卒業論文

HMMを用いた音素認識の高精度化に関する研究

～学習サンプルが少ない場合の出力確率の平滑化手法～

東北大学工学部　情報工学科
阿曽研究室　Ｂ４
鈴木　基之

目 次

第 1 章 序論 1
- 1.1 研究の背景 . . . 1
- 1.2 研究の目的 . . . 1
- 1.3 本論文の構成 . . . 1

第 2 章 HMMを用いた音声認識の原理 2
- 2.1 音声認識の概要 . . . 2
- 2.2 HMMのパラメータの推定 . . . 3

第 3 章 学習サンプルが少ない時の問題点 5
- 3.1 問題点 . . . 5
- 3.2 平滑化前での認識実験 . . . 6

第 4 章 出力確率の平滑化 7
- 4.1 出力確率の平滑化手法 . . . 7
  - 4.1.1 手法 1 . . . 7
  - 4.1.2 手法 2 . . . 7
  - 4.1.3 手法 3 . . . 8
- 4.2 認識実験 . . . 9
  - 4.2.1 使用データ . . . 9
  - 4.2.2 実験方法 . . . 9
  - 4.2.3 パラメータの決定 . . . 10
  - 4.2.4 重みのかかり方 . . . 11
  - 4.2.5 生起確率の分散 . . . 12
- 4.3 結論 . . . 14

第 5 章 手法 1、2の改良 15
- 5.1 改良点 . . . 15
- 5.2 認識実験 . . . 15
  - 5.2.1 打ち切りの個数の決定 . . . 15
  - 5.2.2 生起確率の分散 . . . 16
- 5.3 結論 . . . 16

第 6 章 まとめ 17

第 7 章 今後の課題 18
- 7.1 距離による平滑化の改良 . . . 18
- 7.2 出力確率平滑化のその他の方法の検討 . . . 18
- 7.3 その他の課題 . . . 19

謝辞 20

参考文献 21

図目次

1 *Left-to-Right* モデル 2
2 HMM「東京」が出力する系列と出力確率 2
3 未知入力「toookyooooo」の認識 2
4 HMMの例 3
5 $\alpha(i,t)$ の計算 3
6 $\beta(i,t)$ の計算 3
7 新しいシンボルを含む系列の認識 5
8 平滑化を行なう前の認識結果 6
9 *Left-to-Right* モデル 9
10.1 パラメータの決定：手法 1 10
10.2 パラメータの決定：手法 2 10
10.3 パラメータの決定：手法 3 10
11.1 手法 1での重み 11
11.2 手法 2での重み 11
12.1 生起確率の分散：手法 1 12
12.2 生起確率の分散：手法 2 12
12.3 生起確率の分散：手法 3 12
12.4 生起確率の分散：充分に学習したHMM 13
13 実験結果 14
14.1 打ち切りの個数の決定：手法 1 15
14.2 打ち切りの個数の決定：手法 2 15
15.1 生起確率の分散：手法 1 16
15.2 生起確率の分散：手法 2 16

第1章　序論

1.1　研究の背景

近年、私達のまわりに電気機器があふれ、これらが生活に密着するようになってきた。これらの機器は現在ではほとんどがボタンを押すことで操作しているが、便利さを追及するあまり、これらの機器が多機能になればなるほどボタンの数は増え、操作は複雑になってしまう。その結果一部のユーザだけが恩恵をうけ、大多数の人々にとってはかえって使いにくいものになる。これでは本末転倒であるので、いかに多機能でありながら、誰にでも簡単に使えるようにするか、ということが課題になってきた。

操作を簡単にする手段として考えられるのは、音声による入力であろう。それも、制限された単語のみでは操作が簡単になったとは言えないが、機械が自然に発声された(制限のない)音声を理解できるのならば、小さい子供からお年寄りまでなんの知識もいらずに機械を使うことができる。つまり、誰でもが、あたかも人間に話すように機械に命令することができるのである。これが、究極のマンマシンインターフェースといえよう。

しかし、現在の認識技術では、人間が機械相手だということを頭において、限られた言葉をゆっくりと話さなければ認識は難しい。そこで、語彙を限定しない自然発声された音声を高精度に認識できるシステムが望まれているのである。

1.2　研究の目的

音声認識の手法としては数々のものが提案されており、それぞれに長所、短所があるが、本研究では、数ある認識手法のなかで有力といわれているHMM(*HiddenMarkovModels*)を取りあげる。

HMMを使った認識の詳細は第2章で述べるが、簡単にいうと1つの認識単位(ここでは音素)の音声モデルとしてHMMを用い、認識させたい音素数だけHMMを用意しておく。そして、未知入力に対して一番尤度の高いモデルに対応する音素を認識結果とするものである。

ここで、当然重要になってくるのがHMMをいかによいモデルにするか、つまりHMMのパラメータをいかに音声にあうように推定するか、ということである。ところが、現在一般的に使われている推定アルゴリズムである*Forward – Backward*アルゴリズムでは多量の学習サンプルを必要とするため、学習サンプルが少ないとパラメータの推定に偏りができ、認識率がいちじるしく低下してしまう。

そこで、本研究では学習サンプルが少ない時でも認識率が落ちないようにHMMのパラメータを修正する手法を提案する。

1.3　本論文の構成

第1章は序論であり、研究の背景や目的を述べる。
第2章では、HMMを用いた音声認識の原理について述べる。
第3章では、学習サンプルが少ないときの問題点を指摘する。
第4章では、平滑化の手法を提案し、それを用いた認識実験の結果について述べる。
第5章では、第4章の結論をもとにして更に手法を改良し、それを用いた認識実験の結果について述べる。
最後に第6章で、第4章,第5章での結論をまとめ、第7章で、今後の課題について述べる。

第 2 章　HMMを用いた音声認識の原理

HMMを用いた音声認識の手法はいろいろ提案されているが、ここではそれらのもととなる一番簡単な手法について説明する。

なお、実際の認識実験では音素を認識単位としているが、ここでは簡単のため単語を認識単位として説明する。

2.1　音声認識の概要

HMMは、図1のような、いくつかの状態と、それらの間の遷移枝からなる。それぞれの遷移枝には遷移確率a_{ij}とシンボルの出力確率b_{ij}というものが決められてあり、その確率に従って初期状態~通常は左端~から最終状態~通常は右端~に向けて遷移していく。遷移するたびにひとつずつシンボルを出力し、観測者はそれだけを見ることができる。つまり、どこの状態にいるか、といった状態遷移の様子はわからない。これが、*Hidden* といわれる所以である。

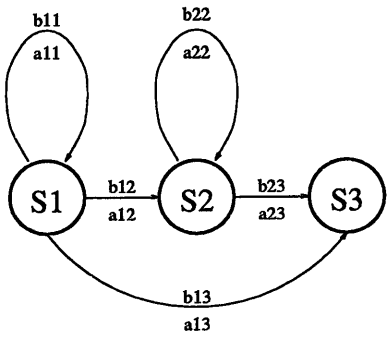

図 1: *Left-to-Right* モデル

さて、このHMMを音声認識にどう使うかというと、ひとつのHMMをひとつの単語~例えば「東京」~に対応させ、そのHMMがその単語音声のシンボル系列~例えば「toookyooo」~を高い確率で出力するように遷移確率や出力確率を推定していく。(図2参照)

そして、認識の時はそのようなHMMを認識させたい単語の数だけ用意し、認識したい未知入力~例えば「toookyooooo」~と同じシンボル系列を出力する確率を各HMMについて計算し、その値の最も高いHMMに対応する単語を認識結果とする。(図3参照)

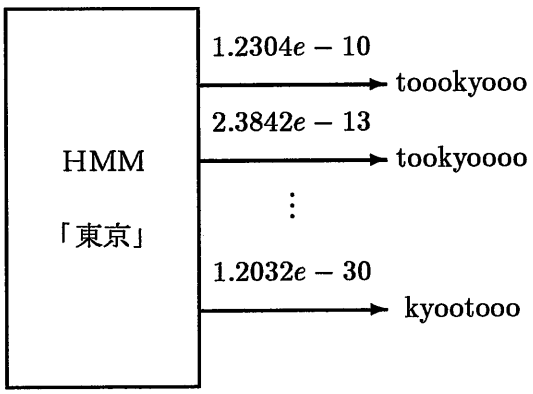

図 2: HMM「東京」が出力する系列と出力確率

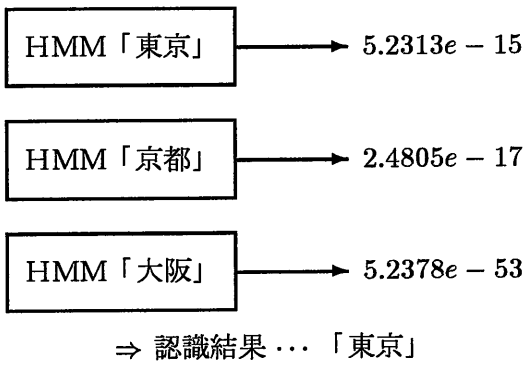

図 3: 未知入力「toookyooooo」の認識

2.2　HMMのパラメータの推定

HMMを使って音声認識をしようとしたときに、HMMのパラメータをどうやって決めるか、ということが一番の問題であるが、一般的には、*Forward-Backward* アルゴリズムが使われる。このアルゴリズムはHMMがある学習サンプルを出力する確率を尤度としてパラメータを最尤推定しよう、というもので、*Baum* らによって提案された。

以下では、このアルゴリズムを簡単に説明する。なお、S_iは各状態、a_{ij}は状態S_iから状態S_jへの遷移確率、$b_{ij}(k)$は状態S_iから状態S_jへ遷移するときにシンボルkを出力する出力確率、$\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$は学習用の音声シンボル系列、$t$は時間である。

また、例として、シンボル数2 (a, b) とし、図4のような初期確率をもつHMMを考える。

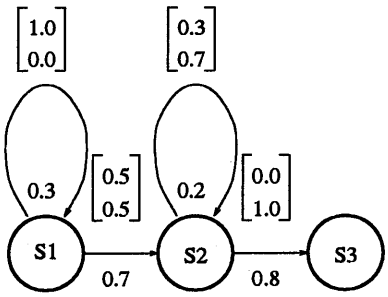

図 4: HMMの例

学習データとして、「a b b」が入力されたとしよう。まず始めに、式(1), 式(2)で定義される$\alpha(i,t), \beta(i,t)$ を計算する。このαは、状態遷移するときの確率の変化を表している。(図5参照) また、βはαの逆で状態遷移を後ろからトレースしていく形になり (図6参照)、αとは双対をなしている。

よって、これらの計算からこのHMMが系列「a b b」を出力する確率は
$\alpha(3,3) = \beta(1,0) = 0.1232$ となる。

$$\alpha(i,t) = \sum_j \alpha(j,t-1) a_{ij} b_{ij}(y_t) \tag{1}$$

$$\beta(i,t) = \sum_j \beta(j,t+1) a_{ij} b_{ij}(y_{t+1}) \tag{2}$$

ただし、$\alpha(1,0) = \beta(3,3) = 1$

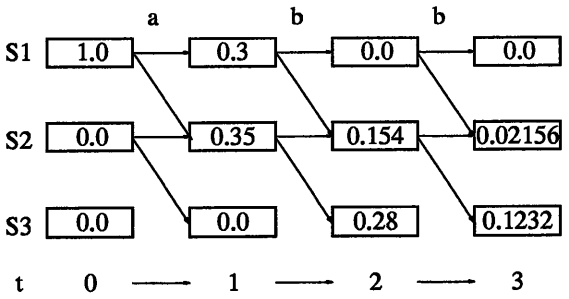

図 5: $\alpha(i,t)$ の計算

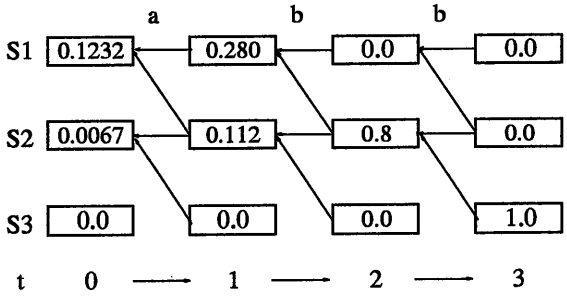

図 6: $\beta(i,t)$ の計算

このα,βを使って、式(3), 式(4)から、パラメータの更新値$\hat{a}_{ij},\hat{b}_{ij}(k)$を計算する。

$$\hat{a}_{ij} = \frac{\sum_t \alpha(i,t-1)a_{ij}b_{ij}(y_t)\beta(j,t)}{\sum_t \alpha(i,t)\beta(i,t)} \tag{3}$$

$$\hat{b}_{ij}(k) = \frac{\sum_{t:y_t=k} \alpha(i,t-1)a_{ij}b_{ij}(y_t)\beta(j,t)}{\sum_t \alpha(i,t-1)a_{ij}b_{ij}(y_t)\beta(j,t)} \tag{4}$$

この一連の計算によって推定されたパラメータ$(\hat{a}_{ij},\hat{b}_{ij})$は推定する前のパラメータ$(a_{ij},b_{ij})$よりも「a b b」の出力確率は高くなっているが、最大であるとは限らない。よって、$\hat{a}_{ij},\hat{b}_{ij}$を新しい$a_{ij},b_{ij}$として一連の計算を繰り返すことによって、パラメータを推定していくのである。

この方法は、必ずしも最大に収束するとは限らないが、極大に収束することは証明されている*。

なお、計算の初期パラメータは通常$a_{ij}=\frac{1}{2}$、また出力確率はシンボル数をnとして、$b_{ij}(k)=\frac{1}{n}$ *for all* kまたは、学習サンプル中に含まれるシンボルkの個数をn_k、シンボルの延べ総数をNとして、$b_{ij}(k)=\frac{n_k}{N}$とする。

実際のHMMの学習では、多数の学習サンプルを用いる。その場合には学習サンプルを$\boldsymbol{y}^1,\boldsymbol{y}^2,\cdots,\boldsymbol{y}^n$とすると、HMMが系列$\boldsymbol{y}^l$を出力する確率を$P(\boldsymbol{y}^l)$として、

$$P(\boldsymbol{y}^1)\cdot P(\boldsymbol{y}^2)\cdots P(\boldsymbol{y}^n)$$

を尤度関数として最尤推定していく。

具体的には、まず各サンプルごとに$\alpha(i,t),\beta(i,t)$を計算しておき、l番目のサンプルに対する$\alpha(i,t),\beta(i,t)$を$\alpha^l(i,t),\beta^l(i,t)$として、式(3), 式(4)のかわりに、以下の式を用いればよい。

$$\hat{a}_{ij} = \frac{\sum_l \frac{1}{P(\boldsymbol{y}^l)}\sum_t \alpha^l(i,t-1)a_{ij}b_{ij}(y_t^l)\beta^l(j,t)}{\sum_l \frac{1}{P(\boldsymbol{y}^l)}\sum_t \alpha^l(i,t)\beta^l(i,t)} \tag{3'}$$

$$\hat{b}_{ij}(k) = \frac{\sum_l \frac{1}{P(\boldsymbol{y}^l)}\sum_{t:y_t^l=k} \alpha^l(i,t-1)a_{ij}b_{ij}(y_t^l)\beta^l(j,t)}{\sum_l \frac{1}{P(\boldsymbol{y}^l)}\sum_t \alpha^l(i,t-1)a_{ij}b_{ij}(y_t^l)\beta^l(j,t)} \tag{4'}$$

これを、尤度関数が極大値に収束するまで繰り返して計算させればよい。

以上が *Forward-Backward* アルゴリズムによるHMMのパラメータ推定の基本的な部分である。

*L.E.Baum: "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov process",Inequalities, 3, pp.1–8 (1972)

第 3 章　学習サンプルが少ない時の問題点

3.1　問題点

HMMのパラメータ推定には通常 *Forward-Backward* アルゴリズムが使われる。第 2 章で示したHMMの出力確率の推定式を、もう一度示す。

$$b_{ij}(k) = \frac{\sum_{t:y_t=k} \alpha(i,t-1)a_{ij}b_{ij}(y_t)\beta(j,t)}{\sum_{t} \alpha(i,t-1)a_{ij}b_{ij}(y_t)\beta(j,t)} \qquad \text{式 (4)}$$

ここで、式 (4)の分子を見ると、学習サンプル中のシンボル y_t と kが一致する時だけ足し算を行なっている。つまり、学習サンプル中にでてこないシンボルの出力確率は 0 になってしまう。

充分な学習サンプルがあれば、その中にでてこないシンボルというのは他の音素と区別する重要な手掛りになるので、むしろ望ましいのだが、学習サンプルが少ない時は、たまたま学習サンプルに含まれなかったシンボルの出力確率も 0 になってしまうので、モデルの精度が極端に悪くなってしまう。

その結果、学習サンプルに含まれないシンボルを含む系列を認識しようとすると、図 7 のように認識不能になり、認識率はいちじるしく低下する。

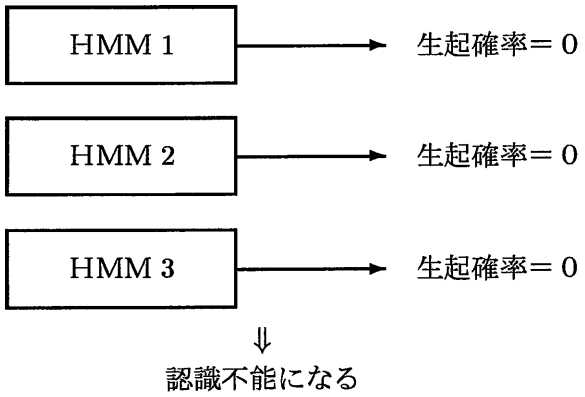

図 7: 新しいシンボルを含む系列の認識

3.2　平滑化前での認識実験

実際にどれくらい認識率が低下するかを見るために、以下のような実験をしてみた。図 8は、/N/,/b/,/d/,/g/,/m/,/n/の 6 子音について、各音素とも学習サンプル 19 個で学習し、平滑化を行なわずに認識実験をしてみた結果である。なお、こまかい設定などは、後述する実験と同じである。

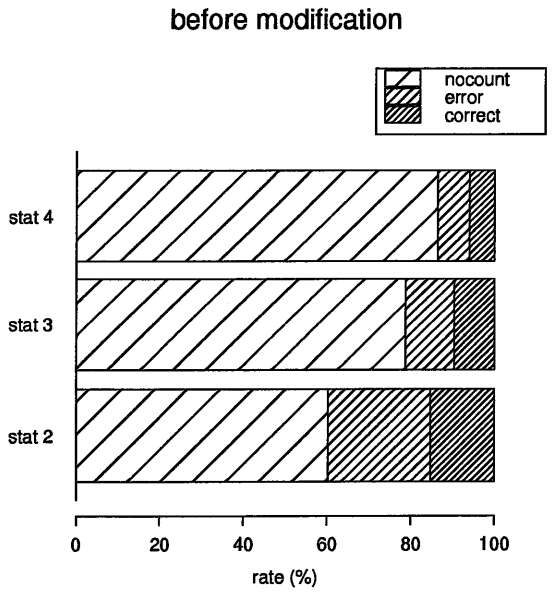

図 8: 平滑化を行なう前の認識結果

これを見ると、ほとんどが認識不能になっていることがわかるであろう。HMMの状態数は 2 から 4 まで実験しているが、特にパラメータの数が多い状態数 4 のHMMでは、85% 以上が認識不能になってしまっている。

そこで、学習サンプルが少ない時でも認識率があまり下がらないよう、特に認識不能ということをなくすために、通常の *Forward-Backward* アルゴリズムで学習させたあと、出力確率のムラをなくすために平滑化をしてやる必要があるのである。

第4章　出力確率の平滑化

4.1　出力確率の平滑化手法

ここでは、出力確率の平滑化手法として、以下の3つを示す。

4.1.1　手法 1：出力確率をベクトル間の距離で平滑化

手法のひとつめは、各ベクトル間の距離によって平滑化を試みる。

これは、学習サンプルに含まれたベクトルと距離が近いベクトルは、たとえ学習サンプルに含まれていなくても出力確率は高いほうが音声にあうだろう、という予測にもとづくものである。

具体的には、学習によって決定されたb_{ij}に対し、以下の式を使ってまわりのベクトルの出力確率に距離に比例するような重みをかけて足し算をし、それを正規化して、平滑化した出力確率$\hat{b}_{ij}$を得る。

$$\hat{b}_{ij}(k) = \frac{b_{ij}(k) + \sum_{l:l\neq k} b_{ij}(l) \times d_{kl}^{-p}}{\sum_{m} \left\{ b_{ij}(m) + \sum_{l:l\neq m} b_{ij}(l) \times d_{ml}^{-p} \right\}} \tag{5}$$

d_{kl} ：ベクトルk,l間の距離

pは定数

こうすることで、たまたま学習サンプル中になかったベクトルでも、それに近いベクトルが学習サンプル中にあればそれによって出力確率が0でなくなり、その結果認識不能がなくなる。

ここで、pはパラメータであり、これを振らせることで認識率が最大になるところを見付ける。

4.1.2　手法 2：出力確率をベクトル間の距離で平滑化 その2

この手法[†]は、基本的には手法 1と同じであるが、手法 1で比例的にかけていた重みを、指数関数的にかけている。そのことによって、より近いベクトルが強調されることになる。その結果、あまりに遠くのベクトルは出力確率が低いままなので、各HMM間の出力確率の特徴を失なうことなく、平滑化をすることができる。

$$\hat{b}_{ij}(k) = \frac{\sum_{l} b_{ij}(l) \times 10^{-d_{kl}\times w}}{\sum_{m}\sum_{l} b_{ij}(l) \times 10^{-d_{ml}\times w}} \tag{6}$$

d_{kl} ：ベクトルk、l間の距離

wは定数

この手法もパラメータwがあり、これを決定する必要がある。

[†]花沢、川端、鹿野：「HMM音韻認識におけるモデル学習の諸検討」 信学技報ＳＰ８８－２２

4.1.3　手法 3：出力確率の最低値を設定

この手法は、単純に出力確率がある値以下ならば、その値に引き上げる、というものである。これは、計算量の面からみて、上の 2 手法より簡単な方法である。

$$\tilde{b}_{ij}(k) = \max(b_{ij}(k), 10^{-m}) \tag{7}$$

m は定数

ただし、このあとで

$if\ \ \tilde{b}_{ij}(k) > 10^{-m}$

$$\hat{b}_{ij}(k) = \frac{(1 - n \times 10^{-m}) \times \tilde{b}_{ij}(k)}{\sum_{k:\hat{b}_{ij}(k)>10^{-m}} \tilde{b}_{ij}(k)} \tag{8}$$

$else$

$$\hat{b}_{ij}(k) = \tilde{b}_{ij}(k)$$

n は出力確率が 10^{-m} である個数

によって正規化する

ここで、出力確率の最低値 10^{-m} の最大値はシンボルの数を n 個として $\frac{1}{n}$、つまり、出力確率がすべてのシンボルについて等確率のときである。

4.2　認識実験

以上の手法の有効性をみるために、音素の認識実験を行なう。

4.2.1　使用データ

使用データは以下のとおり。

１９人の男女による単語発生音声を 24kHz 12bit で標本化し、高域強調後メル・ケプストラム係数の１～10 次を特徴量として１０次元のベクトルにし、それを１２８個のベクトルに量子化

単語数	２１２単語
使用音素	$/N/, /m/, /n/, /b/, /d/, /g/$ の６子音
音素数	学習…各人１個づつ計１９個を使用 ただし、充分に学習させるHMMには$/N/$: 511 $/b/$: 253 $/d/$: 215 $/g/$: 207 $/m/$: 431 $/n/$: 283 を使用 評価…各音素２０４個を使用
距離	ユークリッド距離

4.2.2　実験方法

通常の *Forward-Backward* アルゴリズムによって、各音素に対するHMMを作り、そのパラメータを各手法によって修正する。
HMMの形状は自己ループと隣りの状態への遷移枝のみ持つ、図9のような *Left-to-Right* モデル、状態数は２から４までとした。
また、認識率とは各音素の認識率の平均である。

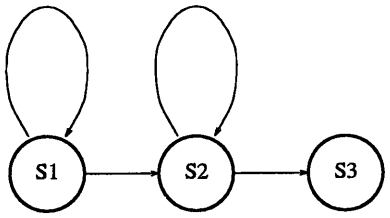

図 9: *Left-to-Right* モデル

4.2.3　パラメータの決定

各手法 1 つづつパラメータがあるため、まずそれを実験的に決定した。図 10.1~3は、それぞれの手法について、パラメータを振らせたときの認識率を示している。

これらのグラフから、もっとも認識率のよいパラメータを決定すると、

手法 1　　状態数 3　　$p = 2.12e-2$
手法 2　　状態数 3　　$w = 1.63e-4$
手法 3　　状態数 2　　$m = 4.1$

となる。そのときの認識率は

手法 1　　43.219%
手法 2　　43.137%
手法 3　　38.154%

であった。

なお、手法 3は、状態数 3・4では出力確率 $\frac{1}{128}$、つまりすべてのシンボルが等確率で出力されるのが最も認識率がよかった。また、認識率の推移はパラメータに対し単純な山型ではなく、いくつかの極大値をもっていた。

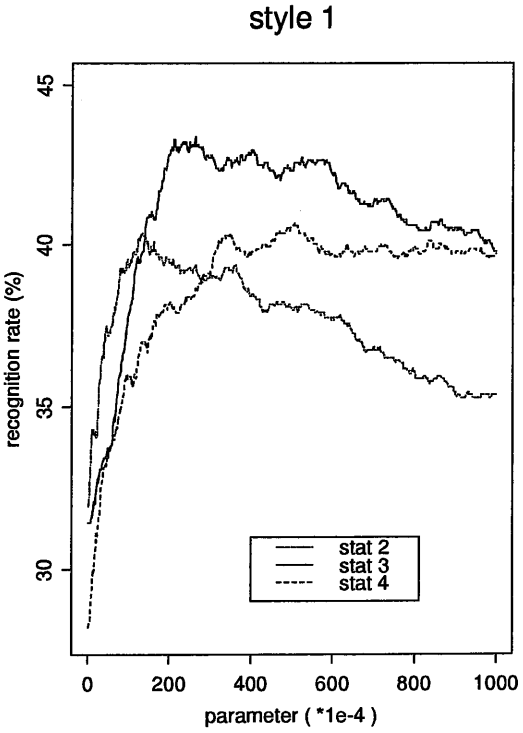

図 10.1：手法 1

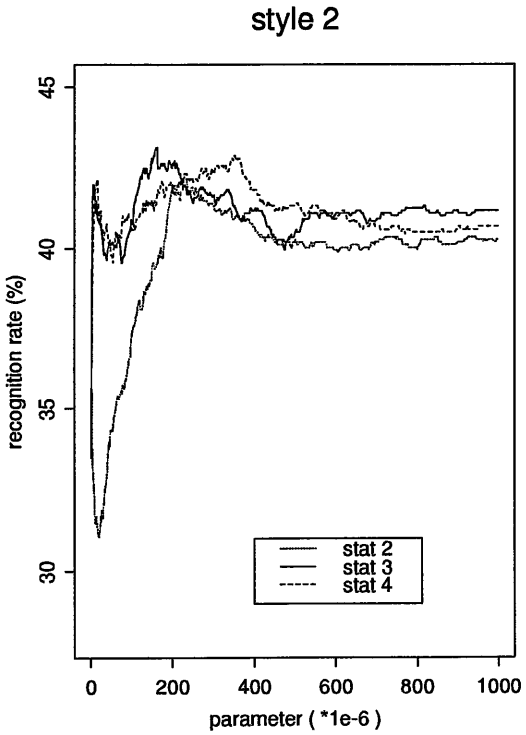

図 10.2：手法 2

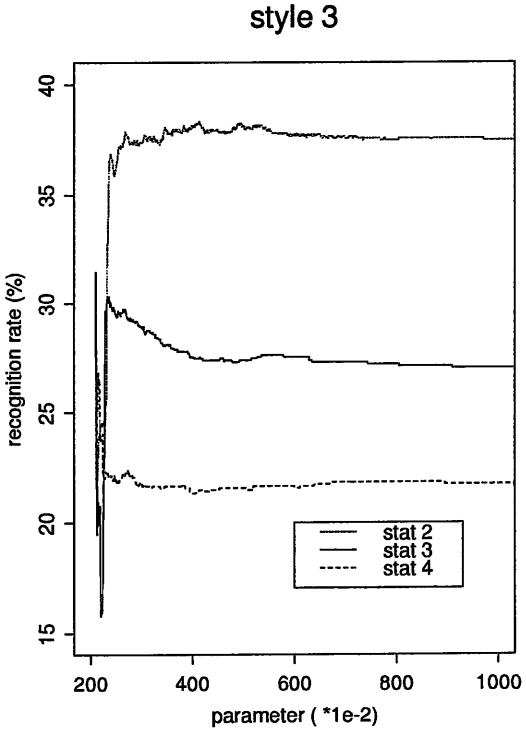

図 10.3：手法 3

4.2.4　重みのかかり方

図 11.1~2は、手法 1、2での最も認識率のよかったパラメータでのベクトル間の距離に対する重みの量を示している。実線は状態数 3、細かい破線は状態数 2、点線は状態数 4 のときの重みである。

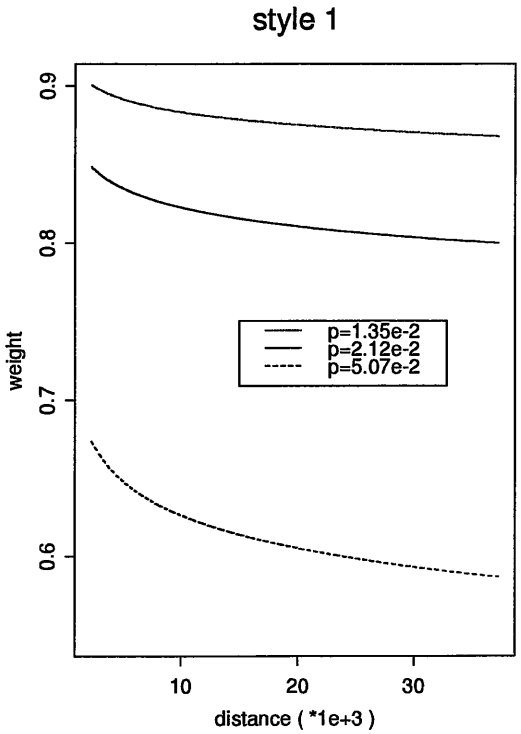

図 11.1：手法 1での重み

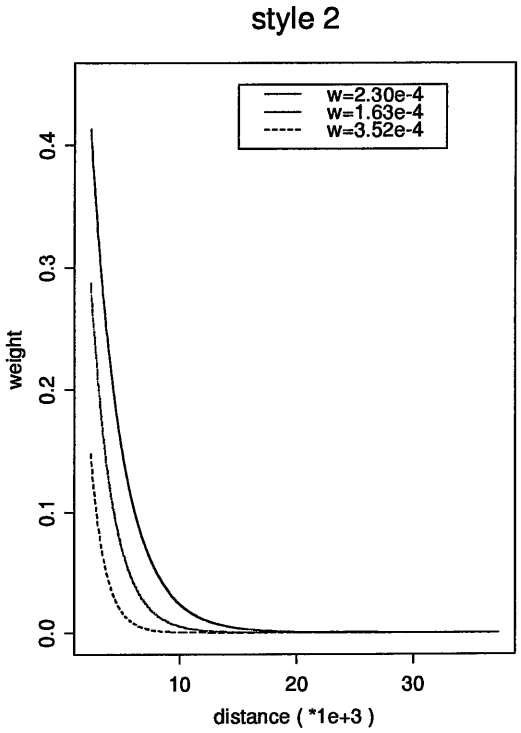

図 11.2：手法 2での重み

これを見ると手法 1は、手法 2に比べて大きな重みが遠くのベクトルまでかかっていることがわかる。つまり、出力確率がかなり一様に近くなっていると思われる。

これだけちがう重みのかかり方をしているが、状態数 3ではどちらの手法もほぼ同じ認識率である。

4.2.5　生起確率の分散

HMMがとれくらいの余裕をもって認識しているかを調べたのが図12.1~4である。横軸は認識すべき系列を正解のHMMが出力する確率、縦軸は誤りのHMMが出力する確率のうちでの最大値である。つまり、図12.2のように右上から左下へと直線を引いたときに、右下の領域にあれば認識成功、左上の領域にあれば誤認識となる。また、境界の直線から離れるほど余裕をもって認識していることになる

プロットは両対数プロット、また、状態数は3のときを示した。また、*all*とは、充分に学習したHMMのことである。

なお、生起確率の最低値は$1e-100$に設定した。

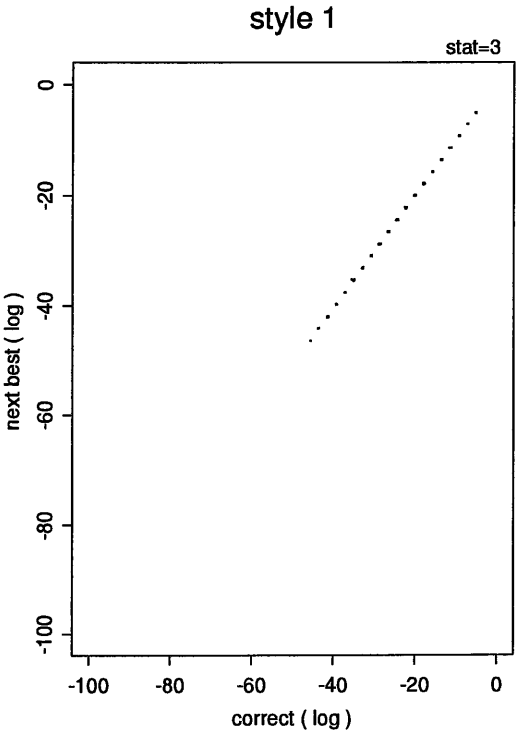

図 12.1：手法 1

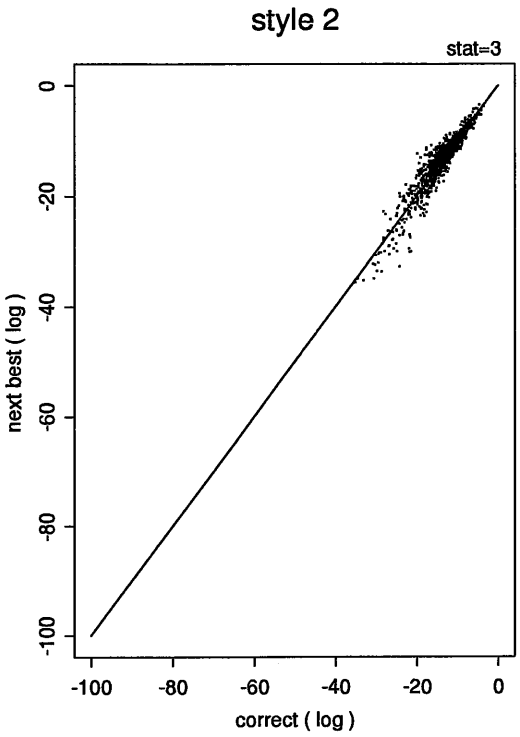

図 12.2：手法 2

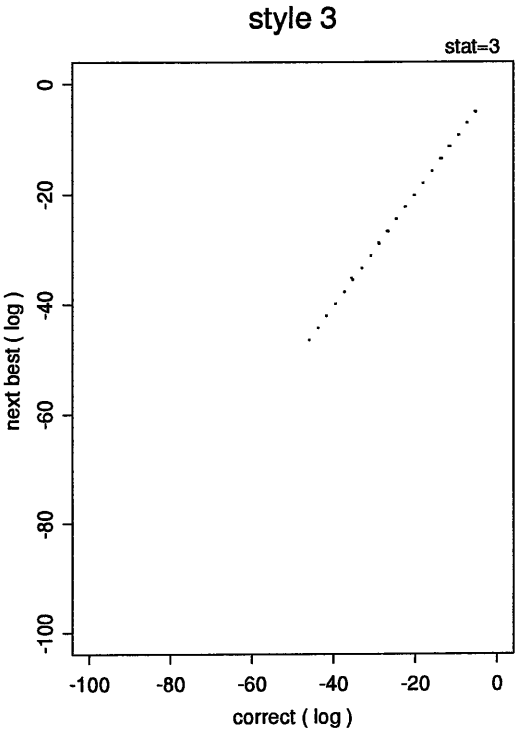

図 12.3：手法 3

これを見ると、手法 1はほとんど手法 3と同じ結果になっているのがわかるだろう。手法 3は出力確率がすべて等確率なので手法 1でもそれに極めて近い状態になっていると思われる。つまり、平滑化のしすぎであろう。その結果、手法 1、3はほとんど境界の直線上にあり、正解と誤りの間にほとんど差がないことがわかる。これは各HMM間での生起確率に差がないので認識に余裕がないことを示している。さらに離散化しているように見えるのは、出力確率がほぼ一定のため、入力系列が 1つ長くなると、生起確率がどれも同じだけ下がるためである。

それに対し、手法 2は充分に学習したものに似ていると思われる。だた、やはり充分に学習したものと比べると、境界線の近くに集まってしまっているのがわかる。また、生起確率の最低値が $1e-40$ 程度と高い値になってしまっているのは、平滑化のしすぎ、ということであろう。

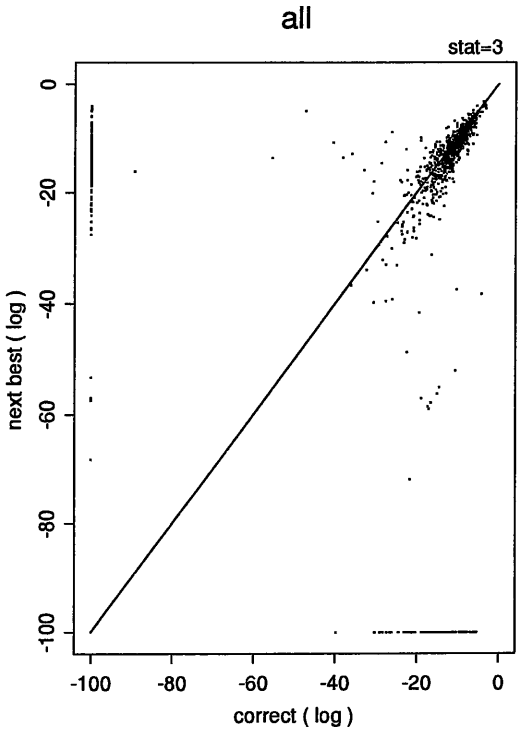

図 12.4：充分に学習したHMM

4.3　結論

以上の実験をまとめると、図13のようになる。このグラフは各手法での最高の認識率を示している。また、all とは、充分に学習したHMMの認識率である。

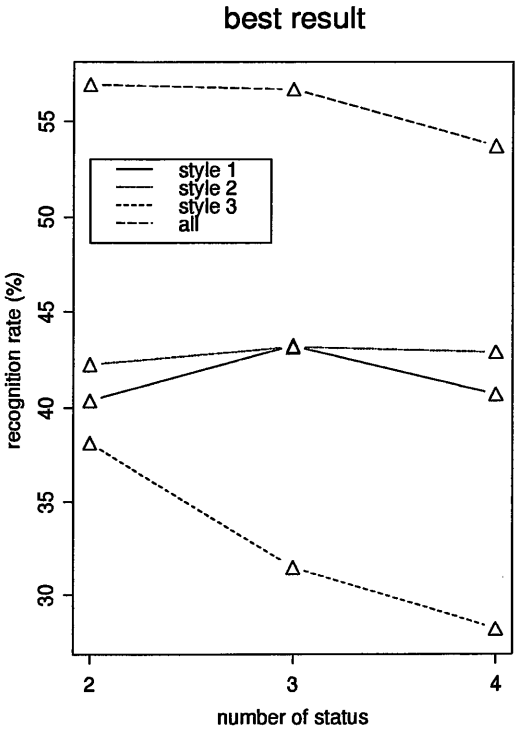

図 13: 実験結果

以上のことから、次のようなことがいえる。

- 状態数は３がよい。これは３つぐらいあれば音素を表現できてしまい、それ以上状態数を増やしても無意味だからなのであろう。
- ここでの手法は平滑化のしすぎになってしまっている。それによって各音素の特徴が失なわれ、認識に余裕がなくなり、誤認識が増えている。
- 手法 １は手法 ２に比べてより平滑化されている。その結果、出力確率が一定に近くなってしまい、認識に余裕がなくなってきた。
- 手法 ３は出力確率一定が一番認識率がよくなってしまった。これでは、音素の特徴は系列の長さだけとなってしまい、認識率が悪いのは当然といえよう。
- 平滑化は、結局手法 ２が最も有効である。これは、遠くのベクトルは重みが軽いために、出力確率が一様にならずに充分に学習したHMMに似た形になったためであろう。

これらの結果をふまえて、次の章では、手法 １、２について更に改良を試みる。

第5章　手法 1、2の改良

5.1　改良点

第4章での結論は平滑化をしすぎているので、遠くのベクトルは重みを軽くしたほうがよい、というものであった。そこで、平滑化の計算において、近いほうから n 個のベクトルまでで計算を打ち切ったほうがよいのではないか、と思われる。

今回は、第4章で結果のよかった状態数3のみについて、この改良を加えてみる。

5.2　認識実験

5.2.1　打ち切りの個数の決定

各手法について、10,15,20個までで打ち切ったときの認識率対パラメータの結果を以下に示す。

グラフ中の横線は、打ち切らなかったときの認識率である。

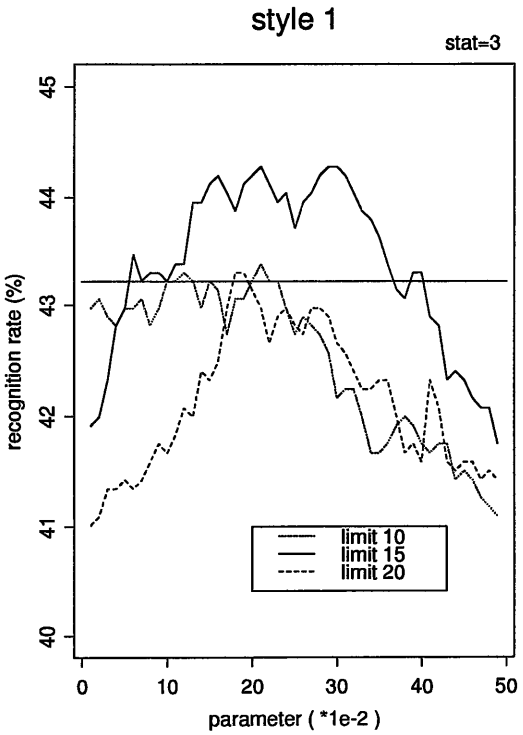

図 14.1：手法 1

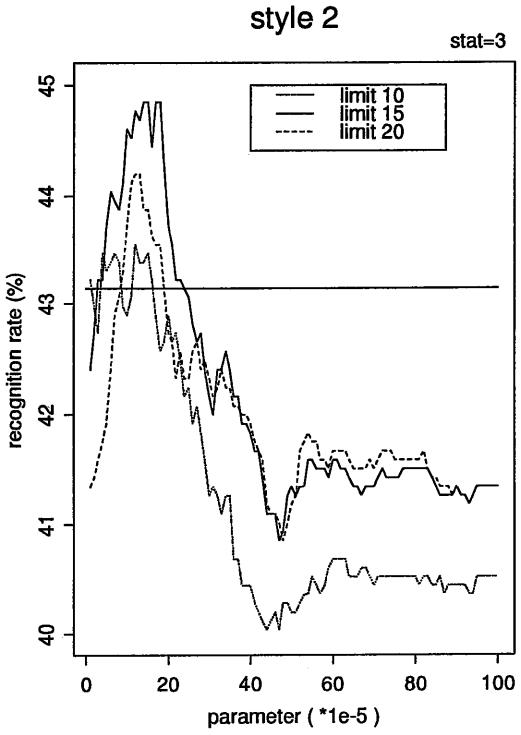

図 14.2：手法 2

これを見ると、どちらの手法でも、打ち切らなかった場合に比べて認識率が上がっており、改良の効果が見られたといえる。

また、打ち切る個数は15個がよいと思われる。このときのパラメータの値は、

手法 1　　$p = 2.1e-1$

手法 2　　$w = 1.4e-4$

となり、手法 1では、打ち切る前とは一桁違っているのがわかる。

また、そのときの認識率は
　手法 1　　44.281%
　手法 2　　44.853%
となっていた。

5.2.2　生起確率の分散

打ち切った場合の分散図は、以下のようになる。

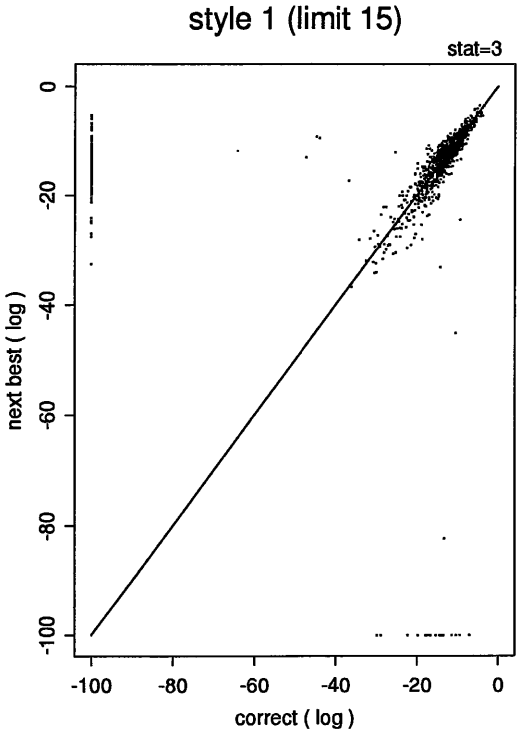

図 15.1：手法 1

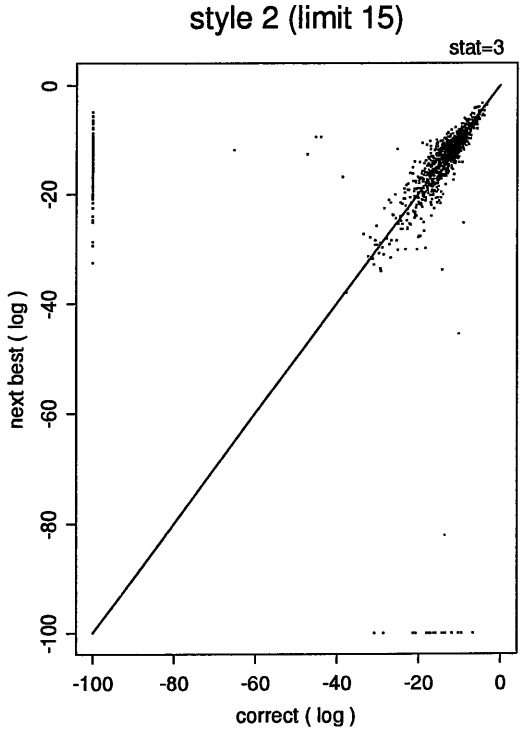

図 15.2：手法 2

これより、改良前と比べて多少ちらばりが増え、充分に学習したものに近くなっていることがわかる。

5.3　結論

- 計算を途中で打ち切るのは、有効である。これは、遠くのベクトルを無視することで、出力確率の特徴を失なわないようにできるからである。これによって分散図にも拡がりができ、認識に少し余裕ができたことがわかる。
- 打ち切りは15個ぐらいが適当である。これは、ベクトルによって差があるがだいたい距離にして8500ぐらいである。

第6章　まとめ

本研究はHMMを用いた音声認識の際に、学習サンプルが少ないときにおこる認識不能を回避し、充分に学習したものと認識率があまり変わらないようにするために出力確率を平滑化する手法を提案し、その有効性を確認した。

これらの実験より、以下のようなことがいえた。

- 認識率は平滑化前と比べて向上した。平滑化前の認識率が10%程度だったのに対し、平滑化を行なうと40%以上にまで上がった。また、認識不能は、平滑化をすると、ほとんど0%になっていた。

- 平滑化の重みは、ベクトル間の距離を用いるのが有効である。これは、近いベクトルは観測されやすい、と考えれば当然といえよう。重みは、距離を指数関数的にかけるのが比例的にかけるのよりよかった。比例的にかけるのでは出力確率がほぼ一定になってしまうほど平滑化されてしまっていた。また、単純に最低確率を底上げする手法では、出力確率一定が一番認識率がよかった。

- 平滑化の計算は、途中で打ち切るほうがよい。すべてのベクトルについて計算をしてしまうと、遠くのベクトルまで出力確率が上がってしまい、HMMの出力確率に関する特徴が失なわれてしまうからである。つまり、遠くのベクトルの出力確率は低いままのほうがよい。

結局、今回実験したものの中では手法 2で計算を15個で打ち切る方法が認識率44.8529%と最もよい成績であった。

第７章　今後の課題

本研究で残された、今後の課題について指摘しておく。

7.1　距離による平滑化の改良

- 打ち切りの閾値を距離にしてみる。今回は打ち切りを近いほうからの個数で行なっていたが、閾値を距離にするほうが妥当であろう。実際、打ち切りのベクトル(近い方から15個)の距離を見てみると、約6000～18000と開きがある。そこで、閾値を距離にして、認識実験をしてみる必要があるだろう。

- 重みのかけ方を改良する。今回は比例的と指数関数的の二つについて実験してみたが、その他にはないのか。また、今回は音声系列中にでてくるシンボルの順番などは考えずに平滑化をしたが、それを考慮した重みのかけ方があれば、そちらの方がよいのではないか。(状態ごとの平滑化など)

- 違う距離を使ってみる。今回使用したのはユークリッド距離だが、もっと音声の特徴をよく表したような距離尺度はないだろうか。

7.2　出力確率平滑化のその他の方法の検討

出力確率を平滑化する方法は、今回実験したもの以外にも数々提案されている。それらについても検討してみる必要があるだろう。

- 学習サンプルを擬似的に増加させてみる[‡]。学習データ中のあるベクトルに近いベクトルは観測されたこととして、そのベクトルと元のベクトルをいれかえた系列も学習データに含めてしまおうというもので、これによって学習データの量を擬似的に増加させることができる。

- ある状態の出力確率を他の状態の出力確率と同じにする[§]。こうすることによって、推定すべきパラメータを減らすことができるため、推定精度がよくなる。これによる効果はどうか。

- *Fuzzy*ベクトル量子化を使ってみる[¶]。*Fuzzy*ベクトル量子化とは、音声信号をベクトル量子化するときに、コードブック中のベクトルの一次結合で表し、その結合係数を使って平滑化するという方法である。だた、この方法は学習サンプルが充分にあるときは、かえって結果が悪くなる、という報告もある。

[‡]西村、年岡:「マルチラベリング手法を用いたHMMによる音声認識」音響学会講演論文集、3-5-11(1986.10)

[§]F.Jelinek and R.Mercer: "Interpolated estimation of Markov source parameters from sparse data" in Pattern Recognition in Practece,ed.E.S Gelsema and L.N.Kanal,North Holland (1980)

[¶]H.Tseng,M.J.Sabin,E.A.Lee: "Fuzzy Vector Quantization applied to Hidden Markov Modeling" Proc.of ICASSP '87,pp641-644,1987

7.3　その他の課題

その他に今回の実験をしていて疑問に感じたことを課題としてあげておく。

- 認識実験で、音素によって認識率にかなりのバラつきがみられた。例えば、/N/は70%程度なのに対し、/b/は15%程度であった。この原因を解明し、それを改良できれば、かなりの認識率向上になると思われる。
- 学習データのゆらぎに強いアルゴリズムを発見する。学習データが少ないときを対象にしているので、ある程度はしょうがないともいえるが、実験の結果が学習データセットによってかなりかわってしまう。これでは実用にはならないので、データセットのゆらぎを吸収するようなアルゴリズムを発見できないか。
- これらの平滑化手法を、充分に学習したHMMに適応したらどうなるか。実用では学習が充分かどうかは判断が難しくなるため、もし充分に学習したものに対しても効果がある(少なくとも認識率が落ちない)ならば、すべてのHMMについて学習後にこれらの手法を適応すればよいことになる。

・謝辞

本研究を進めるにあたり、全般的な御指導とともにこの研究の機会を与えて下さった東北大学工学部 阿曽弘具教授に心から感謝いたします。音声認識の専門分野においては、さまざまな御指導、助言をしてくださった東北大学工学部阿曽研究室の中井満氏、粟津辰功氏に深く感謝いたします。

日々の研究においては、計算機環境を整えていただいた同研究室の大町真一郎氏、沼田一成氏、後藤英昭氏、福田大氏に感謝いたします。また、同研究室の成富敬氏はじめそのほかの皆様にも数々の御討論、御意見をいただいたことに深くお礼申し上げます。

最後に、本研究を行なう上での貴重な音声資料を提供してくださった東北大学応用情報学研究センターおよび松下通信工業株式会社、松下技研株式会社の方々に感謝いたします。

・参考文献

1. 中川聖一：「確率モデルによる音声認識」
電気情報通信学会 (1988)

2. 鹿野清宏：「統計的手法による音声認識」
電気情報通信学会誌、12/'90 (1990)

3. 大河内正明：「マルコフモデルによる音声認識」
電気情報通信学会誌、4/'87 (1987)

4. 花沢、川端、鹿野：「HMM音韻認識におけるモデル学習の諸検討」
信学技報、SP88-22 (1988)

5. Lawrence R.Rabiner:"A Tutorial on Hiden Markov Models and Selected Applications in Speech Recognition"
Proc.IEEE,Vol.77,no.2,pp267–296,Feb.1989

6. 牧野、二矢田、真船、城戸：「東北大–松下単語音声データベース」
日本音響学会誌 48 巻 12 号 43.72.Ne (1992)