# 隠れマルコフモデルの生成に関する研究

平成5年2月10日

指導教官　中野 馨 助教授

東京大学大学院 工学系研究科 計数工学専攻

池 田　思 朗

# 目 次

# 第1章　序論

## 1.1　研究の背景

我々人間は様々な手段を用いてお互いの間で情報の伝達を行なう．それは文字であったり，身振りであったりする場合もあるが，特に音声は最も便利で頻繁に使われる手段である．

このように重要でかつ我々にとって便利な音声を，機械に認識させようという試みは自然である．音声によって機械を操作したり，文章を入力するといったマン・マシンインタフェイスとして，音声認識には大きな期待が寄せられている．実際に様々なグループがこのようなシステムを目指し，音声認識システムを製作している [1][2]．また，盛んに研究が行なわれている自動翻訳の一部としても音声認識は欠かすことができない．しかしながら，音声認識の現在の状況は，登録する単語を限定すれば不特定話者に対する単語認識も可能であるが (このような製品はすでに市場にも現れ始めている [3])，我々が通常用いる語彙全てを理解し，人間と同じ程度の正確さで認識する音声認識システムはいまだに存在していない．

ではなぜ音声認識は難しいのだろうか．それは，音声認識が信号処理から自然言語処理までといった様々な段階を含んでおり，各段階に解決されない問題が残っているからである．

音声波を信号解析し，音素認識をする．更に音素の並んだものである単語の認識，そして単語の並んだものである文章の認識．各段階には音声や自然言語の持つ曖昧さによる問題がある．これが音声認識を難しくしている．それぞれの音の持つ特徴は，個人毎にそして発生毎に，時間上そして周波数上で変動する．また，話し言葉は文法的な誤りを含む場合が多い．各層に様々な問題を含んでいる音声を扱うにはどうしたら良いであろう．

これに対する一つの提案として，70 年代から登場したのが確率モデルを用いた手法である．確率モデルを用いることの利点は主に次の 3 つの点である．1 つは音声の変動を確率を用いて表現できること．2 つめは信号処理から自然言語までの各段階を統一的に扱うことができること．そして 3 つめの点は統計的な手法を用いてモデルをある程度推定できることである．特に 2 つめに示した点は計算機での実現の上では重要である．簡単に説明する．

まず，音素の統計的性質を用いて各音素のモデルを作ることで，得られた時系列 $\boldsymbol{y}$ が音素 $\boldsymbol{Ph}_i$ である確率 $P(\boldsymbol{Ph}_i|\boldsymbol{y})$ を定義する．これを繋ぎ合わせることで $\boldsymbol{y}$ が単語 $\boldsymbol{W}_i$ である確率 $P(\boldsymbol{W}_i|\boldsymbol{y})$ が定義できる．単語を連結して文章のモデルを作る際には，単語間の出現確率を定義することによって自然言語の文法を表現できる．例えば，$P(\boldsymbol{W}_i|\boldsymbol{W}_j)$ のように互いに接している場合の単語間の出現確率を定義する．これによって文章のモデルを構成できる．

発生毎に変動する音声の特徴は決定論的には決定するのは難しい．この点で音声信号は確率的現象であるともいえる．また，発声器官，そして人間の脳の詳細なモデルが分からない以上，音声が発声されるモデルは入出力の関係から推定するより他はない．これらを考え合わせると，音声認識において確率モデルを用いる試みは自然であるように感じられる．

80 年代に入り，確率モデルを用いた音声認識は，必ずしも時系列を必要としない母音の認識や有声音/無声音の判別に用いられた．最近になって時系列パターンに対してもこの種の手法が用いられるようになった．時系列パターンを表現する代表的なモデルが，本論文で取り扱う隠れマルコフモデルである．このモデルは，音素や単語を扱う確率モデルとして現在頻繁に用いられている．

## 1.2 本研究の目的

確率モデルを用いる場合には，問題点もある．そのなかでも大きな 2 つは，確率モデルをどのように決めるか，という点と，統計的処理を行なうために，多量の訓練用データが必要だという点である．

データ量に関しては，集める他にない．では，確率モデルの決定に関してはどうで

あろうか.

確率モデルを決めるということは，その構造とパラメータの値をどのように定めるかということである．パラメータの値は，最尤推定法を用いて推定できる．一方，確率モデルの構造とは，パラメータ数やパラメータ間の関係から定まるものであり，どの確率モデルを用いれば良いかは扱う問題によって異なる．したがって構造を決定する際の一般的な手法というものはない.

この問題は，隠れマルコフモデルにおいても同様である．隠れマルコフモデルは前節で説明したように音素，単語のモデルとして用いられるのだが，実際に音声認識システムを作る場合，どのような構造のものを用いたら良いのかは当然問題になる．隠れマルコフモデルの場合，その状態数や状態遷移といったものを構造と呼んで良い．現在は設計者の試行錯誤によって幾つかうまく行く構造というものが考えられており，それを用いた上で，最尤推定法の一種である EM (Expectation - Maximization) アルゴリズム (2.5.1 節) によってパラメータの値を推定する.

このような確率モデルの構造を機械自身に生成させることはできないであろうか.

これはモデル探索と呼ばれる問題と等価である．モデル探索では，モデルを構造も含めて変化させ，確率モデルを推定したいのであるが，この場合，構造毎にパラメータの推定を行なう必要がある．すなわち，パラメータ推定を含めて，モデルの探索空間が広過ぎるのである．一方，音声認識システムで用いる隠れマルコフモデルは，初期状態と最終状態が定まっており，この点で特殊である．したがってある程度探索空間を狭くすることができる．もちろん狭いからといってモデルの構造を段々に変化させていく場合，パラメータ推定を無視することはできない．このためにある程度試行錯誤的になるのはやむを得ないが，効率良く，より良い構造を選ぶことができれば，システム全体としての機能は結果として良くなるはずであろう.

神経回路網や重回帰問題でのモデル選択の問題としては，近年，AIC 等のモデル選択基準が広く用いられるようになっている．これらの基準は複数のモデル間での良さを比較するものであるが，モデルをどのように変化させていけば良いかは一般には与えられない.

音声認識におけるこのモデル選択の問題を，機械にモデルを生成させるという方法を用いて解決し，音声認識システムとしての機能も向上させようというのが，本研究

の目的である．

現在は，音声認識システムの (すなわち隠れマルコフモデルの) モデル探索を目標にしているが，モデル探索という問題は，他の確率モデルでも必要な問題である．この第一段階として行なった試みについて本論文で述べていきたい．

## 1.3 本論文の構成

本論文の構成を述べる．

まず，2 章で基本となる隠れマルコフモデルにおいて，どのように学習し，認識を行なうかを，学習の基本となる EM アルゴリズムも含めて述べる．

3 章では本論文で提案する 2 つのアルゴリズムについて説明する．1 つは状態を増やすアルゴリズムであり，もう 1 つは状態数を固定したまま状態遷移確率を増やすアルゴリズムである．さらに計算機の上で作った確率的情報源によるデータに対して，これらのアルゴリズムを用いてモデルを構成し，認識を行ない，アルゴリズムの有用性を述べる．

4 章では実際に音声を使った実験について述べる．音声を用いた実験では，各段階毎に幾つかの既存の技術を用いている．それらも含めて説明し，実験の具体的な内容と結果を示す．5 章でその結果についての考察を行なう．

# 第2章　隠れマルコフモデル

## 2.1　マルコフ連鎖 [4][5]

隠れマルコフモデルはマルコフ連鎖をその基礎としている．まず，そのマルコフ連鎖について述べる．

ある系の状態を離散的な時点 $n = 0, 1, 2, \cdots$ で考える．その系のとり得る状態の数は有限，または加算無限個であるとし，それらを $s_1, s_2, \cdots$ で表す．その全体の集合 $S = \{s_1, s_2, \cdots\}$ を**状態空間**という．簡単のため状態空間を $S = \{1, 2, \cdots\}$ とあらわすことにする．

$x_t$ を時点 $t$ で，状態が $s_j$ のとき $x_t = j$ となる確率変数列 $\{x_t\}$ とする．このとき $x_0 = i_0$ , $x_1 = i_1$ , $\cdots$ $x_{t-n} = i_{t-n}$ , $x_{t-n+1} = i_{t-n+1}$ , $\cdots$ , $x_{t-1} = i_{t-1}$ , $(t > n)$ であったとき，$x_t = j$ となる条件付き確率が次式で与えられるとき，この確率過程を $n$ 重マルコフ過程という．

$$P(x_t = j|\boldsymbol{x}_1^{t-1} = \boldsymbol{i}_1^{t-1}) = P(x_t = j|\boldsymbol{x}_{t-n}^{t-1} = \boldsymbol{i}_{t-n}^{t-1}) \tag{2.1}$$

ここで，$\boldsymbol{x}_{t_1}^{t_2} = (x_{t_1}, x_{t_1+1}, \cdots, x_{t_2})$ , $\boldsymbol{i}_{t_1}^{t_2} = (i_{t_1}, i_{t_1+1}, \cdots, i_{t_2})$ $(t_1 \leq t_2)$ を表すものとする．特に $n = 1$ のとき，これをマルコフ連鎖 (Markov chain) と呼ぶ．

状態数が有限のマルコフ過程を考える．このとき，一般に，$n$ 重マルコフ過程は $\boldsymbol{x}_{t-n}^{t-1}$ の取り得る $n$ 次空間内の各点 (状態数を $N$ 個とすると，たかだか $N^n$ 個) を新しい確率変数 $y_m$ を用いて新しく定義することによってマルコフ連鎖になる．図 2.1 に例を示す．

さて，マルコフ連鎖では，式 2.1 は式 2.2 となる．

$$P(x_t = j|\boldsymbol{x}_1^{t-1} = \boldsymbol{i}_1^{t-1}) = P(x_t = j|x_{t-1} = i_{t-1}) \tag{2.2}$$

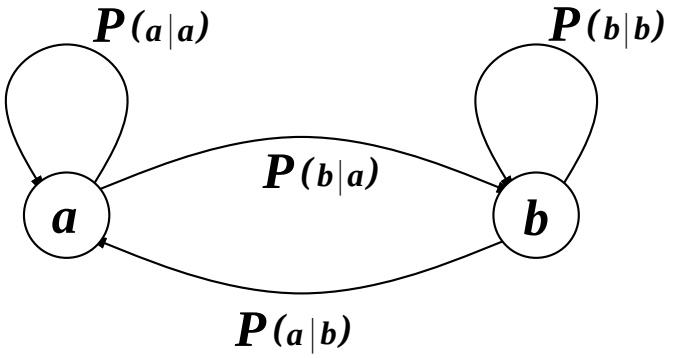


図 2.1: マルコフ連鎖の例

これが常に成立するものとして,

$$P_{ij}(t-1,t) = P(x_t = j | x_{t-1} = i) \tag{2.3}$$

とおくと,

$$P_{ij}(t-1,t) \geq 0, \qquad \sum_j P_{ij}(t-1,t) = 1 \tag{2.4}$$

である．$P_{ij}(t-1,t)$ は，時刻 $t-1$ に状態 $i$ であったものが，時刻 $t$ に状態 $j$ に遷移する確率を表しているのだが，この遷移確率が $t$ に無相関のとき，これを定常なマルコフ過程という．また，初期状態確率 $\pi_i$ を式 2.5 で定義する．

$$\pi_i = P(x_1 = i), \qquad \sum_j \pi_j = 1 \tag{2.5}$$

定常な遷移確率 $P_{ij}$ をもった単純マルコフ過程を考える．遷移確率 $P_{ij}$ を要素とする行列,

$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{pmatrix} \tag{2.6}$$

を用いると，現時点における状態が $s_i$ であるとき，$n$ 時点後に状態 $s_j$ に遷移する確率は $P_{ij}^{(n)}$ は，

$$P_{ij}^{(n)} = \sum_{x_{t_1}^{t_2}} \prod_{t=t_1}^{t_2-1} P_{x_{t_1} x_{t_1+1}} \qquad x_{t_1} = i \wedge x_{t_2} = j$$

となるが，これは $\boldsymbol{P}$ の $n$ 乗である $\boldsymbol{P}^n$ の第 $(i, j)$ 成分と等しい．

マルコフ連鎖のように各状態から出力されるシンボルが異なる有限情報源はユニフィラー (unifilar)[6] と呼ばれる．

ある確率過程があるとき，その確率過程が定常なマルコフ過程であることが分かっているとしよう．ただし，各状態遷移確率は実験によって推定しなければならない．このとき，その確率過程から得られる無限に長いサンプル系列を用いることで，その遷移確率を推定することができる．例えば，図 2.1 において，$P(b|a)$ を推定したいとする．その場合は，状態 $a$ を通過した回数 $N_a$，$a$ の次に状態 $b$ を通過した回数を $N_a(b)$ として，

$$P'(b|a) = \frac{N_a(b)}{N_a} \tag{2.7}$$

として推定することができ，サンプルが長いとき，推定された $P'(b|a)$ は $P(b|a)$ と等しくなる．

## 2.2 隠れマルコフモデルの基本

HMM ( Hidden Markov Model: 隠れマルコフモデル) では出力シンボルは各状態毎に定義される出力確率分布にしたがう．それぞれの出力確率分布の間には重なりがあるので，HMM はユニフィラーではない．すなわち，マルコフ連鎖と異なり，出力シンボル系列からは状態遷移先が一意には決まらない．

一般的に言う HMM は通常のマルコフ連鎖と同様に最終状態のないものを指すことが多く，そのような HMM 間での距離や判別，一意性について研究がなされている [7][8] [9]．一方，音声認識等で用いる HMM は初期状態，最終状態を設定する．特にこのようなモデルは left-to-right モデル ( L-R モデル) と呼ばれる．

HMM を音声認識に用いるということは，次のように考えられる．HMM の各状態に存在する間，出力シンボルは定常的な確率分布に従う．その存在する状態を確率的に切替えることによって音声を表そうとしているのである．音声では，発音が始まり，終るまでその時間方向に周波数的な特徴が変化していくが，逆方向に戻ることはない．したがって，このような L-R モデルを用いた方が良い．また，音声認識では，各状態で信号 $x_i$ を出力する「状態出力」型よりも状態 $i$ から状態 $j$ へ遷移する時に信号 $x_i$ を出力する「遷移出力」型を用いることが多いが，互いに等価変形することができることから，本論文では前者を用いて説明を行なう [10].

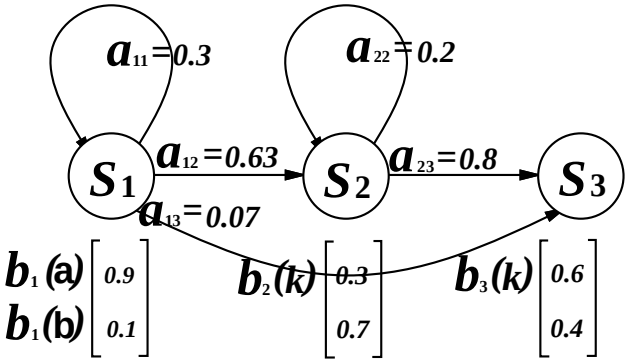


図 2.2: HMM の例

また，各状態に定義される出力確率分布は，様々な確率分布が考えられる．本研究においては，多項分布を考える．音声認識で良く用いられるのは，多項分布，あるいは多次元の正規分布 (各成分を独立としたものが多い) である．図 2.2 にあるのが，簡単な HMM である．図中では出力シンボルは a, b の二つである．

## 2.3 定式化

図 2.2 に示された HMM $\boldsymbol{M}$ を用いて簡単に HMM を説明していく．シンボル列，$\boldsymbol{y} =$ abba が観測されたとしよう．この時，そのシンボル列 abba が HMM $\boldsymbol{M}$ から発生した確率 $P(\boldsymbol{y}|\boldsymbol{M})$ は次の手順で計算される．

$\boldsymbol{M}$ から abba が得られたとしたとき，可能性のある状態遷移 $\boldsymbol{x}_i$ は初期状態を $S_1$

最終状態を $S_3$ として $\boldsymbol{x}_1 = S_1S_1S_1S_3$，$\boldsymbol{x}_2 = S_1S_1S_2S_3$，$\boldsymbol{x}_3 = S_1S_2S_2S_3$ の 3 通りである．従って，

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{M}) &= \sum_i P(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{M}) && (2.8) \\ &= \sum_i P(\boldsymbol{y}|\boldsymbol{x}_i) P(\boldsymbol{x}_i|\boldsymbol{M}) && (2.9) \end{aligned}$$

とかける．$\boldsymbol{x}_2$ を例に $P(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{M})$ を計算する．$P(\boldsymbol{x}_2|\boldsymbol{M})$ は，

$$\begin{aligned} P(\boldsymbol{x}_2|\boldsymbol{M}) &= a_{11} \times a_{12} \times a_{23} \\ &= 0.3 \times 0.63 \times 0.8 = 0.1512 \end{aligned} \quad (2.10)$$

また，$P(\boldsymbol{y}|\boldsymbol{x}_2)$ は，

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}_2) &= b_1(\mathsf{a}) \times b_1(\mathsf{b}) \times b_2(\mathsf{b}) \times b_3(\mathsf{a}) \\ &= 0.9 \times 0.1 \times 0.7 \times 0.6 = 0.0378 \end{aligned} \quad (2.11)$$

従って，

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}_2, \boldsymbol{M}) &= P(\boldsymbol{y}|\boldsymbol{x}_2) P(\boldsymbol{x}_2|\boldsymbol{M}) \\ &= 0.1512 \times 0.0378 = 0.00571536 \end{aligned} \quad (2.12)$$

となる．$P(\boldsymbol{y}|\boldsymbol{M})$ は，

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{M}) &= \sum_i P(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{M}) = P(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{M}) + P(\boldsymbol{y}|\boldsymbol{x}_2, \boldsymbol{M}) + P(\boldsymbol{y}|\boldsymbol{x}_3, \boldsymbol{M}) \\ &= 0.00003402 + 0.00571536 + 0.02667168 = 0.03242106 \end{aligned} \quad (2.13)$$

と計算される．このように HMM$\boldsymbol{M}$ は次の 6 つの組 $\boldsymbol{M} = (\boldsymbol{S}, \boldsymbol{Y}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}, \boldsymbol{F})$ で定義される．

$\boldsymbol{S}$ ：内部状態の集合，$\boldsymbol{S} = \{s_i\}$ HMM ではマルコフ モデルと異なり，実際に意味のあるものとは明確な対応づけができない場合が ある．

$\boldsymbol{Y}$ : 出力シンボルの集合．さまざまな定義の方法が考えられるが，ここでは離散で有限個のシンボルの集合と考えることにする．

$\boldsymbol{A}$ : 状態遷移確率の集合．$\boldsymbol{A} = \{a_{ij}\}$ ; $a_{ij}$ は状態 $s_i$ から状態 $s_j$ への遷移確率．

$$\sum_j a_{ij} = 1$$

$\boldsymbol{B}$ : 出力確率の集合． $\boldsymbol{B} = \{b_i(k)\}$ ; $b_i(k)$ は状態 $s_i$ に おいてシンボル $k$ を出 力する確率．

$$\sum_k b_i(k) = 1$$

$\boldsymbol{\pi}$ : 初期状態確率の集合． $\boldsymbol{\pi} = \{\pi_i\}$ ; $\pi_i$ は 初期状態が $s_i$ である確率．

$$\sum_i \pi_i = 1$$

また，$\pi_i \neq 0$ である状態の集合を $\boldsymbol{S}_I$ とする．

$\boldsymbol{F}$ : 最終状態の集合．

このようにして定義される HMM$\boldsymbol{M}$ に関して重要な基本問題を次の順で説明していく [11] [12] [6] .

1. モデルを用いての認識 : モデル $\boldsymbol{M}$ がシンボル列 $\boldsymbol{y}$ を出力する確率 $P(\boldsymbol{y}|\boldsymbol{M})$ の効率的な求め方 (2.4 節).

2. モデルの推定 : 訓練用のシンボル系列 $\boldsymbol{y}$ を与えて，$P(\boldsymbol{y}|\boldsymbol{M})$ が最大になるように $\boldsymbol{M}$ のパラメータを推定する (2.5 節).

3. モデルの設計 : 状態数や遷移先の種類等，実際の音声認識の場でどのように HMM を設計しているか (2.6 節).

## 2.4 モデルを用いての認識

音声認識等で HMM を用いる場合，カテゴリ (音素や単語) の数だけ HMM を用意し，認識したいデータに対し，それぞれからそのデータが生じる確率を計算し，認識を行なう．すなわち，カテゴリの数だけ $\{\boldsymbol{M}_1, \boldsymbol{M}_2, \cdots \boldsymbol{M}_n\}$ を用意する．各カテゴ

リに対して，$P(\boldsymbol{y}|\boldsymbol{M}_i)$ を計算し，その最大のものをもってデータの属するカテゴリとする．正確には文法等が入ってきてもう少し複雑になるのだが (2.6 節)，おおよそ，このような手順である．

式 2.8，2.9 に示したように，$P(\boldsymbol{y}|\boldsymbol{M}_i)$ は可能な状態遷移に対する確率を加え合わせたものである．特に大語彙に対するシステムの場合は，この $P(\boldsymbol{y}|\boldsymbol{M}_i)$ を効率良く計算することが重要になる．実際の音声認識の場においては，可能性のある全ての状態遷移に対して計算を行なわず，可能性の高い幾つか (あるいは一つ) の状態遷移のみに対して計算を行なう方法 ( Viterbi によって提案されたことから，Viterbi アルゴリズムと呼ぶ ) が使用される場合が多い．モデルが大規模になった場合，計算量が多くなり過ぎることからこのような方法を用いる必要がある．また，特に 1 つの状態遷移のみを考えることによって $P(\boldsymbol{y}|\boldsymbol{M}_i)$ は対数で展開でき，計算上の桁落ち等の問題を回避できる．

さて，$P(\boldsymbol{y}|\boldsymbol{M}_i)$ を効率良く求める方法として forward アルゴリズムと呼ばれるものを説明する．これは認識の過程だけではなく，パラメータ推定 (EM アルゴリズム) の際にも用いる．

**Forward Algorithm**

まず，1 次のマルコフモデルを用いていることから，

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{M}) &= \sum_i P(\boldsymbol{y}|\boldsymbol{x}_i)P(\boldsymbol{x}_i|\boldsymbol{M}) \\ &= \sum_i \pi_{x_i^1} b_{x_i^1}(y_1) \prod_{t=1}^{T-1} a_{x_i^t x_i^{t+1}} b_{x_i^{t+1}}(y_{t+1}) \end{aligned} \tag{2.14}$$

$x_i^t$ は状態を示し，$\boldsymbol{x}_i = x_i^1, x_i^2, \cdots, x_i^T$ であり，$x_i^1 \in \boldsymbol{S}_I$ 且つ $x_i^T \in \boldsymbol{F}$ である．この計算は漸化式を用いて次のように効率良く計算できる．

$$\alpha_i(t) = \begin{cases} 0 & t = 1 \wedge i \notin \boldsymbol{S}_I \\ \pi_i & t = 1 \wedge i \in \boldsymbol{S}_I \\ \sum_j \alpha_j(t-1) a_{ji} b_i(y_t) & t > 1 \end{cases} \tag{2.15}$$

このようにして求まった $\alpha_i(t)$ は，時刻 $t$ で，$\boldsymbol{y}_1^t = y_1, y_2, \cdots, y_t$ を観測してきて，現

在状態 $i$ にいる確率を表していることになる．これより明らかに，

$$P(\boldsymbol{y}|\boldsymbol{M}) = \sum_{i \in \boldsymbol{F}} \alpha_i(T) \tag{2.16}$$

となる．

## 2.5 学習

HMM を実際に用いる場合，各パラメータ (状態遷移確率及び出力確率分布) をいかに決定するかが問題となる．

観測値が与えられた時に，その確率密度関数が $f(\boldsymbol{y}|\boldsymbol{\theta})$ であるとしよう．このとき，パラメータ $\boldsymbol{\theta}$ を推定するには，最尤推定を用いる場合が多い．しかしながら，HMM のように，出力シンボル系列からは内部の状態系列は直接観測できない場合，直接最尤推定を行なうことは難しい．これは直接見えるデータの他に付加データがある時のパラメータ推定の問題と等価である．このとき良く用いられる方法が，EM アルゴリズム [13] [14] [15] と呼ばれる繰り返し演算である．HMM に対するパラメータの推定として良く用いられる Baum-Welch アルゴリズムは，この EM アルゴリズムを HMM に適応したものになっている．

### 2.5.1 EM アルゴリズム

最尤推定を行なう場合，確率密度関数やその対数をとったものを尤度関数 ( Likelihood Function ) $L(\boldsymbol{y}, \boldsymbol{\theta})$ ($\boldsymbol{y}$ はサンプルデータ，$\boldsymbol{\theta}$ は HMM の場合 $\{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{B}\}$ だと考えれば良い) として定義し，それを $\boldsymbol{\theta}$ の関数だとみなし，最大になるように $\boldsymbol{\theta}$ を選ぶ．言い換えれば尤度 $L(\boldsymbol{y}, \boldsymbol{\theta})$ は最尤推定法によって求まった点 $\boldsymbol{\theta}^*$ で最大になる．

しかし，HMM のように直接観測できないデータがある場合，最尤推定法を用いるのは難しいことから，EM アルゴリズムと呼ばれる繰り返し演算によって推定することが多い．EM アルゴリズムの 1 回の ステップによって $\boldsymbol{\theta}$ から $\hat{\boldsymbol{\theta}}$ が求まったとき，この $\hat{\boldsymbol{\theta}}$ に対して，

$$L(\boldsymbol{y}, \hat{\boldsymbol{\theta}}) \geq L(\boldsymbol{y}, \boldsymbol{\theta}) \tag{2.17}$$

が成り立てば，$\hat{\boldsymbol{\theta}}$ を $\boldsymbol{\theta}$ に置き換えて，この演算を繰り返すことによって，$\boldsymbol{\theta}$ は最大点あるいは極大点に収束する．このアルゴリズムは観測データの中に付随して得られる付加的なデータが含まれている場合のパラメータ推定の問題とと本質的に同じ問題である [15]．

$\boldsymbol{x}$ を付随データ (HMM では状態遷移)，$\boldsymbol{y}$ を観測データ，$\boldsymbol{\theta}^i$ を $i$ 回目に推定された点として，$i+1$ 回目にパラメータの推定を行なう場合を考える．$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ を次のように定義する．

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = \int_x \log\{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}^i) d\boldsymbol{x} \tag{2.18}$$

HMM の場合は，$\boldsymbol{x}$ が離散であるので，積分が和になり，

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i) = \sum_x \log\{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}^i) \tag{2.19}$$

のように定義される．

EM アルゴリズムの手順は次の通りである．

**EM アルゴリズム**

1. パラメータ $\boldsymbol{\theta}$ の初期値を設定．
2. $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ を求める．
3. $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ を最大にする $\boldsymbol{\theta}$ を求め，$\hat{\boldsymbol{\theta}}$ とする．
4. $\boldsymbol{\theta}^{i+1} = \hat{\boldsymbol{\theta}}$ とする．
5. 収束したら終了．そうでなければ 2 へ戻る．

$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ は $E_{\theta^i}[\log\{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\}]_{x|y}$ であることから，2. を $Estep$(Expectation Step) 3. を $Mstep$(Maximization Step) と呼ぶ．

ここで問題になるのは，3. で求まった $\hat{\boldsymbol{\theta}}$ に対して，式 2.17 が成り立つかどうかである．まず，$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ を式 2.20 のように書き換える．

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i) \quad = \quad \int_x \log\{p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta})\} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\theta}^i) d\boldsymbol{x}$$

$$
\begin{aligned}
&= \int_x \log\{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta})\}\frac{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}d\boldsymbol{x} \\
&= \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\int_x \log\{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta})\}p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)d\boldsymbol{x} \qquad (2.20)
\end{aligned}
$$

ここで，2.17 は log の凸性を用いて，次式から証明できる．

$$
\begin{aligned}
Q(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}^i) - Q(\boldsymbol{\theta}^i,\boldsymbol{\theta}^i) &= \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\left[\int_x \left\{\log\{p(\boldsymbol{x},\boldsymbol{y}|\hat{\boldsymbol{\theta}})\} - \log\{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)\}\right\}\right. \\
&\quad \left.\times p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)d\boldsymbol{x}\right] \\
&= \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\int_x \log\frac{p(\boldsymbol{x},\boldsymbol{y}|\hat{\boldsymbol{\theta}})}{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)}p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)d\boldsymbol{x} \qquad (2.21) \\
&\leq \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\int_x \left\{\frac{p(\boldsymbol{x},\boldsymbol{y}|\hat{\boldsymbol{\theta}})}{p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)} - 1\right\}p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)d\boldsymbol{x} \\
&= \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\int_x \{p(\boldsymbol{x},\boldsymbol{y}|\hat{\boldsymbol{\theta}}) - p(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{\theta}^i)\}d\boldsymbol{x} \\
&= \frac{1}{p(\boldsymbol{y}|\boldsymbol{\theta}^i)}\{p(\boldsymbol{y}|\hat{\boldsymbol{\theta}}) - p(\boldsymbol{y}|\boldsymbol{\theta}^i)\} \qquad (2.22)
\end{aligned}
$$

式 2.22 より，もし $Q(\hat{\boldsymbol{\theta}},\boldsymbol{\theta}^i)$ が $Q(\boldsymbol{\theta}^i,\boldsymbol{\theta}^i)$ より大きければ，$p(\boldsymbol{y}|\hat{\boldsymbol{\theta}})$ が $p(\boldsymbol{y}|\boldsymbol{\theta}^i)$ より大きくなることが証明できた．

### 2.5.2 Baum-Welch アルゴリズム

HMM のパラメータ推定に用いる Baum-Welch のアルゴリズム [14]，を説明する．これは前節で説明した EM アルゴリズムを HMM に適応したものになっている．準備として，2.4 節に示された Forward Algorithm と同様に Backward Algorithm を定義する．

**Backward Algorithm**

式 2.15 と同様に $x_i^t$ は状態を示し，$\boldsymbol{x}_i = x_i^1, x_i^2, \cdots, x_i^T$ である．ここに $\beta_i(t)$ を漸化式を用いて次のように定義する．

$$
\beta_i(t) = \begin{cases} 0 & t = T \wedge i \notin \boldsymbol{F} \\ 1 & t = T \wedge i \in \boldsymbol{F} \\ \sum_j \beta_j(t+1)a_{ij}b_j(y_{t+1}) & 0 \leq t \leq T-1 \end{cases} \qquad (2.23)
$$

このようにして求まった$\beta_i(t)$は, 時刻$t$で, 状態$i$にいて, それ以降$\boldsymbol{y}_{t+1}^T = y_{t+1}, y_{t+2}, \cdots, y_T$を観測する確率を表していることになる．式2.15と合わせることによって,

$$P(\boldsymbol{y}|\boldsymbol{M}) = \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1) \tag{2.24}$$

となることが分かる．

Baum-Welch アルゴリズムでは，EM アルゴリズムにおける $Estep$ ははっきりと行なわずに $Mstep$ を行なう．$Mstep$ は式2.19において，$P(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}^i)$ を固定したままで $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i)$ を最大化する $\boldsymbol{\theta}$ を求めようと言うものである．式2.14，式2.19より,

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^i) &= \sum_{\boldsymbol{x}_i} P(\boldsymbol{x}_i|\boldsymbol{y}, \boldsymbol{\theta}^i)\log\{P(\boldsymbol{x}_i, \boldsymbol{y}|\boldsymbol{\theta})\} \\ &= \sum_{\boldsymbol{x}_i}\frac{P(\boldsymbol{x}_i, \boldsymbol{y}|\boldsymbol{\theta}^i)}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)}\{\log\pi_{x_i^1} + \sum_{t=1}^{T-1}\log a_{x_i^t x_i^{t+1}} + \sum_{t=1}^{T}\log b_{x_i^t}(y_t)\} \end{aligned} \tag{2.25}$$

式2.25から $\boldsymbol{\pi} = \{\pi_i\}$，$\boldsymbol{A} = \{a_{ij}\}$，$\boldsymbol{B} = \{b_i(k)\}$，はそれぞれ，$\sum_i \pi_i = 1$，$\sum_j a_{ij} = 1$，$\sum_k b_i(k) = 1$，のもとで独立に最大化することで決定できることが分かる．$\{a_{ij}\}$ について考えてみよう．$a_{ij}$ は式2.25から $a_{ij}$ に関しの部分をとりだし，式2.26を最大にする $a_{ij}$ を求めることで求められる．式中の $n_{lm}(\boldsymbol{x}_i)$ は状態系列 $\boldsymbol{x}_i$ 中で状態 $l$ から $m$ への状態遷移のある回数である．

$$\begin{aligned} &\frac{1}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)}\sum_{\boldsymbol{x}_i} P(\boldsymbol{x}_i, \boldsymbol{y}|\boldsymbol{\theta}^i)\sum_{t=1}^{T-1}\log a_{x_i^t x_i^{t+1}} \\ &\quad = \frac{1}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)}\sum_{\boldsymbol{x}_i} P(\boldsymbol{x}_i, \boldsymbol{y}|\boldsymbol{\theta}^i)\left[\sum_{l=1}^{N}\sum_{m=1}^{N} n_{lm}(\boldsymbol{x}_i)\log a_{lm}\right] \end{aligned} \tag{2.26}$$

これは，ラグランジュの未定定数法から,

$$\frac{\partial}{\partial a_{lm}}\left[\sum_{\boldsymbol{x}_i}\frac{P(\boldsymbol{x}_i, \boldsymbol{y}|\boldsymbol{\theta}^i)}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)}\sum_{l=1}^{N}\sum_{m=1}^{N} n_{lm}(\boldsymbol{x}_i)\log a_{lm} - \sum_l \lambda_l\left(\sum_m a_{lm} - 1\right)\right] = 0 \tag{2.27}$$

を解くことで求められる．この解は,

$$A_{lm} = \sum_{\boldsymbol{x}_j}\frac{P(\boldsymbol{x}_j, \boldsymbol{y}|\boldsymbol{\theta}^i)}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)} n_{lm}(\boldsymbol{x}_j) \tag{2.28}$$

と置き，

$$\hat{a}_{ij} = \frac{A_{ij}}{\sum_j A_{ij}} \tag{2.29}$$

とすることで求まる．$\{\pi_i\}$，$\{b_i(k)\}$ も同様である．

ここで，見方をかえると，式 2.28 は，

$$A_{lm} = \sum_{\boldsymbol{x}_j} \frac{P(\boldsymbol{x}_j, \boldsymbol{y}|\boldsymbol{\theta}^i)}{P(\boldsymbol{y}|\boldsymbol{\theta}^i)} n_{lm}(\boldsymbol{x}_j) = a_{lm} \frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{lm}} \tag{2.30}$$

であるので，$\hat{a}_{ij}$ は，

$$\hat{a}_{ij} = \frac{a_{ij} \dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}}{\sum_j a_{ij} \dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}} \tag{2.31}$$

として表すことができる．

具体的な手順を示すために Forward algorithm と Backward Algorithm を用いてこれを書き直す．まず $\gamma_{ij}(t)$ を定義する．

$$\gamma_{ij}(t) = \frac{\alpha_i(t) a_{ij} b_j(y_{t+1}) \beta_j(t+1)}{\sum_i \alpha_i(T)} \tag{2.32}$$

このとき，

$$\sum_{t=1}^{T-1} \gamma_{ij}(t) = a_{ij} \frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} \tag{2.33}$$

であることは明らかである．これらを用いて，式 2.31 を書き直す．また，同様に $\{\pi_i\}$，$\{b_i(k)\}$ についても更新法が定義できる．それらを合わせて，Baum-Welch アルゴリズムでの一回の更新ルールは次の通りである．

$$\hat{\pi}_i = \frac{\sum_j \gamma_{ij}(1)}{\sum_i \sum_j \gamma_{ij}(1)} \tag{2.34}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_{ij}(t)}{\sum_{j} \sum_{t=1}^{T-1} \gamma_{ij}(t)} \tag{2.35}$$

$$\hat{b}_{i}(k) = \frac{\sum_{j} \sum_{t:y_t=k}^{T-1} \gamma_{ji}(t)}{\sum_{j} \sum_{t=1}^{T-1} \gamma_{ji}(t)} \tag{2.36}$$

このように Baum-Welch のアルゴリズムは EM アルゴリズムでありながら，Forward Algorithm と Backward Algorithm 両方を用いることで $Estep$ をはっきり行なわずに $Mstep$ を行なうことができる．

音声認識の場合のように多くのサンプルに対して EM アルゴリズムを用いるには，上式を複数のサンプルに対して拡張しておかなければならない．簡単に言えば，全観測系列を順に並べて一つの系列としてみなせば良い．順に並べることで，データが $\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^n$ となった場合，$P(\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^n) = P(\boldsymbol{y}^1) \cdot P(\boldsymbol{y}^2) \cdots P(\boldsymbol{y}^n)$ とすることができる．尤度はこの対数をとって,$\log P(\boldsymbol{y}^1, \boldsymbol{y}^1, \cdots, \boldsymbol{y}^n) = \sum_i \log P(\boldsymbol{y}^i)$ とする．このとき式 2.34，2.35，2.36 はそのまま拡張して用いることができる．各データに対する $\gamma_{ij}(t)$ を $\gamma_{ij}^n(t)$ として，

$$\hat{\pi}_i = \frac{\sum_{n} \sum_{j} \gamma_{ij}^n(1)}{\sum_{n} \sum_{i} \sum_{j} \gamma_{ij}^n(1)} \tag{2.37}$$

$$\hat{a}_{ij} = \frac{\sum_{n} \sum_{t=1}^{T-1} \gamma_{ij}^n(t)}{\sum_{n} \sum_{j} \sum_{t=1}^{T-1} \gamma_{ij}^n(t)} \tag{2.38}$$

$$\hat{b}_{i}(k) = \frac{\sum_{n} \sum_{j} \sum_{t:y_t=k}^{T-1} \gamma_{ji}^n(t)}{\sum_{n} \sum_{j} \sum_{t=1}^{T-1} \gamma_{ji}^n(t)} \tag{2.39}$$

となる．

このパラメータの更新を続けていく．パラメータが変化しない，あるいは尤度が変化しなくなった点をもって収束点とし，その時の $\boldsymbol{\theta}$ を EM アルゴリズムによる推定点 $\boldsymbol{\theta}^*$ とすれば良い．

ここで注意しなければならないのは，最尤推定されたパラメータというのは通常その点 $\boldsymbol{\theta}^*$ において，

$$\left.\frac{\partial L(\boldsymbol{y},\boldsymbol{\theta})}{\partial \theta_i}\right|_{\theta^*} = 0 \qquad for\ \forall\ i \tag{2.40}$$

が成り立つのであるが，HMM の場合は異なる．例えば，状態遷移確率 $a_{ij}$ は全て独立ではなく，$\sum_j a_{ij}$ のもとで尤度を最大にしていることから，

$$\left.\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}\right|_{\theta^*} = \left.\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij'}}\right|_{\theta^*} \qquad for\ \forall\ j'\ \ s.t.\ a_{ij'} \neq 0 \tag{2.41}$$

が成り立つが，この値が 0 にはならない．これは，$\{\pi_i\}$，や $\{b_i(k)\}$ も同様である．

## 2.6 音声認識における HMM の用いられ方

まず，認識の基本的原理となるベイズの識別規則を説明する．

観測された音の列を $\boldsymbol{y}_1^t = (y_1^1, y_1^2, \cdots, y_1^t)$ とする ($y_1^i$ は音声信号をケプストラム解析し，いくつかのカテゴリーに量子化したものであったり，バンクフィルタの出力からなる多次元のベクトルであったりする．$\boldsymbol{y}_1^t$ は，それが時間 $\Delta t$ 毎の時系列となっていることを示している)．ここで，我々が知りたいのはこの $\boldsymbol{y}_1^t$ がどの単語 $\boldsymbol{W}_i$ を表しているかである．したがって，$P(\boldsymbol{W}_i|\boldsymbol{y}_1^t)$ を全ての単語について求め，その中で最も確率の高いものを正しい $\boldsymbol{W}_i$ であるとする．ここで，

$$P(\boldsymbol{W}_i|\boldsymbol{y}_1^t) = \frac{P(\boldsymbol{y}_1^t|\boldsymbol{W}_i)P(\boldsymbol{W}_i)}{P(\boldsymbol{y}_1^t)} \tag{2.42}$$

の関係を用いる．すると，

$$P(\hat{\boldsymbol{W}}_i|\boldsymbol{y}_1^t) = \max_{\boldsymbol{W}_i} \frac{P(\boldsymbol{y}_1^t|\boldsymbol{W}_i)P(\boldsymbol{W}_i)}{P(\boldsymbol{y}_1^t)} \tag{2.43}$$

から，求めたい単語が推定される．ここで，$P(\boldsymbol{y}_1^t)$ は $\boldsymbol{W}$ と無関係なので $P(\hat{\boldsymbol{W}}_i|\boldsymbol{y}_1^t)$ を求める際には無視できる．また，$P(\boldsymbol{W}_i)$ は前に述べたように用いる文法によって決定される．具体的には単語間での出現確率，$P(\boldsymbol{W}_i|\boldsymbol{W}_j)$ を用いて計算するのが通常である．

残りの部分 $P(\boldsymbol{y}_1^t|\boldsymbol{W}_i)$ を求める場合に，HMM を用いるのである．単語毎に HMM を作り，それに基づき認識を行なう．しかしながら，単語としてのサンプルは普通 HMM のパラメータ推定ができるほど多くない．特に大語彙 (1000 単語以上) で不特定話者に対するシステムの場合，訓練用データ量の問題は深刻である．そこで，音素毎のモデルをいろいろな単語からその音素を切り出すことで作る．そうすることによって音素としてのデータ数は多くなり，正確なモデルを推定することができる．これらを繋ぎ合わせて単語のモデルを作る．たとえば,「いけだ」というモデルを作る場合，/i/ , /k/ , /e/ , /d/ , /a/という音素毎のモデル 5 つを繋ぎ合わせて単語の HMM を作るのが通常である．

音声認識で音素のモデルとしてよく用いられる HMM の構造には図 2.3 のようなも

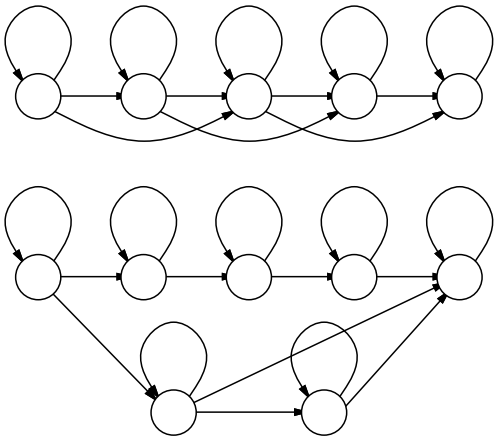

図 2.3: マルコフ連鎖の例

のがある．図中の下に示したものは，短い音素も長い音素も同じ構造の中に含んでいる．CMU (Carnegie Mellon University) において開発された SPHINX という音声認識システムでは [1][16] この構造のモデルを用いることで，母音も子音も全ての音素を表すことにしている．一方，図中の上に示された構造のものは，広く一般に用いられ

ている．どのくらいの状態数のものを用いるかは，場合によって異なる．また，各状態に定義される出力確率分布としても大きく分けて離散分布として定義する場合と多次元の正規分布として定義する場合があり，さらに正規分布も各次元独立とするか，また各次元に複数の正規分布を重ねて定義するかどうかといったことによって細分化できる．SPHINX では 256 のシンボルからなる多項分布を用いている．最近の傾向としては多次元の正規分布を用いることが多い．

# 第3章　隠れマルコフモデルを生成する

## 3.1　モデルを探す

最近，HMM を用いた音声認識が盛んであることは前にも述べたが，その理由の一つは HMM を用いるとあまり細かいことを気にしなくても “うまくいく” ことである．なぜうまくいくのか．それは，恐らく次のような理由からであろう．つまり，カテゴリの数だけモデルを用意し，各モデルに対する確率を計算し，最も良い確率を出したモデルを持ってカテゴリを決定するこの方法では，各データに対するそのモデルの “良さ” と，他のモデルのそのデータに対する “悪さ” が問題となるのである．これは各モデルがそれに対するサンプルデータに対してある程度 “良く” 他のデータに対してある程度 “悪け” れば，各モデルがサンプルデータを完全に良く表現していなくても全体としての (つまり認識できるか否かという点において) 性能はよいである．つまりはそれほど難しく考えなくてもある程度のモデルを用いておけば良いというのである．

HMM を定義に戻って眺めてみると，これは確率モデルである．得られたサンプルデータから統計的手法を用いてパラメータ推定を行なうことから，パラメータはサンプルデータの統計的性質を反映する．ゆえに，各モデルの構造 (ここでは幾つの状態遷移確率を用いるか，各状態がどのように結合されているかを指す) は同じであっても，パラメータ推定のみでカテゴリわけができるのである．

では，パラメータの推定のみでいいのであろうか．HMM は構造も含めて確率モデルなのである．したがって，モデルの良さをいうならば，どのような構造の HMM を用いるかと，そのときどのようなパラメータでモデルを定義するかの両方を合わせて考えるべきではないであろうか．

簡単に考えても，例えば/s/といった子音や/a/等の母音は定常的な音であるが，/b/

や/d/ といった音は破裂音であることから，そのモデルも母音と比べて定常的ではないもので表す方が良いだろう．一方，実際に用いられている場での研究を見ると，HMM の構造の点に於いては，人間の経験に頼っているのがほとんどである．例えば，2.6 節 に示したように SPHINX では全ての音素に対して同じモデルを用いている．これは，以前にこのグループで作ったシステム Angel の経験に基づいて選ばれている．また，他方では，各音素の出だしと終りに関してのモデルを中間の音に対するモデルと分けた方が良いという考え方や，子音と母音ではモデルを変えるという考え方もある [10]．

このような確率モデル等のモデル選択という問題に対しては AIC [17] [18] [19] [20] 等の基準が提案されている [21]．幾つかのモデルがある時にその中からもっとも良いものを選ぼうという場合，このような基準を用いてモデルを測ることによってどのモデルを選択するかを決めることはできる．すなわち，次のような手順でのモデル選択を行なおうというものである．

1. 代表的な幾つかの構造のモデルを用意しておく．
2. 一つのサンプルデータに対して複数のモデルのパラメータを推定する．
3. 各モデルに対して AIC 等の基準量を計算する．
4. その基準量の最もよいものをもってそのサンプルデータに対するモデルとする．

この手順によってより良いモデルが選ばれるであろうが，最も良いモデルであるかどうかは最初にどのようなモデルを定義しておくかにかかっている．

では，はじめにモデルを用意しておくのではなく，より良いモデルを探索していくことはできないであろうか．これが本研究の目的である．モデルを与えるのではなく，モデルを構造を含めて探索する．そうすることによって人間の負担が減り，更に全体としての性能も上がるであろう．特に，音声認識のように，サンプルデータの統計的性質が複雑であったり，量が多かったりした場合は自動的に最も良いモデルが探索されることが必要となる．

次節ではこのような視点にたって行なっている研究として，ATR に於いて最近行なわれている研究 [22] を紹介し，更に本論文で提案する方法について述べる．ATR での方法は本論文で述べる方法とは問題の設定が異なっている．

### 3.1.1 逐次状態分割法 (SSS) [22]

音声の問題として，前後の音素が何であるかによってその音素自信が影響を受けてしまうということがある．SPHINX[1] のシステムでは，人間が前後の音の並びによって 48 の基本音素を更に細分化して 500 のモデルを作ることでこの問題を取り扱っている．一方，鷹野らの提案した逐次状態分割法 (Successive State Splitting : SSS) では，図 3.1 に示す方法によって，前後のコンテキストによる方向と，時間方向へ状態を増やしていく．これによって HM-Net と呼ぶ HMM を形成し，これを用いて認識をする．用いた HMM その他の条件は次の通りである．

**認識タスク**

6 子音 (/b/ , /d/ , /g/ , /n/ , /m/ , /N/)

**各状態の分布**

単一ガウス分布，対角共分散行列.

**コンテキスト要因数**

3 要因 (先行音素，当該音素，後続音素)

**時間方向への状態分割制限**

1 モデル当たりの状態数を最大 4 に制限.

**パラメータ**

logpow,cep(16),$\Delta$logpow,$\Delta$cep(16) からなる 34 次元ベクトル.

**分析条件**

サンプリング周波数 12kHz,16bit 量子化，20msec ハミング窓，フレーム周期 5msec.

具体的な手順は次の通りである．

1. 図 3.1 に示された初期モデルを 6 つ用意しそれを連結して初期の Net を形成する．

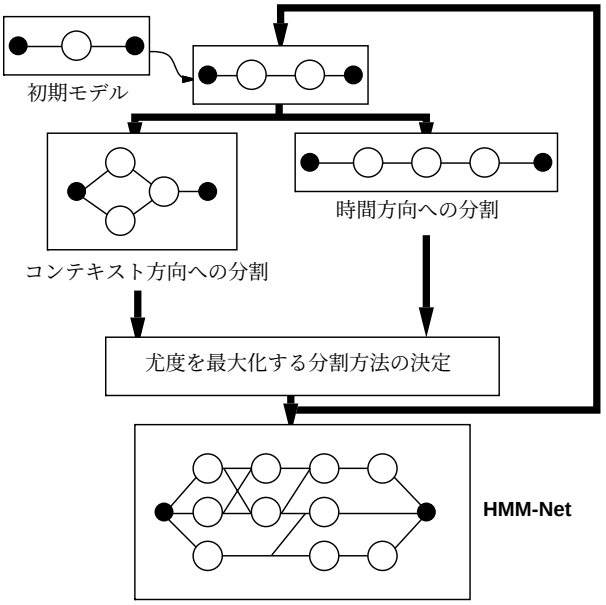


図 3.1: SSS の原理

2. 各状態に混合数 2 の混合ガウス分布 (対角共分散行列) をもつ出力分布を定義し，パラメータを推定する．

3. 各状態の共分散行列から，被分割状態を決定する．

4. 時間方向，コンテキスト方向の両方に分割を試してみて良い方向に分割する．

これによって得られた HMM-Net を図 3.2 に示す．

## 3.2 提案するアルゴリズム

全節で述べた方法には，幾つかの問題点があるのではないだろうか．SSS では，Net として HMM を生成していく．これはつまり他のサンプルとの関係において，そのモデルの良さが測られるということである．しかし，HMM が確率モデルであることを考えると，他のサンプルを見なくても一つのサンプル毎にモデルを定義できる．これは，他のモデルとの間で尤度の比較をしていないことから，全体の認識率が上がることは考えていないようにみえるが，データを正確に表すという点からはこの方が自

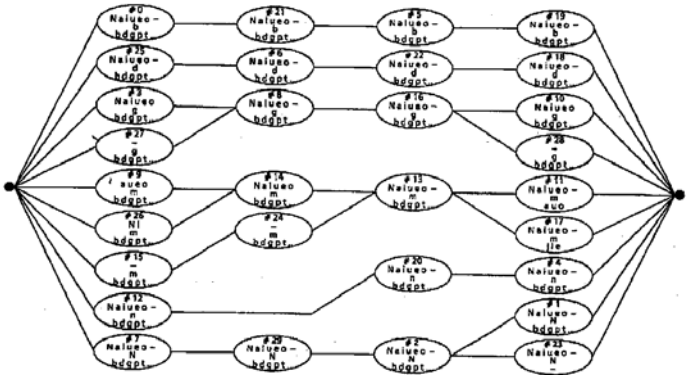

図 3.2: HMM-Net

然である．また，他のモデルとの比較の中からパラメータを決定する [23] 方法も考えられるが，カテゴリの数が増えた場合，この方法は適さないと考える．もし，同一カテゴリ毎に求めたモデルによってデータが区別できないのであれば，それはそれらのデータが元々区別のできないものであることを示すだけである．

どうやってモデルを探すことができるのだろう．HMM のモデルを決める大きな要因は，幾つの状態を用いて HMM を定義するか，また，それぞれの状態間の状態遷移確率をどのように定義するかである．モデルを探すということは，それらを変化させていこうということに他ならない．構造を変えた場合，同時にその構造毎に最適なパラメータの値も変化する．したがって，構造を変化させる毎にパラメータ推定を行なわなければならない．うまく構造を変化させていかなければ，モデルを変えた毎にパラメータ推定をし直すということは，効率が悪い．これらを考え合わせた上で，効率良く状態数と状態間の状態遷移確率を増やしていくアルゴリズムを提案する．

### 3.2.1 状態数を増やす

まず HMM の構造を考える際に，全体を幾つの状態にするかということが問題になる．これに対し，状態を一つづつ増やしていくことによって状態数を決めることができないかと考えてみる．そのとき，ランダムに状態を増やすのではなく，効率良く増やしていけることが望ましい．問題となるのは，新しく増やす状態をどの場所に定義するか，そのときの初期値はどうするか，どこまで増やすか，である．3 つの問題について順に見ていく．

#### 被分割状態の決定

どの状態を分割するかの決定には，その状態の出力確率分布と，その状態 にどのくらいの時間留まるかの期待値を見て決めれば良いと考えられる．出力確率分布 は，そこで受けとることのできるシンボルを反映する．これがばらついている 場合，そのエントロピーも増えるはずである．したがってその状態の悪さとし てエントロピーを用いることができる．また，ある状態に長くいることを避 けるために，そのエントロピーにある状態に留まる時間の期待値を乗じること にした．すなわち，

$$\text{状態 } i \text{ に留まる時間の期待値} = \sum_{t=1}^{T-1}\sum_{j=1}^{N}\alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1) \quad (3.1)$$

$$\text{状態 } i \text{ のエントロピー} = \sum_{k}^{K} b_i(k)\log b_i(k) \quad (3.2)$$

$$\text{被分割状態の決定の際の比較量} = \sum_{t=1}^{T-1}\sum_{j=1}^{N}\alpha_i(t)a_{ij}b_j(y_{t+1})\beta_j(t+1) \times \sum_{k} b_i(k)\log b_i(k). \quad (3.3)$$

として，この値が最大となる状態を分割することにする．

#### 初期値の設定

被分割状態を決めた後，それを分割し，EM アルゴリズムによってパラメータを推定し直す必要がある．このとき，どのように分割し，それぞれのパラメータの初期値を決定するかが問題となる．

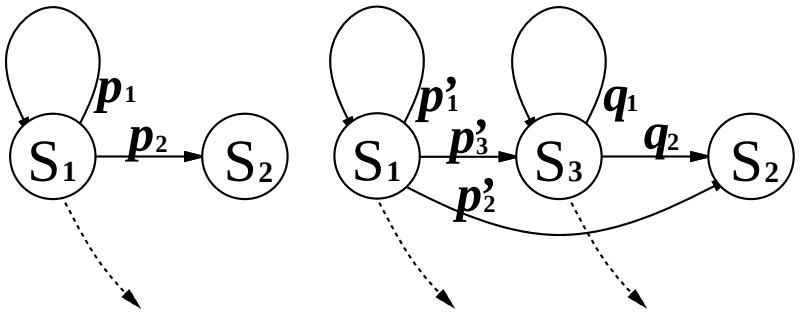


図 3.3: 状態分割

図 3.3 の 2 つのモデルを考える．左側の図に於いて，状態 $S_1$ に $n$ 回留まり，次の時間に $S_2$ にいる確率は確率 $P_n$ は $p+q \leq 1$ に注意して，

$$P_n = p_1^n p_2 \tag{3.4}$$

となる．一方これに状態 $S_3$ を挿入した右側の図ではこれは式 3.5 のようになる．

$$\begin{aligned} P'_n &= {p'}_1^n {p'}_2^n + {p'}_3^n q_2^n \sum_{i=0}^{n-1} {p'}_1^i q_1^{n-i-1} \\ &= \begin{cases} {p'}_1^n {p'}_2 + q_2 {p'}_3 \dfrac{{p'}_1^n - q_1^n}{{p'}_1 - q_1} & {p'}_1 \neq q_1 \\ {p'}_1^n {p'}_2 + n q_2 {p'}_3 {p'}_1^{n-1} & {p'}_1 = q_1 \end{cases} \end{aligned} \tag{3.5}$$

ここでもし，$p'_2 = p_2$ , $q_1 = p_1$ , $q_2 = p_2$ , $p'_1 + p'_3 = p_1$ ならば，式 3.5 の ${P'}_n$ は式 3.4 の $P_n$ と等しくなる．したがって，これらの条件を満たした上で状態 $S_1$ と $S_3$ の出力確率分布が等しければ，2 つのモデルは全く等しくなる．

分割する前は EM アルゴリズムによって収束しているとすれば，被分割状態を定め，図 3.4 のように分割し，$S_1$ と $S_3$ の出力確率分布を同じものとして定義することによって，最尤推定されたモデルと全く等価なモデルを出発点とすることができる．モデルを複雑にしていくことができる．もし，分割する前のモデルが完全にサンプルデータを表現しているのならば，学習しても尤度は上がらないが，そうでないならば，学習していくことで，より良いモデルへと進んでいくことができる．よってこのようにして初期値を定めることができる．

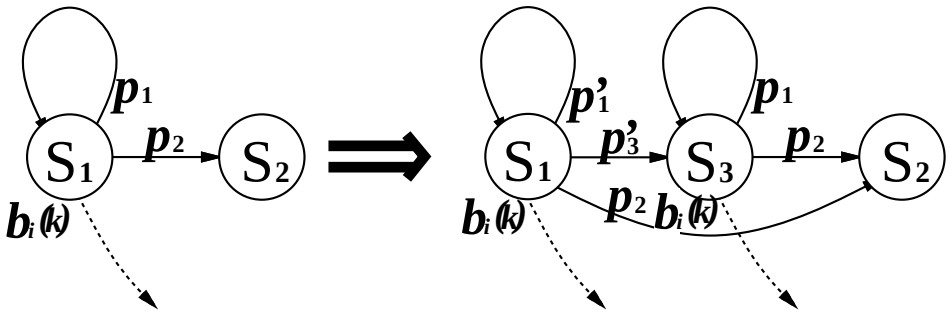


図 3.4: 状態分割

**どこまで状態を増やすか**

ここまで，しばしば “モデルが良い” という表現を用いていた．モデルの良さとは何であろう．以上のように状態数を増やしていく場合，等価なモデルとして状態を挿入できることから，尤度のみをモデルの良さとするのならば，幾ら状態を増やしていってもモデルの良さは変わらない．しかし，これは不自然である．なぜなら，サンプルデータは限られた数しかないにも関わらず，推定すべきパラメータが増えてしまっているからである．推定しなければならないパラメータの数も含めて，何か別の基準量が必要である．本研究ではこの基準量として AIC [24] [20] を用いることにした．

モデル選択の基準として用いられる AIC は，サンプルデータに対するモデルの当てはまりの良さを表す項とモデルの複雑さを表す項から成り立つ．この 2 つのバランスをとることでモデルの良さを表そうというものである．実際の分布 $p(y)$ とモデルの確率分布 $f(y|\theta^*)$(ただし $\theta^*$ は最尤推定された点であるとする) との距離を Kullback 情報量,

$$\begin{aligned} D(p,f) &= \int p(x)\log\frac{p(x)}{f(y|\theta^*)}d\mu \\ &= \int p(x)\log p(x)d\mu - \int p(x)\log f(y|\theta^*)d\mu \end{aligned} \tag{3.6}$$

を用いて定義する．式 3.6 の第 1 項はモデルと関係のない項なので無視し，第 2 項の期待値をサンプルデータから推測することによって，AIC は次のように

定義される．

$$\mathrm{AIC} = (-2)\sum_i \log f(y_i|\theta^*) + 2(\text{推定するパラメータ数}) \tag{3.7}$$

HMM に対して AIC を用いる場合，式 2.41 を考慮に入れなければならない．つまり，AIC は最尤推定点のまわりで，

$$\left.\frac{\partial f(y|\theta)}{\partial \theta_i}\right|_{\theta^*} = 0 \qquad\qquad for\ \forall\ i$$

が成り立つとしての近似である．これは HMM において，Baum-Welch アルゴリズムの収束点では成り立たない．これは $\sum_j a_{ij} = 1$ 等の条件があるからである．数えあげれば，丁度状態数の 2 倍の条件があり，この条件の数だけ独立でないパラメータが存在することになる．$\{a_{ij}\}$ を例にとると，ある 0 でない $a_{ij}$ に対し $a_{ij} = 1 - \sum_{j'} a_{ij'}$ と書いて消去することによってこの問題は避けられ，AIC は同様に定義できる．この時 AIC は，

$$\mathrm{AIC} = (-2)\sum_i \log f(y_i|\theta^*) + 2(\text{パラメータ数} - 2 \times \text{状態数} + 1) \tag{3.8}$$

とすればよい．

この AIC を用いて，状態をどこまで増やすかを決める．すなわち状態を増やしても AIC が減らない場合，そこで状態を増やすことをやめればよい．

以上をまとめて，状態を増やしていく手順は次のようになる．

1. 被分割状態を決める．
2. 初期値を与える．
3. Baum-Welch アルゴリズムを行ない，最尤推定点を求める．
4. AIC を計算し，前の時よりも良くなれば 1 へ戻る．そうでない場合はそこで終了する．

### 3.2.2 状態遷移確率を増やす

HMM では，状態の数とは別に，どの状態からどの状態への状態遷移確率を定義するかの問題がある．状態遷移確率も段々に増やしていくことで，より良いモデルを推定できないであろうか．

状態を増やした手順では，EM の収束点からそれと全く等価なモデル出発し，それを拡張する形でモデルを複雑にしていく．しかしながら，状態遷移確率を増やす場合には，等価なモデルが無い．つまり，状態遷移確率を増やすということは全く違うモデルへ移るということになる．したがって，状態遷移確率を増やす場合は，それによって得られるモデルが実際にうまくいっているのかは，EM アルゴリズムを行なった後にそのモデルの基準量を比較してみるしかない．

つまり，新しく状態遷移確率を定義する場合，それによって尤度が上がっていきそうな状態遷移確率を新しく付け加えて学習するしかないのである．

では，どこに新しい状態遷移確率を付け加えるのが良いのであろう．どの状態遷移確率を付け加えれば良いかを決めるには状態遷移確率を新しく加え，EM アルゴリズムを行なうことによって最終的にどのくらい尤度が上がるのかを知らなければならない．これは予測できない．そこで，EM の 1 ステップでどのくらい尤度があがったかを見ることにする．しかし，ここでもまた問題がある．状態遷移確率を増やす候補は数が多く，それぞれに 1 度づつ EM アルゴリズムを行なうのは時間がかかり過ぎる．そこで，EM の 1 ステップでの各状態遷移確率の増加分と $\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})/\partial a_{ij}$ との積でこれを予測し，この値をもって新しく定義する状態遷移確率を決定したい．EM の 1 ステップでの状態遷移確率の増加分と尤度の変化分について順に見ていく．

#### 状態遷移確率の増加分

EM の 1 ステップでどのくらい状態遷移確率が変わるかであるが，$\{a_{ij}\}$ について式 2.31 を見てみる．

$$\hat{a}_{ij} = \frac{a_{ij}\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}}{\displaystyle\sum_{j} a_{ij}\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}}$$

この式では，分母と分子それぞれに $a_{ij}$ がかかっていることから，もし $a_{ij}=0$ ならばその状態遷移確率はこの更新ルールでは変化せず，したがって HMM の構造は変化しない．では，いままで 0 であった $a_{ik}$ に小さな値 $\delta$ を入れ，新しく状態遷移確率を定義したとして新しく推定される $\hat{a}_{ij}$ がどのようになるかをみてみる．

$$\hat{a}_{ij} = \frac{a_{ij}\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}}{\displaystyle\sum_{j:j\neq k} a_{ij}\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} + \delta\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}} \qquad j\neq k \qquad (3.9)$$

$$\hat{a}_{ik} = \frac{\delta\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}}{\displaystyle\sum_{j:j\neq k} a_{ij}\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} + \delta\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}} \qquad (3.10)$$

$\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})/\partial a_{ik}$ は有限値をとり，発散することはない．これより，もし $\delta$ が十分小さければその項は各式の分母の第 1 項に比べて十分小さいことになり，無視できる．したがって，$a_{ij}$ $(j\neq k)$ に対しては，この条件の下でほとんど変化しないことになる．一方 $a_{ik}$ の増加分であるが，これは以上の結果と式 2.41，及び $\sum_j a_{ij}=1$ を考えて，

$$\begin{aligned}
\hat{a}_{ik}-a_{ij} &= \frac{\delta\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}}{\displaystyle\sum_{j:j\neq k} a_{ij}\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} + \delta\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}} - \delta \\
&\simeq \delta\left[\frac{\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}}{\displaystyle\sum_{j:j\neq k} a_{ij}\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}} - 1\right] \\
&= \delta\left[\frac{\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}}}{\dfrac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}}} - 1\right] \qquad j\neq k \ \wedge \ a_{ij}\neq 0 \qquad (3.11)
\end{aligned}$$

とかける．$\{\pi_i\}$,$\{b_i(k)\}$ からなる他のパラメータも $\delta$ が微小な場合には同様にそ

の変化はほとんどない．したがって，新しく状態遷移確率 $a_{ik}$ を定義したときのパラメータ変化分は $a_{ik}$ のみについて見れば良く，その値は式 3.11 から推定できる．これが正であれば，その状態遷移確率は増加する可能性があり，したがって新しく採用しても良いと考えられる．

**尤度の増加分**

EM アルゴリズムのの 1 ステップで尤度はどのくらい変化するだろうか．実際の尤度の変化分は一度 EM アルゴリズムのステップを行なってみないと分からない．しかし，試してみたい状態遷移確率全てに EM アルゴリズム 1 ステップを行なうことになり，これは時間がかかる．そこで，$\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})/\partial a_{ik}$ を用いて予測する．これも式 2.32 を参考にして，

$$\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} = \frac{\sum_{t=1}^{T-1} \alpha_i(t) b_j(y_{t+1}) \beta_j(t+1)}{\sum_i \alpha_i(T)} \tag{3.12}$$

から求めることができる．以上から，$a_{ik}$ を新しく十分小さい $\delta$ として定義する場合の尤度の増加分を，次のように書くことができる．

$a_{ij}$ $(j \neq k)$，$\{b_i(k)\}$，$\{\pi_i\}$，がほとんど変化しないことを考えて，

$$\begin{aligned} \Delta_{ij} \log P(\boldsymbol{y}|\boldsymbol{\theta}) &\equiv (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \\ &\simeq \delta \left[ \frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ik}} \Big/ \frac{\partial \log P(\boldsymbol{y}|\boldsymbol{\theta})}{\partial a_{ij}} - 1 \right] \\ &\qquad \times \frac{\sum_{t=1}^{T-1} \alpha_i(t) b_k(y_{t+1}) \beta_k(t+1)}{\sum_i \alpha_i(T)} \end{aligned} \tag{3.13}$$

となる．

このように定義された $\Delta_{ij} \log P(\boldsymbol{y}|\boldsymbol{\theta})$ を候補に上がっている状態遷移確率それぞれについて求め，もっとも大きい値を示した状態遷移確率を採用することにする．

以上の方法では，全ての候補に対して $\Delta_{ij} \log P(\boldsymbol{y}|\boldsymbol{\theta})$ を求めるのに Forward Algorithm と Backward Algorithm を一回行なえば良い．

採用した後はそこに小さな値を代入し，EM アルゴリズムを行なって新しいモデルを作ることにする．また，状態数を増加させた場合と同様に，AIC を用いて，どこまで状態遷移確率を増やすかを決めることにする．つまり，EM アルゴリズムが収束する毎に AIC を計算し，AIC を最小にするモデルを最終的に採用することにする．

## 3.3 計算機を用いてのシミュレーション

このアルゴリズムを用いて音声データを用いての実験を行なう前に，計算機を用いてシミュレーションを行なった．確率過程から得られるデータとして，複数の確率モデル (HMM) を作り，それらから出るデータを複数カテゴリのデータとした．実験の内容と，その結果について順に述べる．

### 3.3.1 実験の詳細

ここでは，実際に音声を用いて実験を行なう前に，人工的に作った 5 つの隠れマルコフモデルを信号の発生源として用い，シミュレーションを行なった．5 つのマルコフモデルは，それぞれ状態数，状態遷移，出力確率分布が異なっており．それぞれが出力する信号もこれに応じて異なったものとなっている．それぞれのマルコフモデルの出力した信号を 5 つのカテゴリとみなす．このとき，それらの信号がどの隠れマルコフモデルから発生されたかを認識するという問題を扱うことにする．発生源として用いたマルコフモデルを図 3.5 に示す．

各モデルの各状態に定義された出力確率分布は離散分布で，6 つのシンボルを出力するものとする．従って，各モデルから出力される信号は，0 ～ 5 で示されるシンボルが並んだものである．

乱数を振って各モデルから出力された信号を求める．例えば，モデル 1 から出力された系列を示すと，

$$
\begin{aligned}
\boldsymbol{y}_1 &= 0,0,2 \\
\boldsymbol{y}_2 &= 0,2,5,2 \\
\boldsymbol{y}_3 &= 0,0,1,0,0,1,0,1,2
\end{aligned}
$$

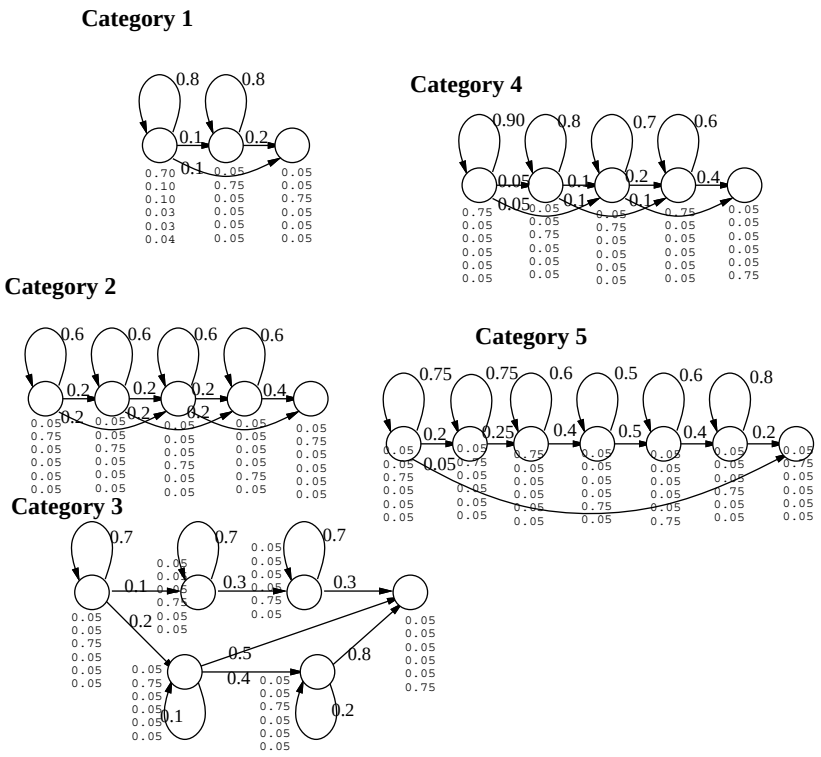


図 3.5: 信号発生用の HMM

$$
\begin{aligned}
\boldsymbol{y}_4 &= 0,1,0,0,1,1,2 \\
\boldsymbol{y}_5 &= 0,0,0,1,2,5,0,0,0,0,0,0,4,0,2,0,2,2 \\
\boldsymbol{y}_6 &= 0,0,0,0,0,0,2 \\
\boldsymbol{y}_7 &= 1,2 \\
\boldsymbol{y}_8 &= 2,0,1,1,1,1,0,3,1,3,1,1,5,1,2,1,1,1,1,1,1,2 \\
\boldsymbol{y}_9 &= 2,2,3,2
\end{aligned}
\tag{3.14}
$$

となってる.

実験の手順としては，次の通りである.

- 各モデルから，訓練用と認識実験用に 1000 個づつ計 2000 個の系列を発生させる.
- カテゴリの数だけ，すなわち 5 つモデルを作り，各データに対して構造やパラ

メータを推定する.

- 認識する際には，各モデルからそのデータの発生された確率 $P(\boldsymbol{y}|\boldsymbol{M}_i)$ を求め，それを最大にする $\boldsymbol{M}_i$ をもってその信号の属するカテゴリとする.

まず，信号発生用のモデル自身でデータの認識実験を行なった．結果を表 3.3.1 に示す．対角線に太い字で書かれたのが正解率，右端に書かれたのが誤認識率である.

表 3.1: 認識結果　元のモデルを用いた時

| **category** | **1** | **2** | **3** | **4** | **5** | 誤認識率 |
|---|---|---|---|---|---|---|
| model 1 | **90.9**% | 0.6% | 1.7% | 6.2% | 0.6% | 9.1% |
| model 2 | 1.4% | **94.5**% | 2.7% | 0.1% | 1.3% | 5.5% |
| model 3 | 1.1% | 3.1% | **92.7**% | 1.6% | 1.5% | 7.3% |
| model 4 | 7.6% | 0.4% | 2.1% | **89.7**% | 0.2% | 11.3% |
| model 5 | 1.0% | 0.9% | 2.8% | 0.3% | **95.0**% | 5.0% |
| 計 | | | | | | 7.64% |

信号源のモデルを用いた認識率が表 3.3.1 の通りであることから，どのようなモデルを用いてもこれ以上の認識率が得られるとは思われない.

### 3.3.2　実験の結果及び考察

では，提案したアルゴリズムを用いて，構造を変化させていった場合はどのようになるであろう．出発点となったモデルは図 3.6 の通りである．このモデルを各カテゴリのサンプルデータに対する初期モデルとしてパラメータ推定をし，そこから，状態数を増加していくことにする.

用いたアルゴリズムは具体的に次の通りである.

1. 状態を増やし，Baum-Welch のアルゴリズムでパラメータを推定する.
2. AIC の基準で AIC が減らなくなったらば 3 へ，そうでなければ 1 へ.

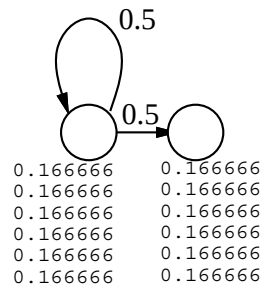


図 3.6: 出発点の HMM

3. 状態推移確率を増やし，Baum-Welch のアルゴリズムでパラメータを推定する．

4. AIC の基準で AIC が減らなくなったらば 1 へ，そうでなければ 3 へ．また，1 つも状態遷移確率が増加しない場合は終了する．

このアルゴリズムを用いて，各カテゴリに対してモデルを作ることにする．作っていく過程で，モデルとしてどのようなことが起こっているかを見たい．外から見えるのは尤度のみである．Category 5 のデータにアルゴリズムを適応させたときの尤度の推移を図 3.7 に示す．

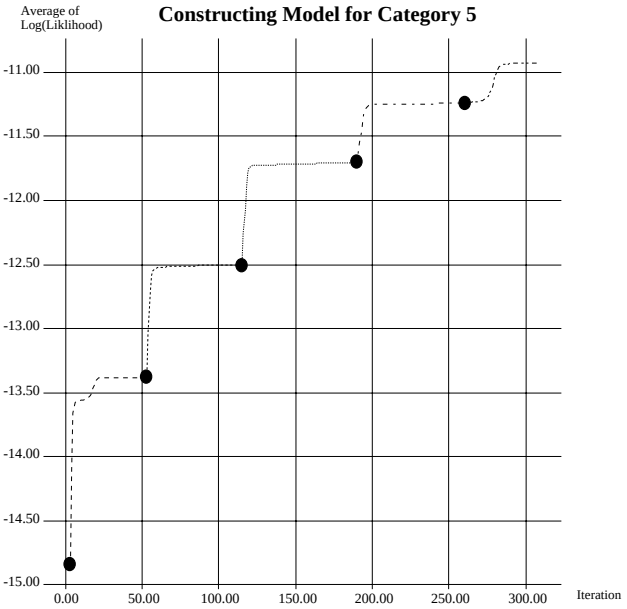


図 3.7: Category 5 に対する学習の結果

横軸は EM アルゴリズムの回数，縦軸は尤度，また図中の黒い丸は各収束点を示す．尤度が増加していく様子が分かる．状態数や，状態遷移確率を増やす毎に異なるモデルに移り，尤度が段階的に上がっていくことが示されている．

具体的な構造を，他のカテゴリに対してのモデルも含めて図 3.8 に示す．この図を見ると分かるが，信号発生用の隠れマルコフモデルの構造を良く反映しているものもあるが，そうではないものもある．同じ構造となる必要はない．尤度としてより高くなり，AIC の基準で見てよくなればよい．

これらを用いての認識率を 3.3.2 に示す．

また，比較のために 3 状態，5 状態，7 状態の隠れマルコフモデルを用いて，同様の実験を行なった結果を上の結果とまとめて図 3.3.2 に示す．この図には，各カテゴリに対する認識率と，全体に対する認識率のみを示してある．

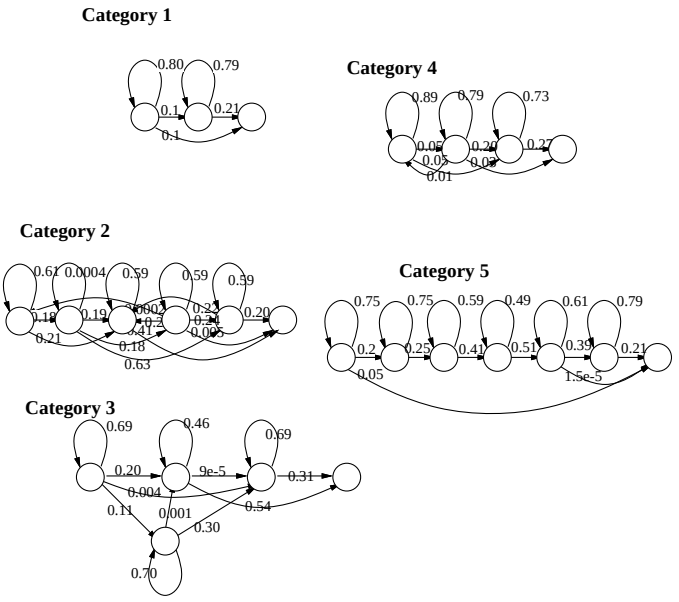


図 3.8: 構造を変化させた結果

このときのそれぞれの AIC を求めてみよう．表 3.3.2 に示す．

AIC を見る限り，このアルゴリズムによって求められたモデルはかなり良いことがわかる．

この実験から，提案するアルゴリズムの有用性がある程度示された．次章ではこれを用いて行なった音声認識の実験について述べる．

表 3.2: 認識結果

| **category** | **1** | **2** | **3** | **4** | **5** | 合計 |
|---|---|---|---|---|---|---|
| 3 状態 | 83.4% | 81.7% | 87.4% | 87.8% | 84.3% | 84.92% |
| 5 状態 | 90.3% | 91.7% | 88.9% | 89.5% | 90.7% | 90.22% |
| 7 状態 | 89.8% | 92.3% | 90.0% | 88.3% | 93.0% | 90.68% |
| 信号発生用のモデル | 90.9% | 94.5% | 92.7% | 89.7% | 95.0% | 92.56% |
| アルゴリズムを用いた場合 | 92.0% | 94.3% | 91.9% | 87.7% | 93.8% | 91.94% |

表 3.3: 各モデルの AIC

| **category** | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| 3 状態 | 13596 | 13779 | 12570 | 23891 | 30836 |
| 5 状態 | 12547 | 12752 | 12134 | 22817 | 28762 |
| 7 状態 | 12570 | 13018 | 12141 | 22839 | 25228 |
| 信号発生用のモデル | 12541 | 12758 | 11421 | 22538 | 25239 |
| アルゴリズムを用いた場合 | 12534 | 12759 | 11577 | 22861 | 25226 |

# 第4章　実験

## 4.1　実験の条件等

以上の方法を用いて音素に対し HMM を生成し，認識を行なった．実験は後続母音を/e/に限定した場合で行なった．この実験は e-set と呼ばれるものである．行なった実験の条件等は以下の通りである．

**使用したデータ**

ATR 研究用日本語音声データベース [25] [26] [27].

**認識タスク**

(/m/, /s/, /t/, /k/, /d/)

**認識タスク**

後続音素は e に限定した (e-set).

**各状態の分布**

離散分布，256 のシンボル.

**時間方向への状態分割制限**

1 モデル当たりの状態数を最大 20 に制限.

**シンボルの定義**

logpow,mel-cep(15),$\Delta$logpow,$\Delta$mel-cep(15) からなる 32 次元ベクトルをベクトル量子化.

**分析条件**

サンプリング周波数 20kHz,16bit 量子化，20msec ハミング窓，フレーム周期 5msec.

**使用したデータ**であるが，これには音素毎にラベル付けがされている．このラベルにしたがって，音素を切り出し，実験を行なった．

**認識タスク**に選んだ 5 つの子音は，統計的処理をすることから，サンプルとして得られる数の多い 5 つの子音を用いた．

なお，実験は単語発生データ (5240 単語) の内，偶数番目を HMM 生成用に用い，奇数番目を認識実験用に用いた．4.2 節でどのように信号を処理してシンボルにしたかを示す．

## 4.2 信号処理

今回の HMM の各状態に定義された出力確率分布は多項分布であるので，音を時間毎に切り出し，周波数を求める処理を行ない，シンボル化しなければならない．信号処理の流れは，大きく見て図 4.1 である．

### 4.2.1 前処理

**ラベルにしたがって切り出す**

前に述べたように ATR の音声データには各音素の存在時間がデータ毎に書かれており (2.5msec 単位)，これをもとに音素の部分を切り出した．

**低域強調**

音声信号を処理する前には，通常この pre-emphasis と呼ばれる処理を施す．実際に Low-Pass-Filter を用いる場合もあるが，今回は信号を $1 - az^{-1}$ で示されるフィルタに通過させることでこの処理を行なった [28] [1]．$a$ の値はできるだけ 1 に近い値がいいとされている．今回はこの値として 0.97 を用いた．

**切り出してハミング窓をかける**


図 4.1: 信号処理の手順

20msec 毎に音を切り出し，それを 5msec 毎にずらしながら行なった．切り出した各信号には窓関数をかけなければならないのだが，窓関数としてはハミング窓用いた．

### 4.2.2 線形予測係数を用いて，

#### LPC ケプストラム係数を求める

自己相関法を用いて LPC ケプストラム係数を求めた．LPC の極のは 18 次として計算し，最終的に 15 次元のケプストラム係数を求めた．

ある時刻で切り出したデータに対する LPC の係数を $\{a_i\}$ とすると (今回の実験では，$1 \leq i \leq 18$) LPC ケプストラム係数 $\{C_i\}$ は

$$C_m = a_m - \sum_{k=1}^{m-1} \frac{k}{m} C_k a_{m-k}, \qquad m > 0 \tag{4.1}$$

#### LPC-メルケプストラム係数を求める

人間の聴覚では，高い周波数になると，周波数の分別能力が下がる．このように，感覚的な周波数軸は実際の線形な周波数軸とは異なる．これはメル尺度と呼ばれている [29]．これを z 平面上の双一次変換を用いて近似する方法が提案されており [30] [31]，これを用いて LPC ケプストラム係数から LPC- メルケプストラムを求めることができる [32]．

**双一次変換**

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{4.2}$$

これをもとに，

$$\sum_{n=0}^{\infty} M_n \tilde{z}^{-n} = \sum_{n=0}^{\infty} C_n z^{-n} \tag{4.3}$$

から，LPC 係数を $\{C_i\}$ をもとに LPC メルケプストラム係数を $\{M_i\}$ を求めることができる．この再帰式は式 4.4 の通りである．

$$\begin{aligned} M_m^{(i)} &= \begin{cases} C_i + \alpha M_0^{(i+1)} & m = 0 \\ (1-\alpha^2) M_0^{(i+1)} + \alpha M_1^{(i+1)} & m = 1 \\ M_{m-1}^{(i+1)} + \alpha (M_m^{(i+1)} - M_{m-1}^{(i)}) & m > 1 \end{cases} \\ i &= \infty, \cdots, 1, 0 \\ M_i &= M_i^{(0)} \end{aligned} \tag{4.4}$$

$i$ が無限に続く形になっているが，実際には

$$C_i = 0, \qquad i > 18$$

であるから， $i = 18, \cdots, 1, 0$ でよい． $\alpha$ の値は 0.63 とした．

**ベクトルを作る**

音声を認識する場合，その時刻でのケプストラムだけでなく，ケプストラムをベクトルとしてみた時のその大きさや，前後のフレームのベクトルの差を用いると認識しやすくなることが分かっている．特に差をとるフレームは，前と後

に 20msec 程度離れいると良いとされている．これに基づき，以下のような量を求め，メルケプストラムと合わせて，32 次元ベクトルを作った．

$$\begin{aligned} \Delta \boldsymbol{M} &= \boldsymbol{M}^{+\Delta t} - \boldsymbol{M}^{-\Delta t} && (4.5) \\ P &= \log \sum_{i}^{N} M_i^2 && (4.6) \\ \Delta P &= P^{+\Delta t} - P^{-\Delta t} && (4.7) \end{aligned}$$

なお，$\Delta t = 20\text{msec}$ である．

以上の手続きから 32 次元のベクトルが 5msec 毎に並んでいる時系列ができた．これを認識したい単語に施すことで，全ての単語を 32 次元ベクトルの時系列としてみなすことにする．

## 4.3 ベクトル量子化

次に，これらのベクトルをシンボルにするために行なったベクトル量子化について述べる [33]．ベクトル量子化とは，連続な空間内のベクトルを量子化することである．すなわち，ベクトルを幾つかのカテゴリに分類するのである．

量子化の手順は，まず，ベクトル間の距離を定義する．次に代表ベクトルを分類したいカテゴリの数だけ用意する．量子化を行なう際には，量子化したいベクトルが代表ベクトルの内どれに近いかをもってそのベクトルの属するカテゴリとすれば良い．

**距離**

前に定義した 32 次元ベクトル間の距離を次のように定義する．

$$\begin{aligned} PDCEP &= \sum_{i=1}^{15} (M_i^r - M_i^t)^2 + W_d \sum_{i=1}^{15} (\Delta M_i^r - \Delta M_i^t)^2 \\ &\quad + W_p (P^r - P^t)^2 + W_{dp} (\Delta P^r - \Delta P^t)^2 \qquad (4.8) \end{aligned}$$

$\{W_d, W_p, W_{dp}\}$ であるが，$\{0.5, 0.0, 0.03\}$ あるいは $\{0.8, 0.01, 0.05\}$ が良いとされている [1]．今回は後者を用いた．

**代表ベクトルの求め方**

ベクトル量子化の手順は以下の通りである．

1. まず，全体の中で，もっとも距離のはなれた 2 つのベクトルを求め，代表ベクトルとする．
2. その 2 つのベクトルにしたがって各 partition を 2 分割する．
3. 代表ベクトルに対する平均誤差を計算する．誤差が収束していなければ，partition 内の平均を取り代表ベクトルとして 2 へ戻る．
4. 収束していた場合，各 partition 内でもっとも離れた 2 つのベクトルを求め，代表ベクトルとし，2 へ戻る．
5. 256 の partition に分かれたら終了する．

これを用いて，256 の代表ベクトルを求めた．

なお，ベクトル量子化を行なうに当たって用いたデータであるが，全てのデータを用いるのは時間がかかり過ぎるため，ATR 研究用音声データから音韻バランス単語としてまとめられているサンプル (351 単語) を用いた．

## 4.4 結果

まず，提案するアルゴリズムを用いて，モデルを生成した．その結果として得られたモデルの状態数と状態遷移数を表 4.1 に示す．あまり大きなモデルにならないように，状態数を 20 に制限している．

表 4.1: 状態数

| 子音 (Sample 数) | m(72) | k(107) | t(66) | s(75) | d(28) |
|---|---|---|---|---|---|
| 状態数 | 11 | 20 | 20 | 13 | 9 |
| 状態遷移の数 | 32 | 71 | 64 | 40 | 27 |

表 4.2 は，このアルゴリズムを適応させた結果のモデルと，比較のため，3,5,10,15,20 の状態数の隠れマルコフモデルを用いて行なった実験について，それぞれの誤認識率を示したものである．通常，隠れマルコフモデルを用いての認識においては，最も確率の高いもののみではなく，上から順に数番目までを可能性のあるものとして残しておく場合が多い．右端の括弧内の数字は，このことを示すため，5 つのモデルの中で，上位 2 つに選ばれた確率である．また，図 4.2 に子音/d/に対し，モデルを変化させていった結果として得られたモデルを示す．なお，比較のために用いたモデルは，図 4.3 示されたように状態数を増やしていったものである．

表 4.2: 誤認識率

| 子音 (Sample 数) | m(72) | k(107) | t(66) | s(75) | d(28) | 計 |
|---|---|---|---|---|---|---|
| 各 3 状態 | 4.2 % | 15.0 % | 18.2 % | 0.0 % | 17.9 % | 10.3(2.3) % |
| 各 5 状態 | 9.7 % | 12.1 % | 12.1 % | 0.0 % | 14.3 % | 9.2(1.1) % |
| 各 10 状態 | 4.2 % | 9.3 % | 12.1 % | 0.0 % | 25.0 % | 8.0(2.6) % |
| 各 15 状態 | 4.2 % | 15.9 % | 16.7 % | 0.0 % | 28.6 % | 11.2(3.4) % |
| 各 20 状態 | 15.3 % | 19.6 % | 18.2 % | 2.7 % | 16.7 % | 14.4(3.4) % |
| 構造探索 | 2.8 % | 8.4 % | 15.2 % | 0.0 % | 28.6 % | 8.3(1.4) % |

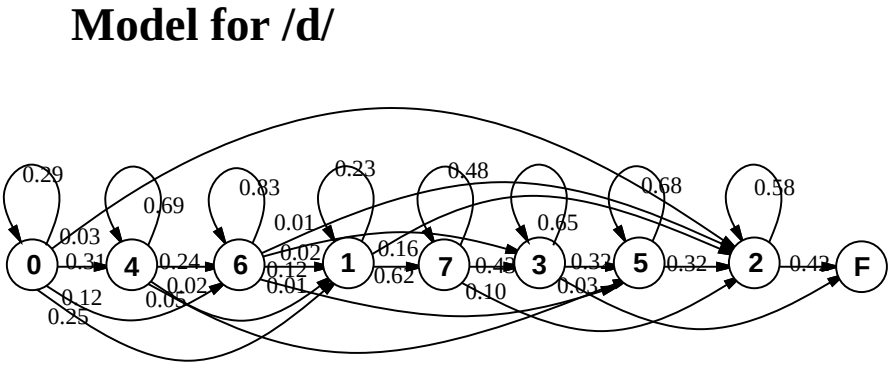


図 4.2: /d/に対してモデルを変化させた結果

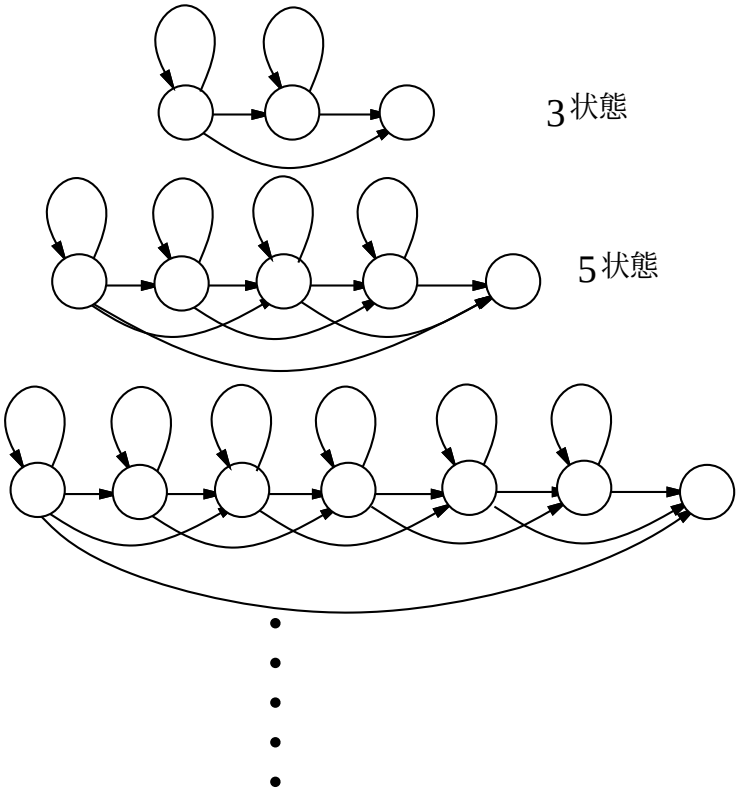


図 4.3: 比較に用いたモデル

これを見る限り，提案するアルゴリズムを用いた場合と，全てを 10 状態で定義した場合は認識システムとしてはほぼ同じ振舞いを示している．

## 4.5 考察

表 4.1 では，それぞれのモデルのパラメータ数が示されている．それぞれの間で差があるのは，そのサンプルの統計的な性質にももちろんよるが，データの量が異なっているからでもある．/k/を始め，/d/以外の子音はデータが多い分，ある程度パラメータが多くても推定ができる．これは AIC を用いていることから生じる．AIC3.7 では尤度の項とパラメータ数からなる項があるが，尤度は各サンプルの対数尤度を加え合わせ，符合をかえたものになる．従ってサンプルの数が多くなれば尤度の項も当然小さくなる．このため，サンプル数の少ないものに関しては推定できるモデルはサ

ンプル数の多いものに比べるとそのパラメータ数が少ないことが予想できる．

4.2 を見る限り，提案するアルゴリズムを用いた場合と，全てを 10 状態で定義した場合は認識システムとしてはほぼ同じ振舞いを示している．しかしながら，10 状態の場合と，提案したアルゴリズムを用いた場合の差は，それぞれのモデルの AIC を見ると明らかである．

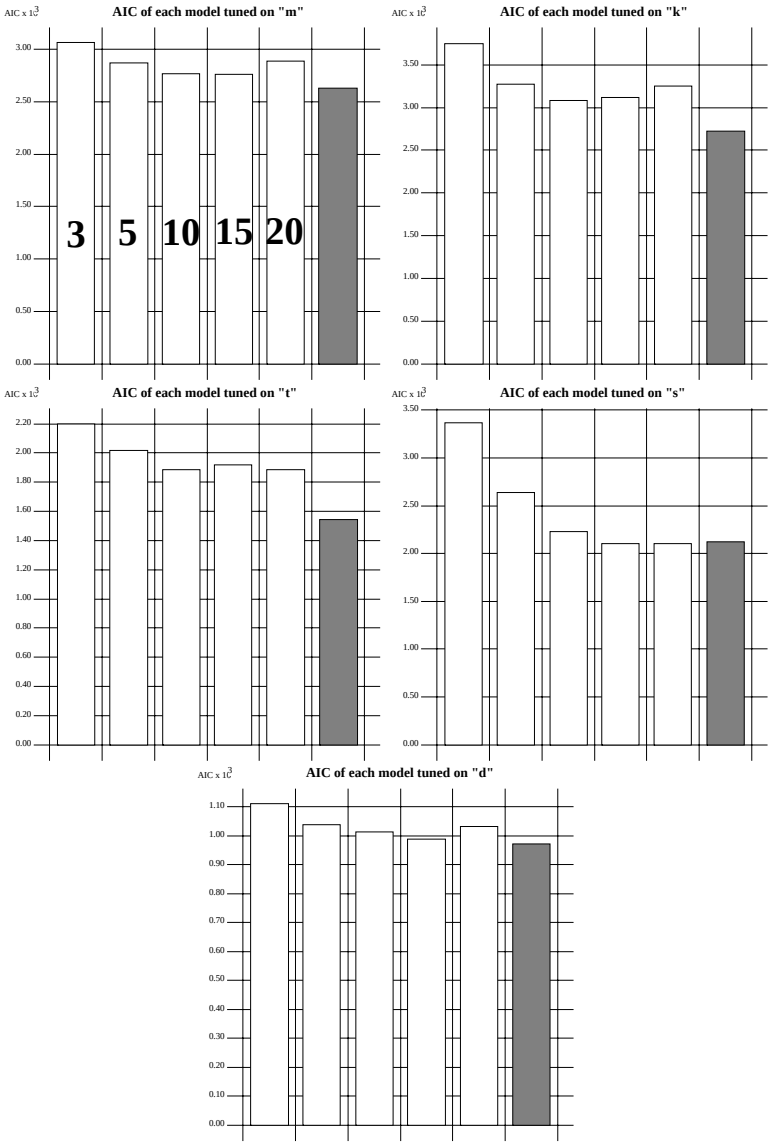


図 4.4: 各音素に対する各モデルの AIC

各グラフの右端にあるのがこのアルゴリズムを用いて求めたモデルの AIC である．

AIC を見る限りではほぼ最適なモデルを選択していることが分かる．

AIC についてもう一度考えてみる．AIC はモデルの分布と真の分布の距離を予測しようというものだった．これによってサンプルデータ (訓練用データ) へ過学習を押えることができる．サンプルデータに対する誤認識率と，認識用データに対する誤認識率を表 4.3 に示す．

表 4.3: 誤認識率

| 子音 (Sample 数) | 訓練用データ | 認識用データ |
|---|---|---|
| 各 3 状態 | 3.7% | 10.3% |
| 各 5 状態 | 2.0% | 9.2% |
| 各 10 状態 | 2.3% | 8.0% |
| 各 15 状態 | 1.4% | 11.2% |
| 各 20 状態 | 1.7% | 14.4% |
| 構造探索 | 2.0% | 8.3% |

状態数を増やしていけば，訓練用のデータに対しての学習は進み，訓練用のデータでの認識率は上がるが，認識用のデータでは誤認識率が高くなってしまう．これより，ある程度 AIC がうまく働き，問題に対する過学習を押えていることが分かる．

この実験では，10 状態で定義されたモデルが構造を探索した場合と同等の認識率をあげている．これは，前述したように，自分のカテゴリに対する尤度と他のモデルに対する尤度の間で，バランスがとれていることによる結果と考えられ，一般には 10 状態のモデルが良いとはいえない．このように各モデルを同じ状態数で定義するのではなく，それぞれに最適なモデルを探すことによって，全体の認識率をあげることができる．

# 第5章　結論

前に述べたように，モデル選択の立場からこのHMMに関する研究を行なってきた．まだまだ問題は残るものの，音声認識への応用として，一応の結果は得られたと思う．モデルを探索する場合，探索しなければならない空間は広く，本当に良いモデルを探索するのは難しい．この点，音声認識でのHMMを用いたのには理由がある．この場合のHMMは，L-Rモデルである．これは前にも書いたように定常的な確率過程を確率的に切替えるもので，この視点から見てDPマッチング (Dynamic Programing) との類似性が指摘されている[34]．したがって，単純には時間方向に状態数を増やしていくことでより良いモデルを選ぶことができると予想できる．

一方このHMMを用いたことには問題もある．モデルを表す空間が，現在考えている限りうまく定義できない．定常確率分布のあるHMMでは，その空間が対数線形の空間において定義できる[21]．このようにモデルがどの空間に存在し，その空間において，どの方向に探索すれば良いかを求めることがわかれば，このような研究にもさらに発展が期待できると考えている．

これからの課題としては，音声認識への応用として，より多くの音素での認識実験を行なうとともに，不特定話者での実験も行ないたい．また，モデル探索の問題は，音声に限るものではない．モデルはどのような空間で表現すればよいか，その中で，どのような基準をもとにモデルの構造を変化させていけば良いのか，このような視点からモデル探索を深く研究していきたいと考えている．

# 謝辞

何よりも研究する場を与えてくれた中野助教授に感謝します．それから，助手の阪口さん，さらに下平さんには有益な助言を頂きました．有難うございます．中野研の皆さんや今年一緒に卒業する修士の皆様にも感謝しています．どうも有難う．

# 関連図書


[1] K.-F. Lee: “Automatic Speech Recognition — The Development of the SPHINX System”, Kluwer Academic Publishers, Norwell, Massachusetts (1989).

[2] J. K. Baker: “The DRAGOM system—an overview”, IEEE Trans. Acoust., Speech & Signal Process., **ASSP-23**, 1, pp. 24–29 (1975).

[3] 平岡：“ビデオ予約用音声認識リモコンの認識評価”, 技術研究報告, 電子情報通信学会誌 (1991).

[4] 有本：“確率・情報・エントロピー”, 森北出版, 東京 (1980).

[5] 北川：“マルコフ過程”, 情報科学講座 A・5・1, 共立出版 (1977).

[6] 中川：“確率モデルによる音声認識”, 電子情報通信学会 (1988).

[7] 伊藤：“情報源の幾何学的構造の研究”, 修士論文, 東京大学, 計数工学科 (1988).

[8] B. H. Juang and L. Rabiner: “A probabilistic distance measure for hidden markov models”, The Bell System Technical Journal, **64**, 2, pp. 391–408 (1985).

[9] N. Merhav: “Universal classification for hidden markov models”, IEEE Trans. Inform. Theory, **37**, 6, pp. 1586–1594 (1991).

[10] 嵯峨山：“数理モデルによる音声認識の現状と将来”, 日本音響学会誌, **48**, 1, pp. 26–32 (1992).

[11] S. Levinson, L. Rabiner and M. Sondhi: “An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition”, The Bell System Technical Journal, **62**, 4, pp. 1035–1074 (1983).

[12] L. Rabiner and B. H. Juang: “An introduction to hidden markov models”, IEEE ASSP Magazine, pp. 4–16 (1986).

[13] A. P. Dempster, N. M. Laird and D. B. Rubin: “Maximum likelihood from incomplete data via the EM algorithm”, J. R. Statistical Society, Series B, **39**, pp. 1–38 (1977).

[14] L. E. Baum, T. Petrie, G. Soules and N. Weiss: “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains”, The Annals of Mathematical Statistics, **41**, 1, pp. 164–171 (1970).

[15] M. A. Tanner: “Tools for Statistical Inference”, Vol. 67 of Lecture Notes in Statistics, Springer-Verlag, Berlin (1991).

[16] K.-F. Lee, H.-W. Hon and R. Reddy: “An overview of the SPHINX speech recognition system”, IEEE Trans. Acoust., Speech & Signal Process., **ASSP-38**, 1, pp. 35–45 (1990).

[17] 赤池：“統計的検定の新しい考え方”, 数理科学, 198, pp. 51–57 (1979).

[18] 赤池：“モデルによってデータを測る”, 数理科学, 213, pp. 7–10 (1981).

[19] 村田：“学習の統計的漸近理論”, 博士論文, 東京大学, 計数工学科 (1991).

[20] 竹内：“AIC 基準による統計的モデル選択をめぐって”, 計測と制御, **22**, 5, pp. 445–453 (1983).

[21] 下平：“確率モデルに基づく認識と学習—-双対座標を用いた幾何学的アプローチ—”, 修士論文, 東京大学, 計数工学科 (1991).

[22] 鷹見, 嵯峨山：“音素コンテキストと時間に関する逐次状態分割による隠れマルコフモデル網の自動生成”, 技術研究報告, 電子情報通信学会誌 (1991).

[23] Y. Ephraim, A. Dembo and L. R. Rabiner: “A minimum discrimination information approach for hidden markov models”, IEEE Trans. Inform. Theory, **35**, 5, pp. 1001–1013 (1989).

[24] 赤池：“情報量規準 AIC とは何か–その意味と将来への展望”, 数理科学, 153, pp. 5–11 (1976).

[25] 武田, 匂坂, 片桐, 阿部, 桑原：“研究用日本語音声データベース利用解説書”, (株)ATR 自動翻訳電話研究所 (1986).

[26] 武田, 匂坂, 片桐, 桑原：“研究用日本語音声データベースの構築”, 日本音響学会誌, **44**, 10, pp. 747–754 (1988).

[27] 匂坂, 浦谷：“ATR 音声・言語データベース”, 日本音響学会誌, **48**, 12, pp. 878–882 (1992).

[28] 古井：“ディジタル音声処理”, 東海大学出版会 (1985).

[29] 三浦：“聴覚と音声”, 電子通信学会 (1980).

[30] A. V. Oppenheim and D. H. Johnson: “Discrete representation of signals”, Proc. IEEE, **60**, 6, pp. 681–691 (1972).

[31] 徳田, 小林, 今井：“メル一般化ケプストラムの再帰的計算法”, 電子情報通信学会論文誌 A, **J71-A**, 1, pp. 128–131 (1988).

[32] 小林, 今井：“一般化対数目盛における音声スペクトルの平滑化法”, 電子通信学会論文誌, **J64-A**, 6, pp. 473–474 (1981).

[33] Y. Linde, A. Buzo and R. M. Gray: “An algorithm for vector quantizer design”, IEEE Tr. Communications, **COM-28**, 1, pp. 84–95 (1980).

[34] B.-H. Juang: “On the hidden markov model and dynamic time warping for speech recogniton — a unified view”, The Bell System Technical Journal, **63**, 7, pp. 1213–1243 (1984).