文章编号:1672 - 058X(2012)05 - 0068 - 05

一种用于语音识别的高效分帧函数的研究*

赵明明, 王洪春
(重庆师范大学 数学学院,重庆 400047)

摘 要:模式识别技术迅速发展,要求相应的模式识别算法仿真和验证简单方便;在语音处理中,基于声管模型的提取特征时,大多时候要把得到的一个特征序列分成若干个子序列,有的还需要每个序列间有重叠部分,利用 MATLAB 编写一个分帧函数便使得这个工作变得异常简单;要用好这个函数,需要详细了解此函数的各个参数的意义;然后通过此函数的解读,还可以在需要时,适当的改造函数,当然也可以将其中一些算法应用到其他领域中。

关键词:人工智能;模式识别;MATLAB;分帧函数;enframe
中图分类号:TP372 **文献标志码**:A

虽然现在语音识别系统已经小规模应用,比如安卓操作系统上的 Voice Actions,IPHONE 上的语音 SIRI 语音智能助手,还有最近 QQ2011 上的多功能辅助输入的语音识别输入方法,但经过使用就会发现,这些系统虽然有个框架,听起来很智能,但效果并不很理想,使用起来舒适度也不高。大多数时候还会选择更快速的按键输入,或者更方便的手写输入方法。当今语音识别技术并没实现人与机器的自由交谈,语音识别的功能上仍有很大发展空间。此项技术仍需要科研人员从语音识别的各个环节,各个模块入手,来尽快实现技术突破。

分帧加窗函数可用于研究语音识别的预处理模块中。在语音识别研究中,声道模型基本上是基于声管模型而建立的,具有短时平稳性,也就是说长长的一段语音信号波形实际上是由很短的小段的平稳信号构成,基于此特性,要提取到语音信号的频域特征,首先需要把信号进行分帧研究,帧长范围为 10 ~ 30 ms,为了建立仿真模型,首先需要一个快捷方便又好用的工具,自动对较长的语音信号进行分割保存,并采取适当的滤波去噪声,使得研究人员直接进行特征提取等后续工作。在此利用 MATLAB 集成仿真系统,建立一个实现此功能的函数。

1 完整的函数以及流程

大多数研究在处理语音识别问题时,对分帧函数总是解释的不够清楚明了,不能满足想了解语音识别详细细节的读者的要求,在 MATLAB 中对分帧函数进行详细的研究,并且对分帧的各个过程做了详细解读。

函数具体写法如下:

Function f = enframe(x, win, inc)

收稿日期:2011 - 10 - 18;**修回日期**:2011 - 11 - 23.
* **基金项目**:重庆市自然科学基金项目(CSTC2011BB2116).
作者简介:赵明明(1986-),男,河南洛阳人,硕士研究生,从事模式识别研究.

```
%    F = ENFRAME(X,LEN) splits the vector X(:) up into frames, Each frame is of length LEN and occupies one row of the output matrix. The last few frames of X will be ignored if its length is not divisible by LEN. It is an error if X is shorter than LEN.
nx = lenth(x(:));
nwin = length(win);
if(nwin = = 1)
   len = win;
else
   len = nwin;
end
if(nargin < 3)
   inc = len;
end
% 得到要分出的帧
nf = fix((nx-len + inc)/inc);   % fix 的意思向零方向取整
f = zeros(nf,len);% 生成 nf * len 的全零矩阵
% indf = inc * (0:(nf-1)).';% 得到分开后每一帧的端点
inds = (1:len);
f(:) = x(indf(:,ones(1,len)) + inds(ones(nf,1)));
   % 取 x 的某元素到制定位置来得到一个新的矩阵.
if(nwin > 1)
   W = win(:)';% 转置
   F  = f. * w(ones(nf,1),:);
end
```

2　输入输出参数分析

首先分析此函数的输入输出参数的名字、功能、以及如何使用? Function f = enframe(x,win,inc)是函数的声明或者定义,f 为函数 enframe( )的返回值,enframe(x,win,inc)为函数的表示方法,即在其他地方调用时的格式。x,win,inc 这 3 个参量是函数的输入参数,以下对 3 个输入参数逐一说明:x 是要被分帧的向量,在此为了方便下面论述,举例说明:x = [1　2　3　4　5　6　7　8　9　…　9 999　10 000]这样一个行向量,设它一共有 10 000 个元素,第一个为 1,最后一个为 10 000,每个元素之间间隔 1。那么分帧的意思就是说要把这个有 10 000 个元素的行向量分成小于 10 000 的若干个行向量,具体怎么分,还要看下面两个参数的值。x 是指由语音的抽样信号而来的一维向量。

WIN 是规定了所需要得到帧的长度,即把 x 分帧后每一帧的长度,也即每帧包含的元素个数,即帧长,是个标量;再就是表示一个矢量,即窗向量。知道语音识别的预处理通常包括预加重、分帧、加窗 3 个过程,这个窗向量 *WIN* 是要与每一帧的数据做内积的,在此就能完成加窗的功能。窗函数的长度就刚好是帧长度,也就是说这个参数至少可以表示每一帧的长度。一旦在输入的时候把一个向量输入 *win* 所在的位置,那么它就表示一个窗向量,此函数的功能就是在对语音抽样信号分帧后再加窗,然后输出。

INC 表示滑动范围,即帧与帧之间的重叠量,帧与帧并不是手拉手的,终点连起点,而是有重叠的。当然也可能是有间隔的。具体要分出怎样的帧,调整 *inc* 为合适的值就可以了。

3 分帧原理及其实现

函数体下面是对这个函数的简单说明。每行前面都有%,是注释符号,意思是:形如 F = ENFRAME(X, LEN)的函数,把向量 $X(:)$分成许多帧,每一帧的长度 LEN,输出矩阵中的一行就是一帧,如果最后一帧的长度小于 LEN,那么这一帧将被抛弃。另外,LEN 不能小于 X 的长度,否则函数运行时会报错。

```
%   F = ENFRAME(X,LEN) splits the vector X(:) up into
%   frames, Each frame is of length LEN and occupies
%   one row of the output matrix.  The last few frames of
%   X will be ignored if its length is not divisible by LEN.
%   It is an error if X is shorter than LEN.
```

对上面翻译的说明:本程序的函数是早期版本,显然它没有表示重叠的参数,只能是最典型的一种分帧。因此如果要对改造本函数,最好把改造情况在此说明。Nx = length($x(:)$);得到向量 X 的长度,并保存在变量 NX 中,以便下面的程序使用。$nwin$ = length(win);得到 win 的长度,即帧的长度。

```
If(nwin == 1)
    Len = win;
Else
    Len = nwin;
End
```

这是一个 if 判断语句,$nwin$ 是上面得到的 win 的长度,如果长度为1,说明 WIN 表示的是所需要分帧的长度,如果不为1,说明 WIN 表示的是窗函数,那么窗函数的长度即为帧长。正是这个选择语句,让 win 参数有了两个意思,可表示帧长,也可表示窗向量。

```
If(nargin <3)
    Inc = len;
End
```

增强了本函数弹性,nargin 是一个默认参数,它提供了它所在的函数所拥有的输入参数个数。可以这样理解:nargin 表示输入的参数数量。那么本句的意思是如果输入的参数数量小于3,即参数数量为0,1,2 等情况时,把长度的值赋予滑动变量,即表示典型的无重叠的分帧函数。nf = fix((nx-len + inc)/inc);此句得到要分出的帧数 $FIX()$ 函数的意思:向零方向取整,nx 是向量 x 的长度设为100,那么如果要分帧的帧长为20,每帧的重叠量为10,那么一共能把 X 分多少帧呢?(100 - 20 + 10)/10 = 9,一共能分9帧,$fix()$函数就是计算可以分为多少帧的公式,那么为何要加一个 $fix()$函数,试想,如果向量 X 的长度不是刚好为100,而是102,显然要把小数点后这0.2截去,那么 $fix()$函数就是做这个用的,即$fix(9.2) = 9$;F = ZEROS(NF, LEN),此句生成 $nf * len$ 的全零矩阵,$indf$ = inc * (0:(nf - 1))。

得到分开后每一帧的端点。比如说(0:10)是指这样一个行向量:[0 1 2 3 4 5 6 7 8 9 10],那么3 * (0:10)就等于[0 3 6 9 12 15 18 21 24 27 30],那么3 * (0:10),就是把行向量变为列向量。因此 $indf$ = inc * (0:(nf - 1))的意思就刚好得到分开后每一帧的起点的序号,放在列向量 $indf$ 中,元素个数有 nf 个。

$Inds$ = (1:len);得到一个拥有 len 个语句的行向量 $f(:)$ = x($indf$(:, ones(1, len)) + $inds$(ones(nf, 1), :));此句是整个函数的核心算法,也体现了 MATLAB 在做矩阵运算时的简便与快捷。此句比较长,从左

到右依次解释,$f(:)$是个矩阵,用来存放将输入序列分帧后所行成的矩阵,行数等于帧数,列数等于帧长。在等号后面,向量 $x()$里有两部分内容相加,其实是两个矩阵相加,$indf(:,ones(1,len))$是指保持列向量 $indf$ 每一行的元素不变,将其扩充为 len 行,这样就得到一个帧数 * 帧长的矩阵;$inds(ones(nf,1),:)$是保持行向量的每列元素不变,将其扩充为 nf 列,所以它也得到一个帧数 * 帧长的矩阵,那么两者相加就刚好这样的一个序号矩阵,即分帧后每一帧的元素在序列 x 中的序号所组成的矩阵,那么将这样一个矩阵放入 $x()$中,在此便得到一帧数 * 帧长的矩阵,每一行为一帧,行数即为帧数,至此分帧完成。

4　加窗原理以及实现

对语音信号的加窗处理在计算上也就是将时间序列与窗函数相乘,设某有限长窗函数:

$$W_N(t) = \begin{cases} w(t) & -N/2 \leqslant t \leqslant N/2 \\ 0 & \text{其　他} \end{cases}$$

设语音信号为 $S(t)$,那么对其加窗可以这样表示:

$$X_N(t) = S(t)W_N(t)$$

那么编写程序就可以用此公式来编写,写程序前,先来推导这样做对信号频谱的影响。

按照卷积定理,$f(x) * h(x) \xrightarrow{\text{适当变换}} F(\omega)H(\omega)$　$f(x)h(x) \xrightarrow{\text{适当变换}} F(\omega) * H(\omega)$, * 表示做卷积。

两个连续函数在空间域中的卷积可求其相应的二个傅立叶变换乘积的反变换而得。反之,在频域中的卷积可用的在空间域中乘积的傅立叶变换而得。这一定理对拉普拉斯变换、双边拉普拉斯变换、Z 变换、Mellin 变换和 Hartley 变换等各种傅立叶变换的变体同样成立。那么的信号 $s(t)$加窗后的频谱函数:

$$F(\omega) = W_N(\omega) * S(\omega)$$

因此,信号经过加窗后,频谱会与原信号的频谱有改变,这个改变完全由窗函数的频谱决定。因而可以依信号处理实际情况选择相应的窗函数。加窗主要是为了减少频谱泄漏。因为傅立叶变换对应的是无限信号,或者无限长的信号,经过分帧后,信号的长度缩短为原来的 $1/N$,$N = \text{size}(win)/\text{size}(x)$,因此原来的信号就变成了一帧一帧的有限信号,此时进行傅立叶变换,高频部分将有“泄露”,因而需要加窗以防止高频泄漏。加窗这种运算和变换一般用 $T[\cdot]$表示。但另一方面,窗函数本身的频谱在高频部分是截止的,所以每帧信号加窗后的傅立叶变换,频谱肯定“泄露”,但相比不加窗的信号,高频部分缺失现象有所改善。程序如下:

```
If(nwin > 1)
    w = win(:)';
    F = f. * w(ones(nf,1),:);
End
```

在 if 语句中,根据输入参数的特点来判断是否加窗处理;语句 $w = win(:)'$是指将列向量 win 变为行向量,保存在 w 中;那么 $w(ones(nf,1),:)$;的意思是将 w 扩展成一个帧数 * 帧长的矩阵,这个矩阵每一行是个行向量 w,此矩阵与分帧后的矩阵内积后便得到分帧加窗后的分帧数据,保存在矩阵 F 中。

5 小 结

对语音信号处理中预处理部分的分帧函数做了详细研究,并对一般的分帧函数功能做了扩充,能根据实际情况完成自动判断是否需要加窗处理,当然也可以进行后续更多的处理,比如根据语音识别后续处理的效果来自动调节帧长、帧移、甚至窗函数,让其更加智能化,增强整体语音识别的鲁棒性。

参考文献:

[1] 边肇祺,张学工. 模式识别[M]. 2版. 北京:清华大学出版社,2000

[2] 王炳锡,屈丹,彭煊. 实用语音识别基础[M]. 北京:国防工业出版社,2005

[3] MUTHUSAMY Y K, Jain N, Cole R A. Perceptual benchmarks for automatic language identification[J]. ICASSP, 1994b, 1: 333-336

[4] ITAHASHI S, DU L. Language identification based on speech fundamental frequency[J]. Eurospeech. 1995(2): 1359-1362

[5] 张雪英. 数字语音处理及MATLAB仿真[M]. 北京:电子工业出版社,2010

[6] RAFAEL C. Gonzalez. Digital Image Processing Using MATLAB[M]. 北京:电子工业出版社,2005

[7] KALURI V. Rangarao. Digital Signal Processing: A Practitioner's Approach[M]. 西安:西安交通大学出版社,2007

[8] 马志欣. 语音识别技术综述[J]. 昌吉学院学报,2006(5):56-59

[9] 赵力. 语音信号处理[M]. 2版. 北京:机械工业出版社,2010

A Study of High Efficient Frame Function Applied to Phonetic Recognition

ZHAO Ming-ming, WANG Hong-chun

(School of Mathematics, Chongqing Normal University, Chongqing 400047, China)

Abstract: Pattern recognition develops rapidly, which requires simple and convenient corresponding algorithm simulation and verification of pattern recognition. In speech processing, when we extract features based on sound tube model, an obtained feature sequence is divided into several subsequences most of time, some times each overlapped part is needed between each sequence, which is very easy if MATLAB is used to write a frame function. To have a good command of this function needs to know in detail the meanings of each parameter of this function, through the digest of this function, this function can be properly transformed when necessary, of course, some of the algorithms of the function can be applied to other fields.

Key words: artificial intelligence; pattern recognition; MATLAB; frame function; enframe

责任编辑:代小红