6T-9

組込型音声認識システムのための低演算特徴量抽出法*

松井清彰
東北大学工学部†

伊藤彰則
東北大学大学院工学研究科‡

1 はじめに

近年，音声認識システムの発展に伴い，アップル社のSiriやNTTドコモのしゃべってコンシェル等，音声認識機能を搭載した機器が現れてきた [1]．しかし，これらの機器は，外部のサーバと通信を行う分散型音声認識を使用しているため，高速な通信環境が必要となる．そこで，外部との通信の必要がなく，全ての処理を自身で行える組込型音声認識システムの実現が望まれている．そのためには，低演算で行える音声認識手法が必要である．本論では，特徴量抽出を低演算で行う手法に関して考察する．

2 特徴量抽出

音声認識システムは，いくつかのステップで成り立っている．その中でも最初のステップが，特徴量抽出である．特徴量抽出は，その名の通り，入力発話の何かしらの特徴量を得ることで，コンピュータによる音声の解析を可能にするプロセスである．用いられる特徴量としては，MFCC(Mel Frequency Cepstrum Coefficient)と呼ばれるものが一般的に用いられている．MFCCは，メル尺度上で等間隔に配置されたフィルタバンクを用い，フィルタバンク分析を行い，各帯域におけるパワーを離散コサイン変換することで得られる係数であり，音声の周波数スペクトル概形を少ないパラメータで効率的に表現することができる．通常，低次の成分のみを用い，低次より12次元＋帯域パワー，さらにそれらの時間変化を表すΔ係数，ΔΔ係数を加えた39次元が用いられる．

3 Haar-Wavelet 変換

今回，MFCCに代わる手法として提案しているのがHaar-Wavelet変換を用いた特徴量抽出である．Haar-Wavelet変換は，入力音声を低周波成分と高周波成分に分けるフィルタの一種である．画像処理の分野において，圧縮等によく用いられる [2]．入力を $x_0^{(0)}, x_1^{(0)}, x_2^{(0)} \cdots$

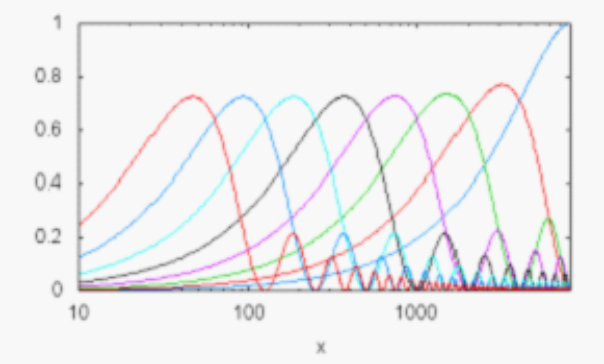

図 1: Haar フィルタバンクの周波数応答

とすると，

$$y_n^{(k+1)} = \frac{x_{2n}^{(k)} - x_{2n+1}^{(k)}}{2} \quad (1)$$

$$x_n^{(k+1)} = \frac{x_{2n}^{(k)} + x_{2n+1}^{(k)}}{2} \quad (2)$$

という式に従い，$y_n^{(1)}, y_n^{(2)}, \cdots$ を求めていくが，$y_n^{(1)}$ は入力信号のサンプル数の半分，$y_n^{(2)}$ は $\frac{1}{4}$ というように，サンプル数が半分に減少していく．そのため，サンプル数は2の累乗になるようとる．このようにして $y_n^{(k)}(k = 1, \cdots, K)$ を求めていき，帯域パワーを求める．Haar-Wavelet によるフィルタバンクの周波数応答を図1に示す．

バンドパスフィルタ出力の対数をとり，離散コサイン変換を行うことにより，MFCCと似たパラメータを作ることができる．Haar-Wavelet変換を用いる理由として，下記のようなものが挙げられる．

演算が容易である 式 (1), (2) の計算は，全て加減法及びビットシフトのみで実現できるため，計算が単純である．これに対して，MFCCの計算では離散フーリエ変換を必要とするため，組み込み機器に対しては計算量が高い．

計算量が少ない 入力サンプル数をnとすると，全体の計算量はO(n)となり，MFCCより高速である．また，対数をとる際に，底を2とすると，整数に対する高速な対数演算アルゴリズムが存在する [3]．また，DCTにも，高速なアルゴリズムが存在する [4]．

*Low-Complexity Feature Extraction for Embedded Speech Recognition
†Kiyoaki Matsui (Tohoku University)
‡Akinori Ito (Tohoku University)

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.

表 1: MFCC(26 次元) の認識率 (%)

	学習回数								
分布数	4 回	6 回	8 回	10 回	12 回	14 回	16 回	18 回	20 回
2	83	83	83	80	80	83	83	83	83
4	83	83	83	83	83	83	87	87	87
8	90	90	87	87	90	83	83	80	80
16	90	90	90	87	87	87	90	87	90
32	87	83	87	87	87	83	83	87	80
64	77	87	83	83	77	77	77	77	77

表 2: Haar-Wavelet 特徴量 (16 次元) の認識率 (%)

	学習回数								
分布数	4 回	6 回	8 回	10 回	12 回	14 回	16 回	18 回	20 回
2	60	63	63	63	63	60	60	57	60
4	67	67	63	63	67	63	63	67	60
8	70	70	67	70	67	67	63	70	60
16	67	70	77	77	73	73	70	67	63
32	67	63	67	67	67	60	60	60	57
64	60	67	67	63	60	57	57	57	57

4 実験と考察

今回，Haar-Wavelet 変換を用いた特徴量の認識率を，簡単な実験により調べた．

4.1 実験条件

今回認識に用いたのは，ATR データベースより，男性 2 名，女性 1 名の，計 3 名による，0 から 9 までの数字の発話である．用いる音響モデルを，一方は MFCC，もう一方は Haar-Wavelet 変換による特徴量を用いて学習したモノフォン HMM を用いた．学習データとして，日本語話し言葉コーパス [5] より，1.300 個の発話を用いた．Haar-Wavelet 変換による特徴量として，1 フレームあたり 256 点のサンプルから 8 次元のバンドパスフィルタ出力パワーを計算し，Δ 特徴量と合わせて，16 次元とした．MFCC は，Haar-Wavelet に条件を近付けるため，Δ Δ 特徴量は用いず，26 次元とした．音声認識エンジンとして Julius を用いた [6]．また，HMM 学習の繰り返し回数とモノフォン HMM の分布数を変えながら，同様の実験を行った．

4.2 結果と考察

実験の結果，MFCC の認識率は，図 1 のように，Haar-Wavelet による特徴量の認識率は，図 2 のように変化した． この図から，繰り返し数や分布数が大きいと，逆に認識率が下がる傾向にあることが読み取れるが，これは，学習に用いたデータ数が少ないため，過学習を起こしてしまっているためと考えられる．Haar-Wavelet による特徴量についても，同様の傾向がみられたが，認識不能となる例がいくつかあり，認識率も MFCC よりは低くなった．Julius のデコーダでは，探索の結果ノードをすべて切り捨ててしまい，候補となる出力がなくなってしまうことが原因であったが，全探索を行っても，この問題を解決できなかった．MFCC では，女性の発話に対しての認識率が悪かったが，これは，学習データが男性発話のみであったため，女性の発話に対し上手く適応しなかったからであると考えられる．

5 まとめ

Haar-Wavelet 変換を用いた特徴量の認識率を実験により調べたところ，MFCC に劣るものの，ある程度の認識率を得ることができた．しかし，認識不能となる例がいくつかあり，それらを解決することはできなかった．この認識不能の場合に関して，今後取り組む他，特徴量抽出にかかる時間が，MFCC に比べどれだけ速くなっているかを組込機器上で調べていきたい．

参考文献

[1] 古井貞熙，“人と対話するコンピュータを創っています -音声認識の最前線-”，角川学芸出版，(2009)
[2] C. Mulcahy, “Image compression using the Haar wavelet transform,” Spelman Science and Mathematics Journal, vol. 1, no. 1, pp. 22–31 (1997)
[3] N. Jones, “Integer Log functions”, http://embeddedgurus.com/stack-overflow/2008/05/integer-log-functions，(2008)
[4] N. Brahimi, “An efficient fast integer DCT transform for images compression with 16 additions only”, Proc. Int. Workshop on Systems, Signal Processing and their Applications (WOSSPA), pp.71–74，(2011)
[5] K. Maekawa, “Corpus of Spontaneous Japanese: its design and evaluation,” Proc. Workshop of Spontaneous Speech Processing and Recognition, (2003).
[6] 名古屋工業大学 Julius 開発チーム，大語彙連続音声認識エンジン julius，http://julius.sourceforge.jp/

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.