基于 CHMM 孤立词识别系统的快速实现

廖广锐[1],胡 玥[2],刘 萍[1]
(1.中国船舶重工集团公司 第 709 研究所,湖北 武汉 430074;
2.武汉虹信通信技术有限责任公司,湖北 武汉 430074)

摘 要:介绍了基于连续隐含马尔可夫模型(CHMM)的非特定人孤立词语音识别系统。根据分析该系统计算复杂度,提出了一系列的优化方法,通过 MATLAB 平台下的研究实验数据表明,优化后的语音识别系统与传统 CHMM 语音识别系统对比,计算时间是传统 CHMM 系统的 9.97 %,而识别率仅从传统 CHMM 系统的 94 %下降到 91.3 %。

关键词:HMM;语音识别;MFCC;Viterbi
中图分类号:TP391.42 **文献标识码**:A

Fast implementation of isolated-word recognition system based on CHMM

LIAO Guang Rui[1], HU Yue[2], LIU Ping[1]
(1.709th Research Institute, China Shipbuilding Industry Corporation, Wuhan 430074, China;
2.Wuhan Telecommunication Co., Ltd, Wuhan 430074, China)

Abstract: Speaker-independent isolated-word recognition system based on continuous HMM is presented. This paper analyzed the computational complexity of the system. A series of optimized way is suggested to reduce the computation complexity. The experimental data based on MATLAB platform show that, the computation time of the optimized speech recognition system is 9.97 % of the traditional CHMM speech recognition systems, and the recognition accuracy is degraded only from 94 % to 91.3 %.

Key words: HMM; speech recognition; MFCC; viterbi

语音识别是近年来十分活跃的一个研究领域。在不远的将来,语音识别技术有可能作为一种重要的人机交互手段。目前市场实用的产品多是基于动态时间规整(DTW)算法的特定人语音识别系统,或者是基于离散 HMM 和半连续 HMM 的非特定人语音识别系统。其中离散 HMM 的模型参数少,对训练数据量要求不高,而且离散 HMM 的计算量较少,易于实时实现,但是离散 HMM 的缺点是识别精度不高。半连续型 HMM 的每个状态的输出概率分布是由几个正态分布函数叠加而成的,但是这些正态分布函数与状态无关(实际上与模型也无关),即每个状态都使用共同的正态分布函数,因此半连续型 HMM 用多个正态分布线形相加作为概率密度函数弥补了离散分布的误差,相对于连续型 HMM,半连续型 HMM 用多个各状态共有的正态分布线形相加作为概率密度函数弥补了参数数量多、计算量大的缺陷。连续 HMM 可以进一步提高系统的识别率,但是需要保存的参数数量很多,计算量很大,无法满足实时实现。

本文分析了语音识别过程中的计算复杂度,通过以下4 种方法来降低计算复杂度:(1)降低高斯密度的混合数;(2) 降低每个音节的状态数;(3) 延长待识语音帧移;(4)待识语音长度与测试集中词条字数的比较。达到了提高识别速度的目的,最后提供了实验数据,证实了此方案的可行性。

1 传统的 HMM 语音识别系统

本文所介绍的传统语音识别系统是基于音节的非特定人连续 HMM 孤立词语音识别系统。整个系统总共分为 2 个部分:语音训练过程和语音识别过程。系统原理如图 1 所示。

语音训练过程是在 PC 机上完成,大量的训练数据经过特征参数提取后,利用 Baum-Welch 算法为测试集

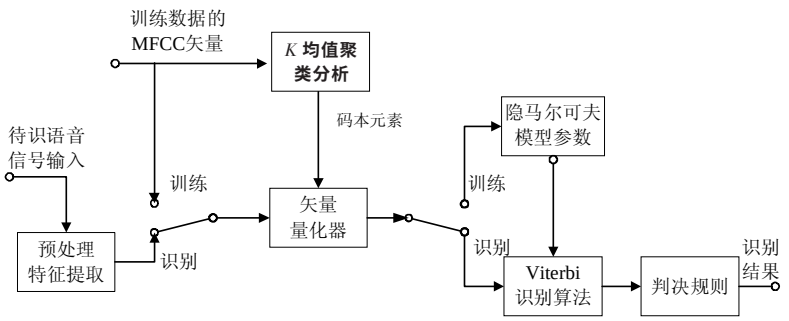

图 1 CHMM 孤立词识别系统原理

中每个词条建立基于音节的 HMM 模型。语音识别过程是将输入的模拟语音信号首先进行预处理，包括预滤波、采样、归一化、分帧、预加重、端点检测等，之后是语音信号的特征参数提取，最后采用 Viterbi 动态规划算法计算出待识别语音的特征参数对于每个词条 HMM 模型的输出概率，通过判决规则输出识别结果。整个语音识别过程主要分为 3 个部分：待识语音信号的预处理过程、特征提取过程、Viterbi 识别算法过程。

1.1 语音信号的预处理

本系统的输入信号首先经过一个截止频率为 4 kHz 的低通滤波器，然后经过 A/D 转换，采样率为 8 kS/s，采样精度为 16 位，对语音数据进行归一化处理，把采样值的范围从[-32 767，32 767]转换到[-1，1]，取 N=256 采样点为帧长，帧移为 80 点得到第 i 帧，第 n 个样本的语音信号为 $s_i(n)$。

预加重处理是将语音信号通过 1 个一阶高通滤波器 $1-0.937\ 5/z$，目的在于滤除 50 Hz 或 60 Hz 的低频干扰，将高频部分的频谱进行提升。第 i 帧、第 n 个样本 $x_i(n)$与语音信号 $s_i(n)$的关系由公式(1)表示。

$$x_i(n)=s_i(n)-0.937\ 5\times s_i(n-1) \tag{1}$$

本系统采用短时能量和过零率双门限的方法进行端点检测，精确地检测到语音的起点和终点。其中第 i 帧语音信号的短时能量由公式(2)得到。

$$e(i)=\sum_{n=1}^{N}|x_i(n)| \tag{2}$$

第 i 帧的过零率加 1 则应同时满足公式(3)、公式(4)的条件(其中 n=2，3，…，N，δ=0.02)。

$$s_i(n)*s_i(n-1)<0 \tag{3}$$

$$|s_i(n)-s_i(n-1)|>\delta \tag{4}$$

1.2 语音信号的特征提取

选择 Mel 频率倒谱参数 MFCC(Mel-scaled Cepstrum Coefficients)作为特征参数，其考虑了人耳的听觉特性能很好地表征语音信号，而且在噪声环境下能取得很好的识别效果。计算通常采用如下的流程：

(1)对每帧语音序列进行预加重处理，把每帧数据乘以一个 hamming 窗，以克服 Gibbs 现象，再经过离散 FFT 变换，取模的平方得到离散功率谱 $P(n)$。窗函数的构造如公式(5)所示

$$w(n)=0.54-0.46\cos\left(\frac{2\pi n}{N-1}\right)$$
$$n=0,1,\cdots,N-1 \tag{5}$$

(2)构造好 24 个在 Mel 频率轴上均匀分布的滤波器组 $H_m(n)$，m=0，1，…，23，n=0，1，…，127。构造过程：如公式(6)所示将实际频率尺度转换为 Mel 频率尺度，如公式(7)计算出 $H_m(n)$，其中 $o(m)$、$c(m)$和 $h(m)$分别是第 m 个滤波器的下限、中心和上限频率。计算 $P(n)$通过这 24 个 $H_m(n)$所得的功率值，即计算 $P(n)$和 $H_m(n)$各离散频率点上乘积之和，如公式(8)所示得到 24 个参数 P_m。

$$f_{mel}=2\ 595\lg(1+f/700) \tag{6}$$

$$H_m(n)=\begin{cases}\dfrac{n-o(m)}{c(m)-o(m))} & o(m)\leqslant n\leqslant c(m)\\ \dfrac{h(m)-n}{h(m)-c(m)} & c(m)\leqslant n\leqslant h(m)\end{cases} \tag{7}$$

$$P_m=\sum_{n=o(l)}^{h(l)}H_m(n)|P(n)| \quad m=0,1,\cdots,23 \tag{8}$$

(3)计算 P_m 的自然对数，得到 L_m，对 L_0，L_1，…，L_{23} 计算离散余弦变换(DCT)，得到 D_n，舍去代表直流成分的 D_0，取 D_1，D_2，…，D_K 作为 MFCC 参数。此处 K=12。并保留 DCT 前 12 个系数即为 MFCC 系数(静态特征)。计算过程如公式(9)所示。

$$D_n=\sum_{m=1}^{24}\lg(p_m)\cos\left\{(m-\frac{1}{2})\frac{n\pi}{24}\right\} \quad n=0,1,\cdots,23 \tag{9}$$

(4)根据公式(10)计算其一阶动态特征 C_n，其中 k 通常取 2，将 MFCC 参数和一阶差分参数合并为 1 个矢量，作为 1 帧语音信号的特征参数。

$$C_n=\frac{1}{\sqrt{\sum_{i=-k}^{k}i^2}}\sum_{i=-k}^{k}iD_{(n+i)} \tag{10}$$

1.3 语音识别

本文所采用的语音识别算法是 Viterbi 算法，Viterbi 算法不仅可以找到 1 条足够好的状态转移路径，还可以得到该路径所对应的输出概率。同时 Viterbi 算法的计算量要比全概率公司的计算量小很多。设 $\boldsymbol{O}=o_1,o_2,\cdots,o_T$ 表示待识语音的特征矢量；S_k 表示第 k 个状态，最多 N 个状态；a_{ij} 表示状态 S_i 到状态 S_j 的概率；$P(O/M)$表示给定模型 M 时输出观察序列 O 的概率；$b_{ij}(o_t)$表示状态 S_i 到状态 S_j 发生转移时输出观察矢量 $\boldsymbol{o}_t$ 的概率；$\alpha_t'(j)$表示输出部分观察序列 $o_1,o_2,\cdots,\boldsymbol{o}_t$，并到达状态 S_j 的概率。a_{ij} 由训练所得的 CHMM 模型给出；由于 $\boldsymbol{o}_t$ 是 24 维矢量，所以用多元高斯密度函数表示 $b_{ij}(\boldsymbol{o}_t)$，如公式(11)计算所得，其中 $\boldsymbol{\mu}_{ij}$ 是均值矢量，由 CHMM 模型提供。

$$b_{ij}(\boldsymbol{o}_t)=P(\boldsymbol{o}_t|i,j)=\frac{1}{2\pi^{T/2}\left|\sum_j\right|^{1/2}}$$

$$\exp\left\{-\frac{1}{2}(\boldsymbol{o}_t-\boldsymbol{\mu}_{ij})\sum_j^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_{ij})^t\right\} \quad (11)$$

Viterbi 算法求取最佳状态序列的步骤如下：

(1)给每个状态准备一个数组变量 $\alpha_t'(j)$，初始化时令初始状态 S_1 的数组变量 $\alpha_0'(1)$为 1，其他状态的数组变量 $\alpha_0'(j)$为 0。

(2)根据 t 时刻输出的观察符号 o_t 根据公式(12)计算 $\alpha_t'(j)$：

$$\alpha_t'(j)=\max_i \alpha'_{t-1}(i)a_{ij}b_{ij}(\boldsymbol{o}_t)=$$
$$\max_i\{\alpha'_{t-1}(i)a_{1j}b_{1j}(\boldsymbol{o}_t),\alpha'_{t-1}(i)a_{2j}b_{2j}(\boldsymbol{o}_t),$$
$$\cdots,\alpha'_{t-1}(i)a_{Nj}b_{Nj}(\boldsymbol{o}_t)\} \quad (12)$$

设计 1 个符号数组变量把每一次使 $\alpha_t'(j)$ 最大的状态 i 保存下来。

(3)当 $t\neq T$ 时转移到(2)，否则执行(4)。

(4)把这时的终了状态寄存器 $\alpha_T'(N)$内的值取出，根据公式(13)得到：

$$P_{\max}(S,O/M)=\alpha_T'(N) \quad (13)$$

输出的符号数组变量为所求的最佳状态序列。

2 计算复杂度分析与优化

整个连续 HMM 孤立词识别系统的计算过程主要有 3 个组成部分：(1)语音信号的预处理过程；(2)语音信号的特征提取过程；(3)语音识别过程。在整个语音识别系统中，上述 3 个部分的计算量是变化的，它随着系统的运行环境、语音测试集的长短等因素的变化而变化。采用 Baum-Welch 训练算法得到传统的 CHMM 模型后，在 PC 机上做语音识别实验，实验选用 50 条测试集作为要识别的对象，测试集的字数为 2~8 个 (平均字数 4.2 个)。整个语音识别系统耗时为 11.2 s，各部分的运算时间如表 1 所示。

表 1 各部分运算时间 (单位：s)

预处理	特征提取	语音识别
0.045	0.067	11.088

由表 1 可以看出语音识别过程，即 Viterbi 对数概率计算占用了整个识别过程中的绝大部分时间，当测试集的条数越多的时候语音识别所做的匹配运算时间也就越多。上述的语音识别过程所用的是 50 条命令集与待识语音匹配总共耗费的时间。因此在命令词条比较多的情况下，如果能够减少 Viterbi 对数概率计算的计算量，就可以降低整个语音识别系统的响应时间。以下几种方法都是通过减少 Viterbi 对数概率计算，达到计算复杂度优化的效果。

2.1 降低高斯密度的混合数

传统的语音识别系统连续密度的 HMM 状态输出由 3 个单高斯密度分布加权混合构成。一般来说，增加混合的单高斯密度分布的数目，有助于提高语音识别的识别率，但增加的计算复杂度往往很大。对 1 个具体任务的识别系统，可以通过权衡二者的利弊进行特定目的的优化。对于本系统，降低高斯密度的混合数，HMM 状态输出采用单高斯概率分布函数，语音识别过程中概率计算所花费的时间降低到原先的 30 %左右，识别性能的变化甚微。

2.2 降低每个音节的状态数

传统的语音识别系统中给每个音节分配的状态数为 6 个。同样的道理每个音节分配的状态数越多，语音识别的识别率也会越高，计算量也同样会提高。降低每个音节的状态数，HMM 模型中每个音节状态数选择 4 个，语音识别过程中概率计算所花费的时间降低到原先的 68 %左右，识别性能的变化也不大。

2.3 延长待识语音帧移

传统的语音识别系统中待识语音信号的帧移为 10 ms，即每次移动 80 个采样点，再取后面的 256 个采样点作为下一帧语音的原始数据。理论上来说，帧移越短，越能更好地表示待识语音信号的特征。实际上，帧移太小对于语音识别系统的识别率并没有太大的效果。为了提高识别速度，并保证识别率，取了 1 个比较理想的临界值——帧移为 20 ms，此时语音识别过程中概率计算所花费的时间降低到原来的 50 %左右，特征提取过程中的计算也相应减少。特别注意的是，对于某些具体任务的识别系统，帧移还可以适当放宽，但是帧移不宜太长，否则无法有效地表示待识语音的特征矢量。

2.4 待识语音长度与测试集中词条字数的比较

根据语速的测试所得，人们按照正常的语速说话，平均说每个字所花费的时间为 0.238 s 左右。按照正常语速的标准：人们说话的语速不可能高于平均语速的 1 倍，也不可能低于平均语速的 1/2，即正常语速说每个字所花费的时间在 0.119~0.476 s 之间。即如果待识语音通过预处理过程中的端点检测后，得到有效语音的长度为 k 秒，测试集中词条的字数如果小于 $(k/0.476)$个字，或者大于$(k/0.119)$个字都可以作为拒识条件，从而免去 Viterbi 对数概率计算。判断待识语音长度与测试集中每个词条的字数的方法，虽然增加了程序复杂度，但是在一般的测试集中能够有效地减少计算时间，对提高识别率也有一定的效果。

3 实验结果与讨论

根据统计，中文汉字拼音(带声调)共包含 1 261 个音节，本次实验所用训练集数据库是由 40 个男性各读所有音节 2 遍得到。由于数据库中训练的语音有限，最终所得到的识别效果也不算理想。实验所采用的运行环境为 AMD5000+的 PC 机，Windows XP 的操作系统，2 GB

的内存，以 Matlab7.5 为平台的运算工具。通过这些原始语音数据，根据 2.1、2.2 所述，采用 Baum-Welch 训练算法得到优化后的 CHMM 模型；语音识别程序中修改 2.3 所述的参数，并且添加 2.4 所述功能，得到整个语音识别系统耗时为 1.17 s 左右，各部分的运算时间如表 2 所示。

表 2 各部分运算时间 (单位：s)

预处理	特征提取	语音识别
0.027	0.040	1.106

识别率的实验人员为 1 名男性(参加过训练)，在一般的办公环境中，将测试集中 50 条命名词各说了 3 遍，得到在传统的 CHMM 语音识别系统和优化后的语音识别系统的识别率如表 3 所示。

本文的研究具有积极的意义，尤其是在中小词汇量孤立词语音识别系统中，同时为嵌入式平台上的应用(实时实现)提供了理论依据和实践意义。

表 3 识别性能测试结果

总测试集/条	待识语音/条	传统识别系统中识别率/%	优化后识别系统的识别率/%
50	150	94	91.3

参考文献

[1] 姚天任.数字语音处理.武汉：华中科技大学出版社，2003.
[2] 朱民雄.计算机语音技术.北京：北京航空航天大学出版社，2002.
[3] 杜利民，谢凌云，刘斌.HMM 非特定人连续语音识别的嵌入式实现.电子与信息学报，Jan 2005，27(1).
[4] 赵力.语音信号处理.北京：机械工业出版社，2003.
[5] 何强，何英.MATLAB 扩展编程.北京：清华大学出版社，2002.

(收稿日期：2009-04-08)