第 34 卷 第 4 期 计 算 机 工 程 2008 年 2 月
Vol.34 No.4 Computer Engineering February 2008
·人工智能及识别技术· 文章编号：1000—3428(2008)04—0218—02 文献标识码：A 中图分类号：TP301.6

实时说话人辨识系统中改进的 DTW 算法

李邵梅，刘力雄，陈鸿昶
(国家数字交换系统工程技术研究中心，郑州 450002)

摘 要：识别正确率和抗噪性能是语音识别的研究重点，而识别响应速度也是决定系统实用化的关键。文章改进了传统的动态时间弯折算法结构，将其应用于实时说话人辨识系统中，极大地提高了系统运行速度，随着待识别语音数目的增多，该算法优势更加明显。实验表明，在不影响系统识别率的情况下，该方法使系统的运行速度平均提高了 1.5 倍。

关键词：说话人辨识；美尔倒谱系数；动态时间弯折

Improved DTW Algorithm in Real-time Speaker Identification System

LI Shao-mei, LIU Li-xiong, CHEN Hong-chang
(National Digital Switching System Research Center, Zhengzhou 450002)

【Abstract】 Response rate of recognition is a key factor for a speech recognition system as well as recognition correct rate and noise robust property. This paper improves the conventional DTW algorithm's structure, and fastens the running speed in real-time speaker identification. The advantage of the new algorithm is more obvious with the increment of speeches. Experiments show that the method can increase the operation speed by 1.5 times.

【Key words】 speaker identification; Mel-frequency Cepstral Coefficient(MFCC); Dynamic Time Warping(DTW)

1 概述

说话人识别可以看作是模式识别的一种。它对所接收的语音信号进行处理，从中提取相应的特征或建立相应的模型，然后按照一定的判决规则进行识别，是一种根据说话人的语音来判断说话人身份的技术。根据说话人识别的职能，可以分为说话人辨识、说话人确认和说话人探测/跟踪。从识别基于的对象来看，又可以分为基于文本的说话人识别和文本无关的说话人识别[1]。

动态时间弯折(Dynamic Time Warping, DTW)是语音识别中一种简单有效的方法，该算法基于动态规划的思想，解决了发音长短不一的模板匹配问题，是语音识别中出现较早、较为经典的一种算法。在相同环境条件下的语音识别中，DTW算法和HMM算法的识别效果相差不大，但HMM算法复杂得多，这主要体现在HMM算法在训练阶段需要提供大量的语音数据，经过反复计算才能得到模型参数，而DTW算法在训练中几乎不需要额外的计算。因此，在语音识别，尤其是非特定人语音识别中，DTW算法得到了广泛的应 用[2]。在应用DTW算法进行语音识别时，每次都要将测试语音去匹配所有的声纹模型，然后找出最相近模型对应的说话人作为识别结果。这样，随着模型数目的增多，一次识别所花费的时间会直线上升[3]。因此，提高每次匹配的速度是基于DTW的语音识别系统实用化的关键。

本文基于实时说话人辨识，在 DTW 算法的实现中，利用事先建立的区域表缩小了每次进行相似度比较和距离累加的区域，在不影响识别率的情况下，提高了算法的运算速度。

2 DTW 算法原理

假设测试和参考模板分别用 R 和 T 表示，为了比较它们之间的相似度，可以计算它们之间的距离 $D[T, R]$，距离越小则相似度越高。具体实现中，先对语音进行预处理，再把 R 和 T 按相同时间间隔划分成帧系列：

$$R=\{R(1),R(2),\cdots,R(m),\cdots,R(M)\}$$

$$T=\{T(1),T(2),\cdots,T(n),\cdots,T(N)\}$$

然后采用动态规划进行识别。

把测试模板的各个帧号n=1~N在一个二维直角坐标系的横轴上标出，把参考模板的各帧号m=1~M在纵轴上标出，通过这些表示帧号的整数坐标画出的纵横线即可形成一个网格，网格中的每一个交叉点 (n,m) 表示测试模板中某一帧与训练模板中某一帧的交叉点。动态规划算法可以归结为寻找一条通过此网格中若干格点的路径，路径通过的格点即为测试和参考模板中进行距离计算的帧号[4]。如图 1 所示。

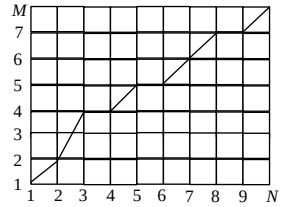

图 1 DTW 算法的搜索路径

为了使路径不至于过分倾斜，可以约束斜率在 0.5~2 内，如果路径已通过了格点 (n_{i-1},m_{i-1})，那么下一个通过的格点 (n_i,m_i) 只可能是下列 3 种情况之一：$(n_i,m_i)=(n_{i-1}+1,m_{i-1}+2)$；$(n_i,m_i)=(n_{i-1}+1,m_{i-1}+1)$；$(n_i,m_i)=(n_{i-1}+1,m_{i-1})$。用 η 表示上

基金项目：国家自然科学基金资助项目(60372038)
作者简介：李邵梅(1982－)，女，硕士，主研方向：通信与信息系统；刘力雄，工程师；陈鸿昶，教授
收稿日期：2007-03-29 **E-mail：**lishaomei_may@126.com

述 3 个约束条件，求最佳路径的问题可以归结为：满足约束条件 η 时，求最佳路径函数 $m_i=\bar{\phi}(n_i)$，使得沿路径的累积距离达到最小值。

从以上分析可以看出，整个算法主要归结为计算测试帧和参考帧间的相似度以及所选路径的矢量距离累加。

3 DTW 的高效算法

3.1 现有的 DTW 高效算法

DTW 算法中，由于匹配过程中限定了弯折的斜率，因此很多格点实际上是到达不了的，如图 2 所示，菱形之外的格点对应的帧匹配距离无须计算。

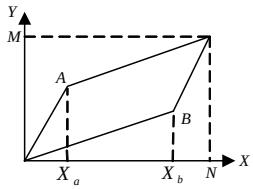

图 2　匹配路径约束示意图

文献[2]提出把实际的动态弯折分成 3 段：$(1, X_a)$，(X_a+1, X_b) 和 (X_b+1, N)，其中，$\begin{cases} X_a=\frac{1}{3}(2M-N) \\ X_b=\frac{2}{3}(2N-M) \end{cases}$，$X_a$ 和 X_b 都取最相近的整数。由此得出对 M 和 N 长度的限制条件：

$$\begin{cases} 2M-N\geq 3 \\ 2N-M\geq 2 \end{cases}$$

不满足以上条件时，认为两者差别太大，无法进行动态弯折。

文献[2]中提出，在 X 轴上的每一帧不再需要与 Y 轴上的每一帧进行比较，而只与 Y 轴上 $[y_{\min}, y_{\max}]$ 间的帧进行比较，其中，$y_{\min}$ 和 $y_{\max}$ 的计算如下式：

$$y_{\min}=\begin{cases} \frac{1}{2}x & 0\leq x\leq X_b \\ 2x+(M-2N) & X_b<x\leq N \end{cases}$$

$$y_{\max}=\begin{cases} 2x & 0\leq x\leq X_a \\ \frac{1}{2}x+(M-\frac{1}{2}N) & X_a<x\leq N \end{cases}$$

也可能出现 $X_a>X_b$ 的情况，此时弯折匹配的 3 段为：$(1, X_b)$, (X_a+1, X_b) 和 (X_a+1, N)。

3.2 改进的 DTW 高效算法

3.1 节中的高效算法缩小了进行相似度判断的测试帧和参考帧的范围及求所选路径的矢量和累加的范围，即减少了计算 X 轴上测试帧和 Y 轴上参考帧矢量距离的次数和计算累加矢量和的次数，从而提高了匹配速度，但计算范围并没有缩到最小。如图 3 所示，原有的 DTW 算法的相似度计算和矢量距离累加的区域为虚线所围区域，改进后的计算区域如竖线部分所示，但是真正需要参与相似度计算和矢量距离累加的区域如横线部分所示。可以清楚地看出，3.1 节的方法只对计算过程进行了部分优化。

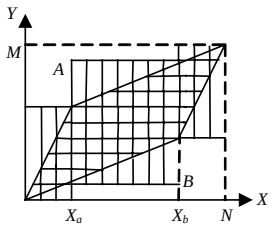

图 3　DTW 算法中的计算区域

另外，3.1 节的方法是针对非实时的语音识别的，识别过程中要事先计算测试语音 N 和参考语音的总帧数 M，然后再根据 N 和 M 的值计算匹配的区域。在实时语音识别系统中，受处理时间的限制，不能等全部语音接收完后再开始计算区域进行判别，而必须边接收边作 DTW 处理。假设 1 帧的接收时间为 x ms，且缓存 n 帧进行一次 DTW 处理，那么 n 帧的处理时间必须小于 nx ms，否则无法满足实时性的要求。

利用实时处理中分批进行固定的 N 帧和 M 帧匹配的特点，可以进一步把计算区域缩小到图 3 中的横线部分区域，最大限度地提高 DTW 算法的识别速度。本文由此提出了基于查表法的相似度计算和矢量距离累加。

因为 N 和 M 已知，所以可以首先离线构造一个表 *Table*[]，此表由二元数组对 {*TestNum*, *RefNum*} 组成。每个二元数组表示在进行相似度计算和矢量距离累加时用到的测试帧的序号和参考帧的序号。这样，在应用 DTW 进行运算的过程中，只须按照 *Table* 中的二元数组对依次计算，保证只有横线区域的测试帧和参考帧对参与计算，节省了计算时间。假设 N=4, M=4，则 *Table*[]={{2,1},{2,2},{2,3},{3,2},{3,3},{3,4}}。在进行相似度计算和矢量距离累加时，只须计算 *Table*[]中的 6 对测试帧和参考帧，而省去了 (4×4-6=10) 对的计算时间。由于 *Table*[]表是事先建的，而查表过程通常只有一个指令周期，运行时间可以忽略不计，因此相对于 2.1 节的高效算法，本算法省去了图 3 中 6 个斜线三角区域的相似度计算和矢量距离累加。

4 实验及结论

本文构建了固定文本的实时说话人实时辨识系统，在 CCS下对改进的DTW算法进行了仿真试验[5]。实验中，采用录音设备录制了 10 个人 3 个不同时间所说的同一句话，共 20 句，平均时间长度为 4.5 ms。从每个人的 3 句话中选择频谱最清晰的一句训练成模板，这样模板库中有 10 个模版，待识别的语音有 20 个。语音预处理中，每帧的长度选为 20 ms。特征参数选择目前运用最广泛的美尔倒谱系数 (Mel-frequency Cepstral Coefficient, MFCC)过零率和能量参数，其中，N和M的取值均为 64。实验结果见表 1。

表 1　试验数据

	耗时/ms	
	模版数=5	模版数=10
原有的 DTW 算法	188	467
3.1 节的改进算法	156	428
本文的改进算法	125	313

从实验结果可以看出，在固定文本的说话人辨识中，本文的 DTW 算法在很大程度上节省了识别时间，并且随着待识别语音的增多，节省效果更加明显。

参考文献

[1] 俞一彪, 王朔中. 基于互信息匹配模型的说话人识别[J]. 声学学报, 2004, 29(5): 462-466.

[2] 何　强, 何　英. Matlab 扩展编程[M]. 北京: 清华大学出版社, 2002-06.

[3] 刘文举, 孙　兵, 钟秋海. 基于说话人分类技术的分级说话人识别研究, 电子学报, 2005, 33(7): 1230-1233.

[4] 崔光照, 吴晓平, 路　康. 基于改进的 DTW 算法的仿真与分析, 福建工程学院学报, 2004, 2(2): 149-151.

[5] 李鹏怀, 徐佩霞. 基于DSP 的嵌入式语音识别系统的实现[J]. 计算机工程, 2005, 31(16): 160-162.