的模型加载和判分识别的任务。

(5) 采用 CSLU 语料库和自建语料库对说话人识别模块进行了实验测试。研究了高斯混合阶数、训练次数、训练数据量、模仿音等因素对识别性能的影响。根据实验结果将最优的参数组合应用于声纹考勤系统中。

(6) 声纹考勤系统评估。主要是对已经在 α 测试阶段的系统进行评估，分析

问题原因，并提出了一些改进的方法，确定了后续工作的内容。

6.2 本文工作的不足和改进方向

本文设计与实现的说话人辨识模块在实验中虽然得到了较高的识别率，将其应用于声纹考勤系统时，在静音实验室中也有出色的表现。但是，当在实验室(普通实验室)中投入使用时，存在着明显的不足，主要有：

(1) 系统抗噪性有待提高。本文收集的语料库是没有噪音(线路噪音除外)的语音数据，训练模型、参数配置、测试等工作皆是以这些数据为研究对象，基本没有对背景噪声的有效处理办法。另外，实验时，我们采用同一型号的麦克风录音。这样，系统搭建的过程中，就几乎忽略了环境和信道影响。而在实际应用中，这两个被忽略的因素恰恰是影响系统性能的关键因素。所以，以后系统可以从这两个方面继续改进，以提高系统的鲁棒性。

(2) 数据预处理和识别策略有待改进。本文声纹考勤系统中，建模的语音数据并没有进行去噪、删除静音区、特征归一化等处理。根据以往研究可以，这些处理都可以提高系统的识别率。另外，系统的判分识别策略是将测试语音在所有说话人模型上得到的似然分数进行比较，得分最大的模型对应的说话人标识作为辨识的结果。实际上，说话人有男女之分，男女之间的声纹特性有明显的差异，所以可将说话人模型分成两个子集，一个子集对应所有的男生说话人模型，另一个则对应所有女生模型；识别时，首先进行性别识别，然后根据结果找到对应的模型子集，再在这个子集上做判分辨识，由于男女声音有着明显的差异，所以这种策略能提高的系统识别率和响应速度。

(3) 对说话人的说话方式有较高的要求。说话人模型体现的是训练数据的声纹特性，不仅包括先天性特征，还包括后天性特征。系统要达到较高的识别率就要求说话人尽量按照训练模型时的讲话方式说话。先天性特征是不变的，但后天性特征与环境、情绪、健康状态有密切的关系，具有长时变动特性。说话人很难再按照训练模型的说话方式说话。

(4) 声纹考勤系统的考勤信息管理模型功能不够完善。目前考勤信息管理模块只实现了考勤记录的查询功能，另外还可以增加请假、销假管理功能，设置用户操作权限等。

鉴于上面的不足，后续的改进方向主要有：

(1) 增加系统处理噪声的能力。简单的方法是在说话人特征提取之前就将噪声滤掉，得到的语音就好像是在无噪声环境下发出的，然后进行特征的

提取和无噪声环境下的说话人识别。另一种方法就是，同样用 GMM 模型来描述噪声特征在特征空间的统计分布情况，将说话人模型与噪声特征模型有机结合起来，形成集成 GMM 模型，从而很好的解决噪声环境下的说话人识别问题。

(2) 增加评分规整技术。评分规整是说话人识别中用于系统判决的一种常用的技术方法，可以很好的反映测试语音与说话人模型的相对似然度。目前可以采用 ZNorm 改进辨识性能，采用 HNorm 减少信道失配的影响。

(3) 扩充语料库。继续语料的收集工作，保持男女数据的均衡性，训练一个“更大”的 UBM。

(4) 继续完善声纹考勤系统的功能。在使用中，根据实际需要增添新的功能。另外，可以尝试采用 SVM 方法进行说话人识别。

参考文献

[1]. TOMMIE GANNERT. A Speaker Verification System Under The Scope: Alize. March 24, 2007.

[2]. 赵力. 语音信号处理(第二版).北京: 机械工业出版社，2009，P236-255.

[3]. Pruzansky S. Pattern-matching procedure for automatic talker recognition. Journal of the Acoustical Society of America. 1963, 35(3):354-358.

[4]. http://www.360baogao.com/2009-04. 2009-2012 年中国生物识别技术产业发展前景与投资战略咨询报告. 报告编号 199830，2009 年 4 月.

[5]. http://www.nist.gov/speech/tests/spk/index.htm.

[6]. http:// www.think.com.cn/doc/TSIE-intro-chn.pdf.

[7]. Reynolds D. A. Experimental evaluation of features for robust speaker identification[J].IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, October 1994, 2(4):639-644.

[8]. Alex Solomonoff, W. M. Campbell and Ian Boardman. Advances in channel compensation for SVM speaker recognition. ICASSP 2005, 1: 629-632.

[9]. http://mistral.univ-avignon.fr/en/index.html.

[10]. Campbell J P. Speaker Recognition: A tutorial. Processing of the IEEE,1997, 85(9).

[11]. http://www.sinobiometrics.com/Chinese/voice.htm.

[12]. http://www.d-ear.com/Technologies&Products/Products-d-Ear%20ID_ch.htm.

[13]. 吴朝晖，杨莹春. 说话人识别模型与方法. 北京：清华大学出版社，2009，P18-22.

[14]. V. Tyagi and C. Wellekens (2005). On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition, in Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP ’05). IEEE International Conference on, vol. 1, pp. 529–532.

[15]. D. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” Speech Commun., VOL. 17, pp. 91–108, 1995.

[16]. Reynolds, D. A., Quatieri, T. F. Dunn, R. B., Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, 2000.

[17]. Resenberg, T. and Furui, S. Likelihood normalization for speaker verification using a phoneme and speaker-indenpdent model, Speech Commun. 17(1995), 109-116.

[18]. Isobe, T. and Takahashi, J. Text-independent speaker verification using virtual speaker based cohort normalization. In Proceedings of the European Conference on Speech Communication and Technology, 1999, pp. 987–990.

[19]. J. L, Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. Speech Audio Process. Vol.2, pp 291-298, 1994.

[20]. http://mistral.univ-avignon.fr/wiki/index.php/Frequently_asked_questions.

[21]. 孙鑫，余安萍. VC++深入详解. 北京：电子工业大学出版社,2007.

[22]. 明日科技. Visual C++程序开发范例宝典. 北京：人民邮电出版社，2007.

致 谢

沿路走来，毕设之路上帮助我的人太多了，在此，我要向所有关心和帮助我的人表示真挚的感谢！

首先，我要感谢我的导师，谢磊教授，这次毕业设计在研究过程中得到了谢老师的认真指导，他经常询问研究进展，并为我指点迷津，帮我开拓思路、精心点播。谢老师一丝不苟的作风、严谨求实的态度是我学习的榜样，他不仅在学习上对我严格要求，在日常的生活中也给予我非常多的帮助和意见。其中，最让我感动的就是谢老师对我的鼓励。在我保研复试成绩很糟糕时，我很灰心，谢老师主动安慰我，给了我极大的鼓舞，这让我重拾上研的信心；在毕设过程中遇到难题一筹莫展的时候，亦是谢老师的细心指导和不断鼓励，让我相信自己是可以跨过难关的。真正的感激之情是难以用语言来表达的，再次向谢老师致以诚挚的谢意！感谢付中华老师，是他教给我一些处理语音信号的基础知识，提供给我语料库，帮助我拓宽研究的思路。我很喜欢跟付老师交流，虽然我信号方面的基础知识薄弱，但是每次他都耐心的给我讲解，并在一些细节方面给了很多好的意见。

我还要对语音组所有的同学表示感谢。在我录制语料库和实验测试的时候，没有你们积极的配合，就没有我这份答卷。特别的，感谢吴鹏师兄教我 VC++入门，并在音频控制编程方面给了我很大了帮助；谢谢赵文淮师兄不厌其烦的帮我调试程序；感谢卢咪咪师姐在 Alize 工具包使用方面对我的指导，感谢郑李磊师兄在生活上和学习上对我的帮助；感谢周祥增同学帮我处理一些计算机方面的问题。另外还要谢谢一起做毕设的李冰锋、孙乃才、田上萱、张健同学，我们一起度过了毕设期间这紧张、充实而又充满挑战的岁月。

感谢我可爱的舍友们，是你们让我在异乡感受到了家的温暖。四年了，我们在一起的时光将是我大学最宝贵的回忆，谢谢你们陪我一路走来。王婷婷，谢谢你的善良，谢谢你忍受我的牢骚，谢谢你在生活中给对我的帮助。另外，我还要感谢班长钱志雷，谢谢他认真负责的工作。

感谢我的家人，谢谢父亲一直默默的支持我；谢谢懂事的弟弟妹妹们对我的理解，你们在家替我尽了姐姐该尽的义务，让我安心学习，我一定会继续努力，不辜负你们对我的期望。

感谢我的母校西工大，为我提供了这么好的学习环境。

最后，我还要感谢我自己。在我人生低潮的时候，我挺了过来；是我的坚持和不抛弃、不放弃的精神让我走到了今天。

毕业设计小结

随着这篇论文的结束，我的毕设之路也就走到了尽头。转身回看这几个月的走过的路，好多的话想说，却又不知从何说起。

自从去年 10 月份找谢磊老师作为我的研究生导师后，我就开始进实验室学习。在这里，我学到了很多之前三年都不曾学到的东西：我开始学会怎样在电脑前学习，开始习惯阅读英语文献，开始不厌恶调试程序……。最重要的是，我开始向我的不自信发起了挑战，因为之前看到程序代码就发懵的我从来都不敢相信我自己可以独立完成一个实验课题。但是，这次毕设让我重建了信心，让我相信：我能行！

成功不是你跌倒了多少次，而是看你最后一次有没有爬起来！所谓“宝剑锋从磨砺出”，虽然实验中遇到了很多的问题和困难，但是每一次问题的解决和困难的克服都使我成长了一步。

实验中，遇到的第一个问题就是用 VC++编写录音程序这部分。在花了若干时间之后，程序终于能录音了，但发现录音中有很多噪声，我仔细地把程序调试了一遍又一遍，就差微软的程序接口没进去，但始终没找到什么原因。后来偶然的原因，我换了麦克风录音，发现声音中又没有那么多的噪音，才知道原来之前的麦克风出现了问题！我从这个事情上吸取了教训：遇到问题，要从多方面分析原因，不能忽略任何的细节。另外一个让我记忆深刻的事情就是怎样解决系统响应速度的问题，我尝试了很多办法，动态链接库方法、直接调用 exe 文件方法，两个工程结合的方法等等，但都行不通。我甚至都开始怀疑自己能不能解决这个问题了，后来谢老师安慰我说，这是真正锻炼你分析、解决问题能力的时候，需要时间来“磨”，他的这句话给了我极大的鼓励。后来我就想，既然这个耗时的部分不能嵌入到界面中，为什么不能把它独立出来呢？只要两者能通信不就解决问题了嘛！查阅很多资料，证明这个办法可行后，我就照着这个思路走，最后终于得到了小于 1 秒的实时响应速度！ “梅花香自苦寒来”，只有付出了，才能体会到成功的快乐！跨过了这座大山，尽管以后仍然道路崎岖，但我都不怕，因为我已经有“宝剑”在手。

这几个月，我最大的体会就是：学习不能囫囵吞枣、不求甚解，要相信“绳锯木断，水滴石穿”。最后，我想用叶剑英元帅的《公关》诗来结束我这篇小结：

攻城不怕坚，攻书莫畏难。

科学有险阻，苦战能过关。

附 录

附录Ⅰ WaveX API

waveInGetNumDevs	返回系统中存在的波形输入设备的数量
waveInAddBuffer	向波形输入设备添加一个输入缓冲区
waveInGetDevCaps	查询指定的波形输入设备以确定其性能
waveInGetErrorText	检取由指定的错误代码标识的文本说明
waveInGetID	获取指定的波形输入设备的标识符
waveInGetPosition	检取指定波形输入设备的当前位置
waveInMessage	发送一条消息给波形输入设备的驱动器
waveInOpen	为录音而打开一个波形输入设备
waveInPrepareHeader	为波形输入准备一个输入缓冲区
waveInStart	启动在指定的波形输入设备的输入
waveInReset	停止给定的波形输入设备的输入，且将当前位置清零
waveInStop	停止在指定的波形输入设备上的输入
waveInUnprepareHeader	清除由 waveInPrepareHeader 函数实现的准备
WaveInClose	关闭指定的波形输入设置
waveOutBreakLoop	中断给定的波形输出设备上一个循环，并允许播放驱动取列表中的下一个块
waveOutClose	关闭指定的波形输出设备
waveOutGetDevCaps	查询一个指定的波形输出设备以确定其性能
waveOutGetErrorText	检取由指定的错误代码标识的文本说明
waveOutGetID	检取指定的波形输出设备的标识符
waveOutGetNumDevs	检取系统中存在的波形输出设备的数量
waveOutGetPitch	查询一个波形输出设备的当前音调设置
waveOutGetPlaybackRate	查询一个波形输出设备当前播放的速度
waveOutGetPosition	检取指定波形输出设备的当前播放位置
waveOutGetVolume	查询指定波形输出设备的当前音量设置
waveOutMessage	发送一条消息给一个波形输出设备的驱动器
waveOutOpen	为播放打开一个波形输出设备
waveOutPause	暂停指定波形输出设备上的播放
waveOutPrepareHeader	为播放准备一个波形缓冲区
waveOutRestart	重新启动一个被暂停的波形输出设备
waveOutSetPitch	设置一个波形输出设备的音调
waveOutSetPlaybackRate	设置指定波形输出设备的速度
waveOutSetVolume	设置指定的波形输出设备的音量
waveOutUnprepareHeader	清除由 waveOutPrepareHeader 函数实现的准备
waveOutWrite	向指定的波形输出设备发送一个数据块

附录Ⅱ 自建语料库录音抄本

1、疾病给人带来了痛苦	21、姐姐帮助弟弟做功课
2、星期一我得了感冒	22、我热爱警察这个职业
3、八月的天气很闷热	23、文明人每天都刷牙
4、性急容易发生错误	24、眼泪有时会流进鼻子
5、阿姨的女儿在照镜子	25、电灯比油灯进步多了
6、一般的雨衣都漏水	26、爱情绝不是娱乐
7、女工生产黑色的书包	27、西北的橘子是南方的
8、书包里是图书杂志	28、这是全自动洗衣机
9、孩子们到野外去放风筝	29、司令员说话真干脆
10、现在点油灯的少多了	30、文艺要为工农兵服务
11、黄瓜是一种果类蔬菜	31、他赛跑得了第一
12、现在过年也不磕头了	32、我给话剧演员化妆
13、佛教是一种精神信仰	33、先生答复学生的问题
14、口渴了就要喝水	34、编辑不做校对工作
15、他的表现太软弱了	35、姐姐把功课做完了
16、那个医生说他姓罗	36、公共汽车上也有电灯
17、医生的工作是治疗疾病	37、讲地理最好有模型
18、今年果树长得很茂盛	38、黄瓜不是黄色的
19、大家都愿意用自来水	39、每人都有一把椅子
20、他的行为应该受到批评	40、新刊物在元旦出版

附录Ⅲ 基于 GMM-UBM 说话人辨识系统的配置文件

(1) 训练 UBM 模型的配置文件 trainworld.xml

```
<config version="1">
  <param name="debug">false</param>
  <param name="bigEndian">true</param>
  <param name="verbose">true</param>
  <param name="loadFeatureFileFormat">HTK</param>
  <param name="saveMixtureFileFormat">XML</param>
  <param name="loadMixtureFileFormat">XML</param>
  <param name="loadFeatureFileExtension">.mfc</param>
  <param name="featureFilesPath">./</param>
  <param name="featureServerMemAlloc">10000000</param>
  <param name="featureServerMask">0-38</param>
  <param name="vectSize">39</param>
  <param name="distribType">GD</param>
  <param name="minLLK">-200</param>
  <param name="maxLLK">200</param>
  <param name="featureFlags">000000</param>
  <param name="baggedFrameProbabilityInit">0.3</param>
  <param name="baggedFrameProbability">0.6</param>
  <param name="nbTrainIt">10</param>
  <param name="nbTrainFinalIt">1</param>
  <param name="normalizeModel">false</param>
  <param name="alpha">0.25</param>
  <param name="frameLength">0.01</param>
  <param name="addDefaultLabel">true</param>
  <param name="defaultLabel">speech</param>
  <param name="labelSelectedFrames">speech</param>
  <param name="initVarianceFlooring">0.2</param>
  <param name="initVarianceCeiling">4</param>
  <param name="finalVarianceCeiling">4</param>
  <param name="finalVarianceFlooring">0.01</param>
  <param name="inputFeatureFilename">wld_mfccFile.lst</param>
  <param name="saveMixtureFileExtension">.xml</param>
</config>
```

(2) 模型自适应的配置文件 modeladaption.xml

```
<config version="1">
  <param name="debug">false</param>
  <param name="bigEndian">true</param>
  <param name="loadFeatureFileFormat">HTK</param>
  <param name="loadFeatureFileExtension">.mfc</param>
  <param name="featureFilesPath">.\mfcc\</param>
  <param name="featureServerMemAlloc">10000000</param>
  <param name="vectSize">39</param>
  <param name="mixtureDistribCount">256</param>
  <param name="topDistribCount">128</param>
  <param name="distribType">GD</param>
  <param name="mixtureFilesPath">.\ClientMixture\</param>
  <param name="loadMixtureFileFormat">XML</param>
  <param name="loadMixtureFileExtension">.xml</param>
  <param name="saveMixtureFileFormat">XML</param>
  <param name="saveMixtureFileExtension">.xml</param>
  <param name="minLLK">-200</param>
  <param name="maxLLK">200</param>
  <param name="featureFlags">000000</param>
  <param name="mixtureServer">target</param>
  <param name="useIdForSelectedFrame">true</param>
  <param name="baggedFrameProbability">1</param>
  <param name="nbTrainIt">5</param>
  <param name="nbTrainFinalIt">1</param>
  <param name="meanAdapt">true</param>
  <param name="MAPAlgo">MAPOccDep</param>
  <param name="normalizeModel">false</param>
  <param name="MAPRegFactorMean">10</param>
  <param name="alpha">0.75</param>
  <param name="frameLength">0.01</param>
  <param name="addDefaultLabel">true</param>
</config>
```

(3) 计算对数似然率的配置文件 computetest.xml

```
<config version="1">
  <param name="debug">false</param>
  <param name="bigEndian">true</param>
  <param name="verbose">false</param>
  <param name="loadFeatureFileFormat">HTK</param>
  <param name="saveMixtureFileFormat">XML</param>
  <param name="loadMixtureFileFormat">XML</param>
  <param name="loadMixtureFileExtension">.xml</param>
  <param name="loadFeatureFileExtension">.mfc</param>
  <param name="featureFilesPath">.\</param>
  <paramname="mixtureFilesPath">.\Mixtures\</param>
  <param name="saveMixtureFileExtension">.xml</param>
  <param name="featureServerMemAlloc">10000000</param>
  <param name="featureServerMask">0-38</param>
  <param name="vectSize">39</param>
  <param name="distribType">GD</param>
  <param name="minLLK">-200</param>
  <param name="maxLLK">200</param>
  <param name="featureFlags">000000</param>
  <param name="baggedFrameProbabilityInit">0.1</param>
  <param name="baggedFrameProbability">0.2</param>
  <param name="nbTrainIt">10</param>
  <param name="nbTrainFinalIt">1</param>
  <param name="normalizeModel">false</param>
  <param name="alpha">0.25</param>
  <param name="gender">M/F</param>
  <param name="frameLength">0.01</param>
  <param name="addDefaultLabel">true</param>
  <param name="defaultLabel">speech</param>
  <param name="labelSelectedFrames">speech</param>
  <param name="segmentalMode">segmentLLR</param>
  <param name="inputWorldFilename">wld_256</param>
  <param name="mixtureDistribCount">256</param>
```

(4) HCopy 提取 MFCC 特征的配置文件 config

```
SOURCEKIND  = WAVEFORM
SOURCEFORMAT  = WAV
ZMEANSOURCE        = F
TARGETKIND    = MFCC_E_D_A
TARGETRATE    = 200000.0  # frame period = 10msec
SAVEWITHCRC     = T
WINDOWSIZE    = 350000.0  # window size = 35msec
USEHAMMING    = T
PREEMCOEF    = 0.95    # 1st order preemphasis, coefficient = 0.95
NUMCHANS     = 24   # num. of filterbank channel = 24
CEPLIFTER          = 22   # num. of cepstra = 22
NUMCEPS          = 12   # num. of MFCC coefficient = 12
ENORMALISE    = T  # energy normalization (live: F, otherwise: T)
```