语音识别自适应算法在智能家居中的应用①

蒋 泰, 张林军

(桂林电子科技大学 计算机与信息安全学院, 桂林 541004)

摘 要: 在基于语音识别的智能家居中, 用于训练的语料库不完备且应用场景复杂, 自然语言语音识别错误接受率远远高于小词汇的语音识别的错误接受率. 作者在设计与实现基于自然语言的语音识别智能家居系统的过程中, 深入研究了 MAP、MLLR 算法在基于 HMM 声学模型参数中的作用, 提出了一种综合的自适应方法, 并基于开源的语音识别工具 CMU SPHIN 最终完整的实现了该系统, 结果表明所提出的自适应新算法可行有效, 较好改善了系统在不同场景中的性能.

关键词: 语音识别; 自适应; MAP; MLLR; 智能家居; 开源工具

Speech Recognition Adaptive Algorithm in the Application of the Smart Home

JIANG Tai, ZHANG Lin-Jun

(School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In smart home based on speech recognition, the corpus used for training is not complete and the application scenario is complex. Besides, the false acceptance rate of natural language speech recognition is much higher than that of small vocabulary speech recognition. During the procedure of designing and trying to implement smart home system based on natural language speech recognition, the author makes an intensive study of the MAP, MLLR algorithm based on the role of HMM acoustic model parameters. This paper presents a comprehensive adaptive method, based on which the author completed the system by using open source tools CMU SPHIN. The experiment result shows that the presented new adaptive algorithm is feasible and effective, and makes the system performance better in different scenarios.

Key words: speech recognition; adaption; MAP; MLLR; smart home; open-source tools

现代社会已经进入了高科技迅猛发展的信息时代, 随着信息技术的飞速发展, 以及人们生活水平的提高, 智能家居开始走进普通家庭. 大众接受智能家居的概念的同时也对智能家居提出了更高的要求. 加上国家政府大力推进城镇化建设, 一些地产开发商需要更高标准的高智能家居设备和智慧社区方案来提高其产品的附加值以便提高其地产产品的竞争力. 如今, 家居的语音智能化控制已成为计算机、通信等行业竞相研究的热点[1].

训练环境与识别环境存在大量差异是造成语音识别系统性能下降的主要原因之一. 说话人、环境和信道特征等不同所引起的语音信号的多变性、差异性, 一直是人们关注的重点和难点[2]. 如何使语音识别系统克服这种差异, 通过少量数据得到较高的性能十分重要. 自适应技术可以解决这个问题, 其主要研究的内容是通过对系统参数进行调整, 使系统更好地适应由麦克风、传输信道、环境噪音、说话人、文体和应用的上下文等引起的差异. 而基于模型层的自适应算法能充分利用有限的自适应数据通过对模型参数进行调整, 逐渐将模型参数变换到实际环境, 从而来提高语音识别系统的性能.

最大后验概率(Maximum A Posteriori, MAP)算法和最大似然线性回归(Maximum Likelyhood Linear Regression, MLLR)算法是基于模型层说话人自适应中

① 基金项目:2014 年国家物联网发展专项资金项目(工信部科函[2014]351 号)

收稿时间:2016-05-17;收到修改稿时间:2016-06-27 [doi:10.15888/j.cnki.csa.005591]

的基本算法[3]. MAP 有很好的渐进性,可以充分利用语音数据的细节信息. 它通过理论给出了结合先验知识和自适应数据的最优解. 基于 MAP 的自适应方法认为模型参数是符合某种先验分布的随机变量, 将先验知识和从自适应数据中得到的知识结合起来估计模型参数, 避免了自适应数据估计的错误. 该方法有很好的渐进性, 当自适应数据不断增加时, 自适应效果将稳步提高. 但是算法收敛速度慢, 只能对有观测数据的模型自适应, 无法处理没有观测值的模型. MLLR 方法是通过一些线性变换来对初始模型进行自适应的. 这种方法的优点是比较简单,而且自适应速度比较快. 即使自适应数据量不足, 方法也可以获得较理想的效果. 由于自由参数少, 很难对每个模型精细描述, 而且比较难以引入先验知识. 针对以隐马尔可夫模型作为建模基础的声学模型, 本文研究以相对较少自适应数据的情况下获得较好性能的自适应方法. 为了提高语音识别的性能, 一般语音识别系统都综合使用几种自适应技术. 本文提出了一种综合渐进自适应方法, 通过在渐进的MAP算法中引入了一个简化的MLLR模块, 用来处理语音识别中训练数据与实际应用场景不同的差异, 并且使用 CMU SPHINX 开源工具使其运用到智能家居领域, 最终使用较少的自适应数据取得了较好的应用效果.

1 语音识别的智能家居框架

本系统由软件和硬件两部分组成, 如图 1 所示. 软件部分又分为云端和嵌入式客户端. 云端和客户端各自集成了几个主要模块, 分别实现不同的功能.

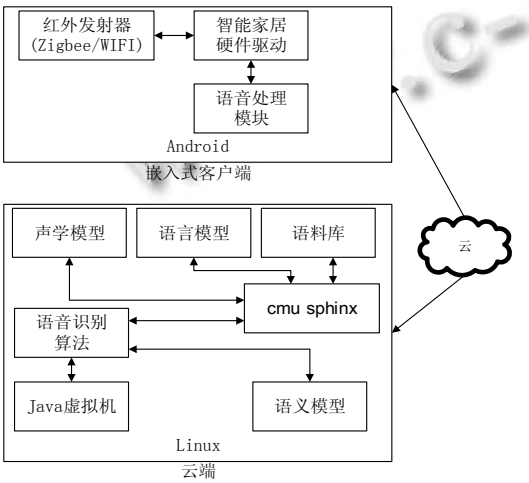

图 1 系统架构模型

云端软件运行在 Linux 上, 它包含声学模型、语音模型、语料库、CMU Sphinx 工具包、Java 虚拟机. 云端主要功能是接受客户端发送的语音文件转化成文本文件返回, 应用程序模块全部使用 Java 开发, 调用 CMU Sphinx 提供的类库. 主要功能包含语音识别算法和语义转换模块, 它被部署在 Java 虚拟机上. 语音识别算法的主要过程有: 语音输入、预处理、特征提取、模型匹配、输出结果. 首先必须使用 CMU Sphinx 的训练工具以特定声学模型为基础对语料库获取匹配的 MFCC 特征数据, 然后使用 MAP 和 MLLR 自适应技术来改进原来的声学模型. 本文主要讨论以 HMM 为基础的声学模型使用 MAP、MLLR 对其参数进行自适应.

嵌入式客户端软件包含语音识别模块和嵌入式硬件驱动模块. 语音识别模块功能包含三个方面: 获取客户语音并生产语音文件、发送语音文件到云端并接受返回的硬件原语文件、通过硬件原语文件来驱动外围电路. 驱动模块主要是控制外围设备并对语音模块提供服务支持.

2 最大后验概率(MAP)

基于 MAP 的自适应方法是基于最大后验准则并通过引入先验知识来求最大后验概率[4], 进而提高自适应效果, 它与最大似然重估方法相对应叫做最大后验概率重估方法[5]. 假设 $O=\{O_1,O_2,\ldots,O_r\}$是概率密度函数为 $p(O)$ 的一系列观察值, 是定义分布的参数集合. MAP 可以看作是给定训练数据序列 O, 估算 λ 的过程, 如下:

$$\lambda_s=\text{argmax}\ p(\lambda|O) \quad (1)$$

利用贝叶斯准则(Bayes Rule), 其中 $p(\lambda)$ 是 HMM 参数的先验分布, 引入符合先验分布 $p(\lambda)$的随机变量 λ, 可以得到:

$$\lambda_{map}=\text{argmax}\ p(\lambda|O)=\text{argmax}\ p(O|\lambda)\,p(\lambda) \quad (2)$$

需要强调的是先验知识是MAP成功的关键. 对于具有高斯混合密度的 CDHMM, 所有参数的先验知识中, 由于在 CDHMM 参数中均值向量对识别结果的影响最大, 因此 MAP 一般只对均值重估.

假设观察值 $\chi_1, \chi_2, \ldots, \chi_3$ 的均值 φ 未知, 且服从方差为 σ^2 的高斯分布, 同时假定 φ 的共轭先验分布与 χ 同分布, 且均值为 μ, 方差是 V^2. 经计算可得:

$$\phi=\frac{\sigma^2 u+nv^2\bar{\chi}_n}{\sigma^2+nv^2} \tag{3}$$

其中 $\bar{\chi}_n=\frac{1}{n}\sum_{i=1}^{n}\chi_i$.

把式(3)经过符号替换后可以得到 MAP 均值重估公式, 即:

$$\hat{u}_k=\frac{\tau_k\mu_k+n_k m_k}{\tau_k+n_k} \tag{4}$$

从式(4)可以看出, n^k 与自适应数据成正比, 即 MAP 重估与自适应数据成正比且随着自适应数据的增加而逐渐逼近最大似然估计(Maximum Likelihood, ML), 当自适应数据趋于无穷时, 用 MAP 法所得的模型与用充分语料做 ML 训练所得的模型基本相同[6]. 对于式(4)转换改写如下:

$$\hat{u}_k=\frac{\tau_k\mu_k+\sum_{t=1}^{T}c_{kt}\cdot o_t}{\tau_k+\sum_{t=1}^{T}c_{kt}} \tag{5}$$

其中 $c_{kt}=\frac{\omega_k\cdot N(o_t/\mu_k,\sum_k)}{\sum_k\omega_k\cdot N(o_t/\mu_k,\sum_k)}$, O_t 是表示对应的训练数据 $N\left(o_t/\mu_k,\sum_k\right)$ 属于高斯分布, $\sum_k$ 是高斯分布的协方差矩阵, ω_k 是混合系数. 由此可知自适应调整后的均值向量为初始均值与相应的训练数据的线性加权之和. 基于 MAP 的自适应有一个较重要的制约, 即要求先验分布 $P(\Phi)$有较准确的估计. 对于 u_k的估计, 可以直接使用已训练好的初始模型参数, 也可以通过多个已训练好的参数建立统计模型.

3　最大似然线性回归(MLLR)

当 HMM 用在声学建模时, 对较重要的参数进行自适应, 一般使用均值向量和协方差矩阵. 最大似然回归(Maximum Likelyhood Linear Regression, MLLR)方法是使用最大似然(Maximum Likelyhood, ML)重估算法对自适应数据估算出一套转移参数, 用转移参数把 SI 系统参数 y 转换成自适应模型参数 x[7], 即

$$\chi=Ay+b \tag{6}$$

对于式(6)进行变换参数的估计, 应用到重新估计 CDHMM 模型的新均值 $\hat{\mu}$, 可以用下面的公式表示:

$$\hat{\mu}_s=W_s\xi_s \qquad \xi=[\omega,u_1,u_2,\ldots,u_n]^T \tag{7}$$

其中 W 为 $n\times(n+1)$维矩阵, $u_1,u_2,\ldots,u_n$ 是变换前的均值向量, ω 为变换的偏移量, $\hat{\mu}$ 为自适应后的向量.

采用单一的 MLLR 转换方法效果有限, 在采用多回归类的情况下, 目标就是得到回归类中所有元素共享的转移矩 W_s. 应用自适应数据的最大似然估计, 通过定义和全矩阵同样的辅助函数, 经推导可得:

$$\sum_{p=1}^{p}\sum_{t=1}^{T_p}\sum_{r=1}^{R}\gamma_{s,k}^{(p)}(t)D_{s,k}^{T}\sum_{s,k}^{-1}o_t^{(p)}=\sum_{p=1}^{p}\sum_{t=1}^{T_p}\sum_{r=1}^{R}\gamma_{s,k}^{(p)}(t)D_{s,k}^{T}\sum_{s,k}^{-1}D_{s,k}\hat{w}_s \tag{8}$$

其中矩阵 $D_{s,k}$ 是这样定义的[8]:

$$D_{s,k}=\begin{bmatrix}\omega & 0 & \ldots & 0 & u_{s,k1} & 0 & \ldots & 0\\ 0 & \omega & \ddots & \vdots & 0 & u_{s,k2} & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0\\ 0 & \ldots & 0 & \omega & 0 & \ldots & 0 & u_{s,kn}\end{bmatrix} \tag{9}$$

现在定义两个矩阵 A(2n×n)和 B(2n×1)和非零向量 ω_s:

$$A=\sum_{p=1}^{p}\sum_{t=1}^{T_p}\sum_{r=1}^{R}\gamma_{s,k}^{(p)}(t)D_{s,k}^{T}\sum_{s,k}^{-1}D_{s,k} \tag{10}$$

$$B=\sum_{p=1}^{p}\sum_{t=1}^{T_p}\sum_{r=1}^{R}\gamma_{s,k}^{(p)}(t)D_{s,k}^{T}\sum_{s,k}^{-1}o_t^{(p)} \tag{11}$$

$$\omega_s=[\omega_{1,1},\omega_{n,1},\ldots,\omega_{2,1},..,\omega_{n,n+1}]^T \tag{12}$$

可以得到

$$B=A\hat{W} \qquad \hat{W}=A^{-1}B \tag{13}$$

4　MAP与MLLR综合的自适应技术

一般而言, MLLR 自适应的速度较 MAP 而言要快, 尤其表现在自适应数据较少时, 但随着数据增加, MAP 的优势会逐步突显出来, 而 MLLR 在自适应数据达到一定数量时, 算法将趋于饱和. 本文在研究了二者的优缺点后, 提出了一种新的自适应方法, 即由简化的自适应模块 MLLR 和渐进的 MAP 自适应模块组合而成[9]. 如图 2 所示.

对于简化的 MLLR 自适应模块, 主要使用当前的语句, 因为 MLLR 方法中所有模型只使用了一个回归类, 换句话说所有的模型的自适应数据都用来重估同一转移矩阵, 这样一来当前语句就可以满足自适应的要求. 对于 MAP 模块, 它的主要目的是消除音素层的差异. 为了能够对每个音素的细微特点进行自适应, 同时考虑到其方法对自适应数据量需要比较大的特点, 在 MAP 模块中使用所有累计的自适应数据. 由于 MAP 的自适应是在 MLLR 的结果上进行, 为了解决计算所有样本的参数计算量太大的问题, 在计算时, 必须充分利用上一次计算的一些中间值. 另外, 通过

测试结果发现, 本文所提出的组合方法在经过很少几次的组合渐进自适应后就有很好的自适应效果, 不必计算所有累计的样本, 只需要计算最近的几次的累计数据. 综合的自适应策略如图 3 所示.

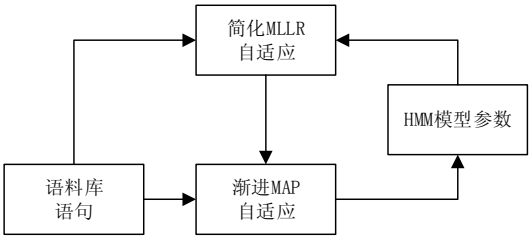

图 2 自适应模块

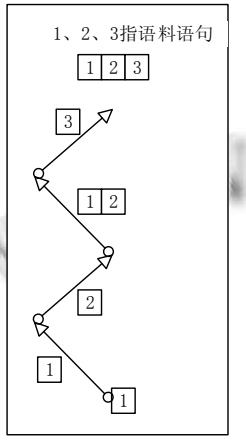

图 3 自适应策略

5 系统的实现

5.1 系统总体流程

系统分为嵌入式客户端和云端, 如图 4 所示.

5.2 嵌入式客户端的实现

客户端为 ARM Cortex-A9 架构处理器, 客户端系统软件用 Android 开发, 使用 Android 可以充分利用其框架进行蓝牙、红外、ZigBee 扩展. 为了保证程序随系统自动启动, 必须包含一个扩展 BroadcastReceiver 组件的类用以捕获 ACTION_BOOT_COMPLETED 这条广播, 并在捕获之后使用 startService 启动程序的 Service, 这样可以使程序以常驻内存的服务一样后台运行. 当用户触发语音指令请求后, 程序调用组件 MediaRecorder 对象从 MIC 获取声源并转换成 wav 文件, 然后通过 TCP 传送 wav 文件给云端, 最后接受云端返回的 JSON 数据, 根据 JSON 数据结果操作设备或者传感器.

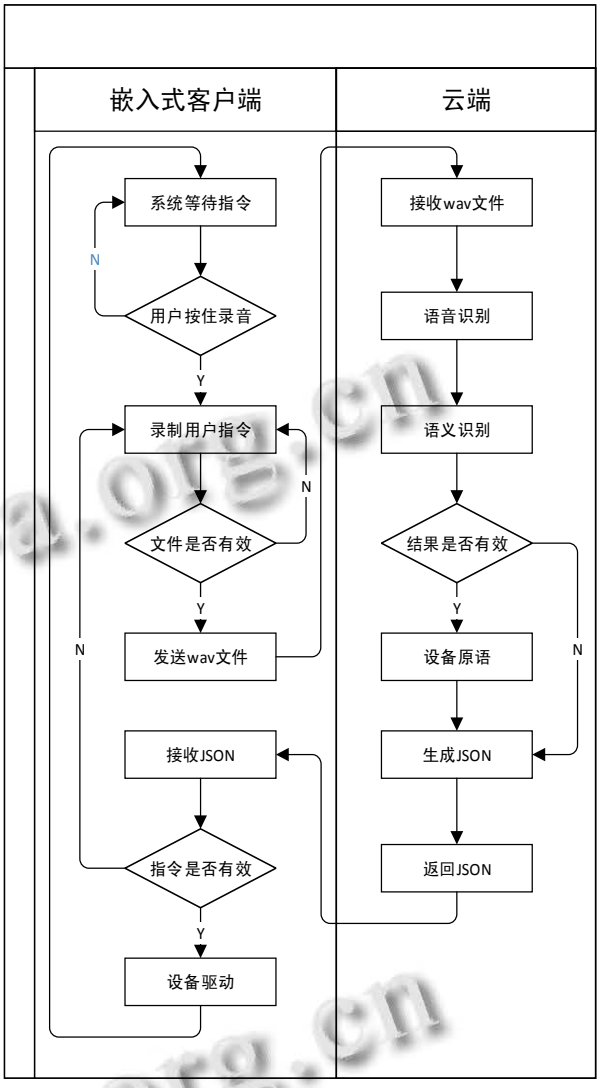

图 4 系统流程图

5.3 云端

云端服务器操作系统为 Linux, 语音识别核心主要使用 Java 开发, 使用 MUC SPHINX 开源工具加载声学、语音学模型和数据字典进行语音识别任务. 程序运行在 Java VM 上, 在指定范围端口等待服务, 当接收到 wav 文件, 程序会调用 MUC SPHINX 加载相应模型数据进行语音解码, 把结果传递给语义模块进行分析最终产生硬件原语以 JSON 的格式返回给客户端. 在现实应用中不可能收集所有情形下的语音数据, 并且场景数据越多识别系统越分散, 任一个具体场景下的识别工作不可能和预期非常匹配, 自适应技术在本系统中使用非常必要. 现重点对现有声学模型进行自适应的改进进行说明.

(1) 创建语料库包含两个文件: arctic20.fileids、arctic20.transcription, 文件 arctic20.fileids 包含语音文件路径, arctic20.transcription 内容为文本句子对应的语音文件.

(2) 为适应语料库里面的每一个句子录制一个语音文件, 录音文件的内容和命名必须同 arctic20.transcription 里的表示一致, 音频文件采样率为 16KHz 的 16bit 单声道的 wav 格式.

(3) 从录制的 wav 语音文件提取 MFCC 特征存储在 feat.params 文件中, 使用 SphinxTrain 训练工具自带的 bw 程序累加 feat.params 文件中的统计数据.

(4) 执行命令行 mllr_solve 通过 MFCC 特征文件和声学模型进行 MLLR 变换产生适应数据文件 mllr_matrix.

(5) 执行命令行 map_adapt 使用自适应数据更新模型参数.

云端主要代码如下:

```
Configuration conf = new Configuration();//加载配置文件
String sModelDirection="resource:/edu/cmu/sphinx/models/Mandarin";
String sWavDirection="/com/ht/";
conf.setAcousticModelPath(sModelDirection);
conf.setAcousticModelPath("file:zh");
conf.setDictionaryPath(sModelDirection+"/cmudict-en-us.dict");
configuration.setLanguageModelPath(sModelDirection+"/en-us.lm.bin");
StreamSpeechRecognizer recognizer = new StreamSpeechRecognizer(conf);
while(true){
    if (lsWavFile.size()>0)//接受到 wav 文件, 启动解码
{   InputStream stream=HtRecognizer.class.getResourceAsStream(sWavDirection+lsWavFile.get(0));
        String sWords="";
        recognizer.startRecognition(stream);
        SpeechResult result;
        while ((result = recognizer.getResult()) != null) {
            for (WordResult r : result.getWords())
        sWords+=r;
            if (sWords.length>0)//成功解码, 返回客户端
sendData(sWavDirection+lsWavFile.get(0),sWords);
            lsWavFile.remove(0);    }
    }
    recognizer.stopRecognition();
```

6 语音识别模块的实验和仿真

为了验证自适应技术的有效性, 实验使用自建语音数据库. 该数据库包含普通话语音以及少量的英文数字语音, 共 4 个说话人(女 2 个, 男 2 个), 且每个说话人包含 810 句录音, 说话人年龄在 25-35 岁之间, 分别来自我国北方、中部和南方地区, 其中都带有轻微的地方口音. 每人选取 720 句作为训练语句, 剩下的 190 句为测试语句. 810 句录音内容由指定的 150 句常用的智慧家用语, 其余的从报纸和字典摘取 360 句和从网络新闻中抽取的 400 句组成.

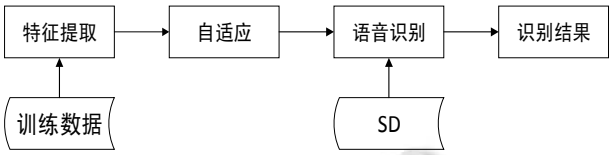

图 5　自适应模型

实验结果如表 1 所示, “SI”列为初始的非特定人系统的性能, “SD”列为训练语音库所对应的特定人语音识别字错误率, “综合”列为系统使用 MLLR 和 MAP 自适应技术后非特定的系统性能, *P*1, *P*2, *P*3, *P*4 表示四个说话人.

表 1　综合实验结果

字错误率	SI(%)	SD(%)	MLLR(%)	MAP(%)	综合(%)
*P*1	15.1	8.4	48.1	10.9	7.9
*P*2	18.5	10.5	50.3	12.4	8.2
*P*3	24.3	12.3	61.5	18.7	9.6
*P*4	28.7	14.9	57.9	20.1	10.3
测试集	21.7	11.53	54.45	15.53	9.00

不同的语音识别系统针对的应用场景不同, 所用的字典和语料库也不同, 所关注的性能评测指标也有所取舍. 本文关心的是系统最终的文字识别率, 假设总文本字数为 *N*, 不正确的识别字数 *R*、没识别出的字

数 D、错误识别的字数 I, 则有字错误率 E 公式:

$$E=\frac{R+D+I}{N}$$

从结果中可以清楚的看到, 使用综合的自适应后的模型比 SI 模型字错误率有了很大的降低, 测试集降低了 12.7%. 此外, 虽然 MAP 和 MLLR 有自身优点和缺点, 但是也有一定的互补性. 实验过程表明: MAP 和 MLLR 自适应技术跟凭此使用的先后顺序没有关系, 综合自适应技术对于提升系统性能都有较好体现.

表 2 自适应实验结果

字错误率		SI(%)	MAP(%)	综合(%)
M	10 句	31.5	25.4	23.1
	15 句	19.1	16.7	14.8
W	10 句	20.4	15.1	14.9
	15 句	17.6	13.5	11.5

如表 2 所示, *M* 表示男性, *W* 表示女性, 10 句和 15 句表示从相应库中随机取 10 句, 15 句. 随着自适应数据的增加, MAP 自适应效果较明显, 不过在 15 句以后这种改善就不是很明显了, 综合的自适应方法不论是在自适应数据比较少的时候, 还是处在有噪声的环境下都比原有的 SI 效果要好, 这说明在 MLLR 方法中引入 MAP 方法对处理说话人差异和环境差异的系统中都取得了较好的效果.

7 总结

本文设计并实现的基于自然语音的智能家居系统与传统的语音识别的智能家居系统在语音识别技术方面有着较大的区别, 对系统在语音识别方面有较好的提升, 适合强健语音识别系统的要求.

参考文献

1 陈哲.智能家居语音控制系统的设计与实现[硕士学位论文].成都:电子科技大学,2013.

2 李虎生,刘加,刘润生.语音识别说话人自适应研究现状及发展趋势电子学报.电子学报,2003,31(1).

3 齐耀辉,潘复平,葛凤培,颜永红.鉴别性最大后验概率声学模型自适应.计算机应用,2014,34(1):265–269.

4 詹新明.基于神经网络的语音识别研究[硕士学位论文].广州:华南理工大学,2013.

5 Atal BS. Automatic recognition of speakers from their voices. Proc. IEEE. 1976, 64(4): 460–475

6 王治锋.基于多核融合和模型参数适应的非特定人语音情感识别研究[硕士学位论文].苏州:江苏大学,2012.

7 Imamura A. Speaker-adaptive HMM-based speech recognition with a stochastic speaker classifier. Proc. IEEE Int. Conf. Acoustic, Speech, Signal Processing. 1991. 841–844.

8 荣薇,等.基于改进 LPCC 和 MFCC 的汉语耳语音识别.计算机工程与应用,2007,43(30):213–216.

9 Digalakis VV, Neumeyer LG. Speaker adaption using combined transformation and Bayesian methods. IEEE Trans. on Speech and Audio Processing, 1996, 4(4): 294–300.