解　説　論　文

Baum-Welch アルゴリズムの動作と応用例

Step by Step the Baum-Welch Algorithm and its Application

村上仁一 Jin'ichi MURAKAMI

アブストラクト

現在，音声認識や統計翻訳などの多くの分野では，モデルのパラメータ推定に EM アルゴリズムがよく使われている．EM アルゴリズムは，確率モデルのパラメータを最尤法に基づいて推定する反復法の一種である．初期値を与えて，期待値ステップと最大化ステップを交互に繰り返す．音声認識においては EM アルゴリズムの 1 つとして，Baum-Welch アルゴリズムがよく利用される．本論文では，この Baum-Welch アルゴリズムのステップバイステップの動作例を示して，このアルゴリズムの動作を説明する．最後に Baum-Welch アルゴリズムの他の分野への応用を紹介する．

キーワード　Baum-Welch アルゴリズム, EM 推定

1. はじめに

現在，音声認識や統計翻訳などの多くの分野で EM アルゴリズムがよく使われている．EM アルゴリズム (Expectation-maximization algorithm)[1]は，確率モデルのパラメータを最尤法に基づいて推定する統計的手法のひとつである． EM アルゴリズムは反復法の一種であり，期待値ステップと最大化ステップを交互に繰り返すことで計算が進行する． 期待値ステップでは，現在推定されている潜在変数の分布に基づいて，モデルの対数尤度の期待値を計算する．最大化ステップでは，期待値ステップで求まった対数尤度の期待値を最大化するようなパラメータを求める．最大化ステップで求まったパラメータは，次の期待値ステップで使われる潜在変数の分布を決定するために用いられる．

音声認識においては EM アルゴリズムの 1 つとして，Baum-Welch アルゴリズム[2]がよく利用される．数学的なアルゴリズムは，X.D.Huang ら，もしくは Steve Young らによって詳細に記述されている[3][4]．また Steve Young らにおいて Baum-Welch アルゴリズムのソースが公開されている[4]．このソースを利用することで，実際の動きが理解できる．

しかし，これら[3][4]に記述されているアルゴリズムは，数式で表現されているため，音声認識の研究を始めたばかりの初心者にとって動作がわかりにくい．そこでこの解説論文では，音声認識の研究／開発の初心者や統計的推定法の基本アルゴリズムを改めて理解したい方を対象に，Baum-Welch アルゴリズムの動作を直感的に理解できるように，簡単な具体例を使ってステップごとに解説する．

本論文の全体の構成を以下に示す． 2. 章においては，HMM（隠れマルコフモデル Hidden Markov Model）のパラメータについて説明する．3. 章においては，HMM のパラメータが与えられたときの入力シンボル系列の尤度の計算方法と，音声認識の方法について述べる． 4. 章では，認識のための HMM のパラメータに説明する．5. 章では，HMM のパラメータの学習方法である Baum-Welch アルゴリズムについて説明する． 最後に 6. 章では，Baum-Welch アルゴリズムの，言語モデルへの適応例や，他の分野への応用例を紹介する．そして，このアルゴリズムの拡張について示すとともに，問題点についても述べる．

2. HMM

2.1 確率付き有限状態オートマトンと HMM

有限状態オートマトンは，オートマトン理論や計算理論で研究されるオートマトンのなかで最も基本的なオートマトンである．有限状態オートマトンは状態遷移図によって表現され，これが直感的で理解しやすい．有限状態オートマトンの例を図 1 に示す．

単純な有限状態オートマトンでは，状態が遷移するときに，1 つのシンボルを出力する．図 1 では，a, b, c がシンボルである．

しかし，複数のシンボルを出力することもできる．複数のシンボルを出力するときに，それぞれのシンボルの出力に確率を付けたのが，確率付き有限状態オートマトンである．また，状態遷移が複数あり遷移先が一意に決定しない非決定性有限状態オートマトンに，遷移の確率を付与したモデルでもある．

村上仁一　正員　鳥取大学大学院工学研究科情報エレクトロニクス専攻
E-mail　murakami@ike.tottori-u.ac.jp
Jin'ichi Murakami, Member (Department of Information and Electronics, Graduate School of Engineering Tottori University 4-101, Koyamachou Minami, Tottori, 680-8552 Japan).

電子情報通信学会　基礎・境界ソサイエティ
Fundamentals Review Vol.xx No.xx pp.1–9 xxxx 年 xx 月

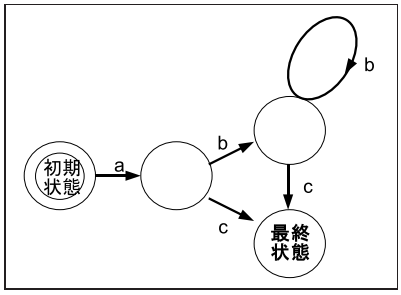


図 1　有限状態オートマトンの例

HMM は，確率付き有限状態オートマトンと同じパラメータを持つ．しかし，確率付き有限状態オートマトンは，任意の時刻に，どこの状態にいるのか明確に定義できるのに対し，HMM は，任意の時刻に，どの状態にいるのか定義できない．つまり，任意の時刻において，それぞれの状態に存在する確率で表現される．

2.2　Ergodic HMM と Left-to-Right HMM

HMM には，大きく分けて Left-to-Right HMM と Ergodic HMM が存在する．Left-to-Right HMM は，必ず状態を左から右に遷移するため，次の状態に遷移すると前の状態には戻ることができない． Ergoic HMM は，次の状態に遷移しても前の状態には戻ることができる場合があるモデルである．図 2 に代表的な Left-to-Right HMM の形を示す．この Left-to-Right HMM は，状態数が 4 つ，自己ループが 3 つあるため 4 状態 3 ループの Left-to-Right HMM と呼ばれる．また，図 3 に代表的な Ergodic HMM の形を示す．

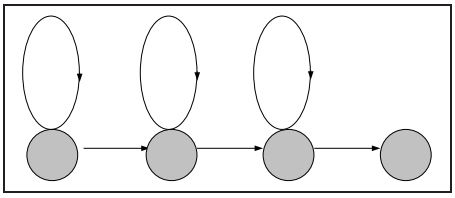


図 2　Left-to-Right HMM

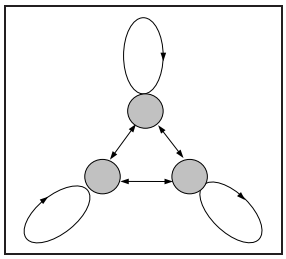


図 3　Ergodic HMM

なお，Left-to-Right HMM は，集合平均と時間平均が一致しない (エルゴード性が成立しない)． 一方 Ergodic HMM は，集合平均と時間平均が一致する (エルゴード性が成立する)．音声信号では，エルゴート性がないと仮定されるため，音声認識では，Left-to-Right HMM が利用される[5][6]．しかし，言語モデルでは，エルゴート性を考慮して，Ergodic-HMM も利用される[7]．

2.3　HMM のパラメータ

以後の説明では，音声認識を想定して Left-to-Right HMM を用いる， HMM のパラメータは，以下の 4 種類から定義されている．

- 初期状態確率 π_i

初期状態確率 π_i は，状態 i における初期状態確率を意味する．音声認識で利用される Left-to-Right HMM では，最初の状態は $i = 0$ のみで始まるため，$\pi_0 = 1.0$ になる．

- 最終状態

HMM の最後の状態を示す． 音声認識で利用される Left-to-Right HMM の最終状態は，だた 1 つである．

- 状態遷移確率 $a_{i,j}$

状態遷移確率 $a_{i,j}$ は，状態 i から状態 j に遷移する確率を意味する．

- シンボル出力確率 $b_{i,j}(O)$

シンボル出力確率は $b_{i,j}(O)$ と表記され，状態 i から状態 j に遷移したときに，シンボル O を出力する確率である．しかし本論文では，parameter tie を取ることで $b_i(O)$ と表記している．parameter tie に関しては，次の 2.4 節を参照のこと．

2.4　シンボル出力確率における parameter tie $b_{i,j}(O) = b_{i,i}(O) = b_i(O)$

HMM は，状態が遷移したときに，シンボルが出力される．したがって，状態 i から状態 j の遷移においてシンボル O が出力する確率を $b_{i,j}(O)$ と表現する．

しかし，本論文では，状態 i から状態 j の遷移におけるシンボル出力確率 $b_{i,j}(O)$ と状態 i から状態 i の遷移におけるシンボル出力確率 $b_{i,i}(O)$ は，同じシンボル出力確率をとると仮定する．これを parameter tie と呼ぶ．つまり，$b_{i,j}(O) = b_{i,i}(O) = b_i(O)$ とする．

音声信号において，短い時間間隔で区切った音声をフレームと呼ぶ．実際の音声では，フレームの前後は類似している．特に母音では，同じフレームが繰り返される．そのため，実際の音声認識では，自己ループの確率が，他の状態への遷移確率より高くなる．そのため，音声認識では，自己ループのときのシンボル出力確率 $b_{i,i}(O)$ は，信頼性が高い．しかし，他の状態への遷移確率は低いため，他の状態への遷移のときのシンボル出力確率 $b_{i,j}(O)$ は，信頼性が低い．そのため音声認識では，信頼性が低くなるパラメータを削除するために，このような parameter tie（$b_{i,j}(O) = b_{i,i}(O)$）をとる場合が多い．

2.5　HMM のパラメータの制約

HMM のパラメータは，確率付き有限状態オートマトンと同じパラメータを持つ．そして，パラメータには，以下の制約がある．

- 初期状態確率の制約

存在するすべての初期状態の確率の総和は 1.0 になる．

$$\sum_i \pi_i = 1.0$$

- 状態遷移確率の制約

ある状態 i から遷移する可能性のあるすべての状態 j への，状態遷移確率の総和は 1.0 になる．

$$\sum_j a_{i,j} = 1.0$$

- シンボル出力確率の制約

ある状態 i から遷移するときに出力される可能性のある全てのシンボルの，シンボル出力確率の総和は，1.0 になる．

$$\sum_{O \in K} b_i(O) = 1.0$$

なお，上式における K は．出力シンボル O の集合を表す．

2.6 HMM のパラメータの例

以下に， HMM のパラメータの例を 2 つ示す． “Model A” のパラメータを表 1 に， ”Model B” のパラメータを表 2 に示す．

これらのモデルは，以下の様に定義されている．

- 4 状態 3 ループの Left-to-Right HMM
  ($i = \{0, 1, 2\}, j = \{0, 1, 2, 3\}$)
- 出力シンボルは α, β, γ の 3 種類
  ($K = \{\alpha, \beta, \gamma\}$)
- 初期状態は，π_0 のみ
  (初期状態は $i = 0$, 最終状態は $i = 3$)

（1） Model A

表 1 Model A のパラメータ

初期状態確率		
$\pi_0 = 1.0$	$\pi_1 = 0.0$	$\pi_2 = 0.0$
最終状態		
3		
状態遷移確率		
$a_{0,0} = 0.7$	$a_{1,1} = 0.6$	$a_{2,2} = 0.1$
$a_{0,1} = 0.3$	$a_{1,2} = 0.4$	$a_{2,3} = 0.9$
シンボル出力確率		
$b_0(\alpha) = 0.7$	$b_1(\alpha) = 0.4$	$b_2(\alpha) = 0.1$
$b_0(\beta) = 0.2$	$b_1(\beta) = 0.3$	$b_2(\beta) = 0.1$
$b_0(\gamma) = 0.1$	$b_1(\gamma) = 0.3$	$b_2(\gamma) = 0.8$

図 4 に表 1 で示した HMM のパラメータを図示する．図で示したほうが理解しやすい．

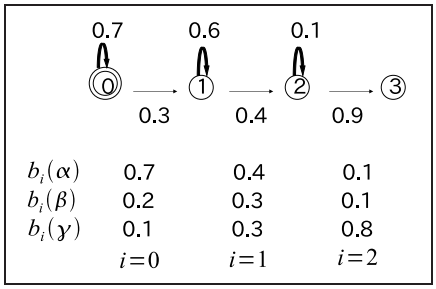


図 4 Model A

（2） Model B

表 2 に Model B のパラメータ例を示す．

表 2 Model B のパラメータ

初期状態確率		
$\pi_0 = 1.0$	$\pi_1 = 0.0$	$\pi_2 = 0.0$
最終状態		
3		
状態遷移確率		
$a_{0,0} = 0.5$	$a_{1,1} = 0.8$	$a_{2,2} = 0.7$
$a_{0,1} = 0.5$	$a_{1,2} = 0.2$	$a_{2,3} = 0.3$
シンボル出力確率		
$b_0(\alpha) = 0.1$	$b_1(\alpha) = 0.4$	$b_2(\alpha) = 0.1$
$b_0(\beta) = 0.5$	$b_1(\beta) = 0.2$	$b_2(\beta) = 0.8$
$b_0(\gamma) = 0.4$	$b_1(\gamma) = 0.4$	$b_2(\gamma) = 0.1$

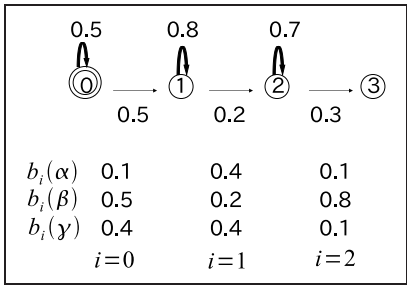


図 5 Model B

（3） HMM のパラメータの制約

2.5 節で示した制約により，これらの Model A および Model B の HMM のパラメータには，表 3 に示す条件を満たす必要がある．

表 3 HMM のパラメータの条件

$\pi_0 + \pi_1 + \pi_2 = 1.0$
$a_{0,0} + a_{0,1} = 1.0$
$a_{1,1} + a_{1,2} = 1.0$
$a_{2,2} + a_{2,3} = 1.0$
$b_0(\alpha) + b_0(\beta) + b_0(\gamma) = 1.0$
$b_1(\alpha) + b_1(\beta) + b_1(\gamma) = 1.0$
$b_2(\alpha) + b_2(\beta) + b_2(\gamma) = 1.0$

3. 尤　　度

3.1　入力シンボル系列

入力系列として，以下の入力シンボル系列を想定する．

$$\alpha \quad \alpha \quad \beta \quad \gamma$$

ここで O_t は時刻 t において観測されるシンボルとする．

$$O_0 = \alpha, O_1 = \alpha, O_2 = \beta, O_3 = \gamma$$

なお，音声認識においては，入力された音声を短時間のフレーム（40ms が多い）に分割し，それぞれのフレームを N 種類のシンボル（256 種類が多い）に変換して，入力シンボル系列としている．

3.2　尤度の計算

尤度とは，ある前提条件に従って結果が出現する場合に、逆に観察結果からみて前提条件が「何々であった」と推測する尤もらしさを表す値である． HMM における尤度とは，入力シンボル系列と HMM のパラメータが与えられたとき，取り得る全ての状態系列を考慮したときに，HMM が入力シンボル系列を出力する確率である． Model A において，入力シンボル系列の尤度の計算は，以下のように計算できる．

まず，入力シンボル系列を出力できる状態遷移を表 4 に載せる．全部で 3 種類ある．

表 4　入力シンボル系列を出力できる状態遷移

t	0	1	2	3
O_t	α	α	β	γ
path1:	0	0	1	2	3
path2:	0	1	1	2	3
path3:	0	1	2	2	3

それぞれの状態遷移における入力シンボル系列の確率を表 5 に示す．

表 5　各状態遷移の確率

path1:
$a_{0,0} \times b_0(O_0) \times a_{0,1} \times b_0(O_1) \times a_{1,2} \times b_1(O_2) \times a_{2,3} \times b_2(O_3)$
$= a_{0,0} \times b_0(\alpha) \times a_{0,1} \times b_0(\alpha) \times a_{1,2} \times b_1(\beta) \times a_{2,3} \times b_2(\gamma)$
$= 0.7 \times 0.7 \times 0.3 \times 0.7 \times 0.4 \times 0.3 \times 0.9 \times 0.8$
$= 0.00889$
path2:
$a_{0,1} \times b_0(O_0) \times a_{1,1} \times b_1(O_1) \times a_{1,2} \times b_1(O_2) \times a_{2,3} \times b_2(O_3)$
$= a_{0,1} \times b_0(\alpha) \times a_{1,1} \times b_1(\alpha) \times a_{1,2} \times b_1(\beta) \times a_{2,3} \times b_2(\gamma)$
$= 0.3 \times 0.7 \times 0.6 \times 0.4 \times 0.4 \times 0.3 \times 0.9 \times 0.8$
$= 0.00435$
path3:
$a_{0,1} \times b_0(O_0) \times a_{1,2} \times b_1(O_1) \times a_{2,2} \times b_2(O_2) \times a_{2,3} \times b_2(O_3)$
$= a_{0,1} \times b_0(\alpha) \times a_{1,2} \times b_1(\alpha) \times a_{2,2} \times b_2(\beta) \times a_{2,3} \times b_2(\gamma)$
$= 0.3 \times 0.7 \times 0.4 \times 0.4 \times 0.1 \times 0.1 \times 0.9 \times 0.8$
$= 0.000241$

尤度は，これらの状態遷移における確率を，すべて合計した値 $0.01349 (= 0.00889 + 0.00435 + 0.000241)$ である．

この例では，入力系列のシンボルの数が 4 つ $(O_0 = \alpha, O_1 = \alpha, O_2 = \beta, O_3 = \gamma)$ であるため，取り得る状態系列の数は 3 つである．しかし．入力系列のシンボルの数が多くなると，取り得る状態遷移の数は，急激に増加する．そして，取り得る状態遷移の数は，入力系列のシンボルの値に関係なく，シンボルの数だけで決まる．例えば，入力系列のシンボルの数が 6 つ (例えば $O_0 = \alpha, O_1 = \alpha, O_2 = \gamma, O_3 = \beta, O_4 = \beta, O_5 = \gamma$) になると，取り得る状態遷移の数は 10 になる．

3.3　Forward アルゴリズム

3.2 節の計算方法において，計算の多くが重複している．計算の重複を避けて尤度を計算する方法として，計算の途中を覚えておく動的計画法が利用できる．この計算方法を forward アルゴリズムと呼んでいる．このアルゴリズムは図による説明がわかりやすい．Model A のパラメータを利用したときのアルゴリズムの動きを図 6 で説明する．図の横軸は時系列，縦軸は状態，これらの格子点である黒丸は grid と呼ばれ、計算の途中を覚えておく場所であり，その値を，$\alpha_i(j)$ と表記する． $\alpha_i(j)$ は，時刻 0 から時刻 i まで状態 j において，入力シンボル系列を出力する確率の総和を意味する．また，O_t は時刻 t において観測されるシンボルである．横軸，斜め軸にある数値が，それぞれの状態遷移とシンボル出力確率の積であることに注意してほしい．計算順序は，入力シンボル系列が 1 つ入るごとに縦方向に計算を行う．尤度は，最終時刻の最終状態の grid の値になる．

forward アルゴリズムの grid の計算式を以下に示す．

$$\alpha_t(j) = \alpha_{t-1}(j) \times a_{j,j} \times b_j(O_{t-1}) + \alpha_{t-1}(j-1) \times a_{j-1,j} \times b_{j-1}(O_{t-1})$$

Model A のパラメータを利用したときの各 grid の計算式を以下に示す．

$\alpha_0(0) = 1.0$
$\alpha_1(0) = \alpha_0(0) \times a_{0,0} \times b_0(O_0) = \alpha_0(0) \times a_{0,0} \times b_0(\alpha)$
$= 1.0 \times 0.7 \times 0.7 = 0.49$
$\alpha_1(1) = \alpha_0(0) \times a_{0,1} \times b_0(O_0) = \alpha_0(0) \times a_{0,1} \times b_0(\alpha)$
$= 1.0 \times 0.3 \times 0.7 = 0.21$
$\alpha_2(1) = \alpha_1(1) \times a_{1,1} \times b_1(O_1) + \alpha_1(0) \times a_{01} \times b_0(O_1)$
$= \alpha_1(1) \times a_{1,1} \times b_1(\alpha) + \alpha_1(0) \times a_{01} \times b_0(\alpha)$
$= 0.21 \times 0.6 \times 0.4 + 0.49 \times 0.3 \times 0.7 = 0.1533$
$\alpha_2(2) = \alpha_1(1) \times a_{1,2} \times b_1(O_1) = \alpha_1(1) \times a_{1,2} \times b_1(\alpha)$
$= 0.21 \times 0.4 \times 0.4 = 0.0336$
$\alpha_3(2) = \alpha_2(2) \times a_{2,2} \times b_2(O_2) + \alpha_2(1) \times a_{1,2} \times b_1(O_2)$
$= \alpha_2(2) \times a_{2,2} \times b_2(\beta) + \alpha_2(1) \times a_{1,2} \times b_1(\beta)$
$= 0.0336 \times 0.1 \times 0.1 + 0.1533 \times 0.4 \times 0.3$
$= 0.018732 (\fallingdotseq 0.01873)$
$\alpha_4(3) = \alpha_3(2) \times a_{2,3} \times b_2(O_3) = \alpha_3(2) \times a_{2,3} \times b_2(\gamma)$
$= 0.018732 \times 0.9 \times 0.8 = 0.01348704 (\fallingdotseq 0.01349)$

$$\text{尤度} = \alpha_4(3) = 0.01348704$$

得られた尤度は，3.2 節で得られた尤度と等しい．

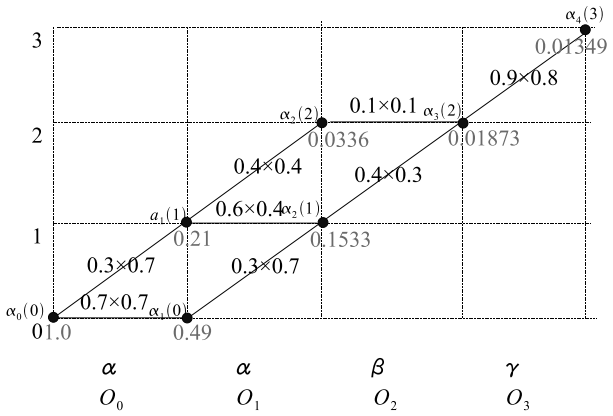


図 6　forward アルゴリズム

3.4　音 声 認 識

（1）　認識用の入力シンボル系列

入力系列として，以下の入力シンボル系列を想定する．

$\alpha \quad \alpha \quad \beta \quad \gamma$

ここで O_t は時刻 t において観測されるシンボルとする．$O_0 = \alpha, O_1 = \alpha, O_2 = \beta, O_3 = \gamma$

（2）　モデルの選択

Model A の HMM と Model B の HMM がある．この 2 つの HMM における尤度を計算する．

Model A の HMM の尤度は．0.01349

Model B の HMM の尤度は．0.00009

この場合 Model A の尤度は Model B の尤度より高いため Model A が選択される．つまり，Model A が入力の認識候補となる．

4.　HMM のパラメータの学習

3. 章においては，HMM のパラメータが与えられたときに，そのパラメータを用いた入力シンボル系列の尤度の計算方法について述べた．本 4. 章と次の 5. 章において，認識に用いるための HMM のパラメータの学習方法について説明する．

4.1　学習用の入力シンボル系列

入力系列として，以下の学習用の入力シンボル系列を想定する．

$\alpha \quad \alpha \quad \beta \quad \gamma$

ここで O_t は時刻 t において観測されるシンボルとする．

$O_0 = \alpha, O_1 = \alpha, O_2 = \beta, O_3 = \gamma$

HMM のパラメータは，学習用の入力シンボル系列に対して，尤度が最大となるように，決める必要がある．ただし，HMM のパラメータは 2.5 節で示した制約を満たす必要があるため表 3 に従う．

4.2　尤度が最大になる HMM のパラメータ

この例のように，入力シンボル系列のシンボル数が少ない場合に限定したとき，入力シンボル系列を最も尤もらしく表現する HMM のパラメータ，すなわち，入力シンボル系列に対して尤度が最大となるパラメータを，人間は推定することが可能である．おおよそのやり方は，シンボル出力確率をできるだけ 1.0 にする．そして，同じシンボルは同じ状態から出力する．

4.1 節の入力シンボル系列の尤度を，最大になる HMM のパラメータを表 6 および図 7 に示す．

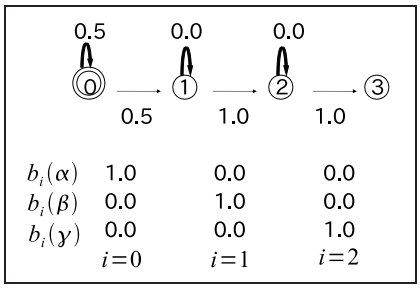


図 7　尤度が最大になるパラメータ

表 6　尤度が最大になる HMM のパラメータ

初期状態確率		
$\pi_0 = 1.0$	$\pi_1 = 0.0$	$\pi_2 = 0.0$
最終状態		
3		
状態遷移確率		
$a_{0,0} = 0.5$	$a_{1,1} = 0.0$	$a_{2,2} = 0.0$
$a_{0,1} = 0.5$	$a_{1,2} = 1.0$	$a_{2,3} = 1.0$
シンボル出力確率		
$b_0(\alpha) = 1.0$	$b_1(\alpha) = 0.0$	$b_2(\alpha) = 0.0$
$b_0(\beta) = 0.0$	$b_1(\beta) = 1.0$	$b_2(\beta) = 0.0$
$b_0(\gamma) = 0.0$	$b_1(\gamma) = 0.0$	$b_2(\gamma) = 1.0$

このときの forward の grid の値を図 8 に示す．

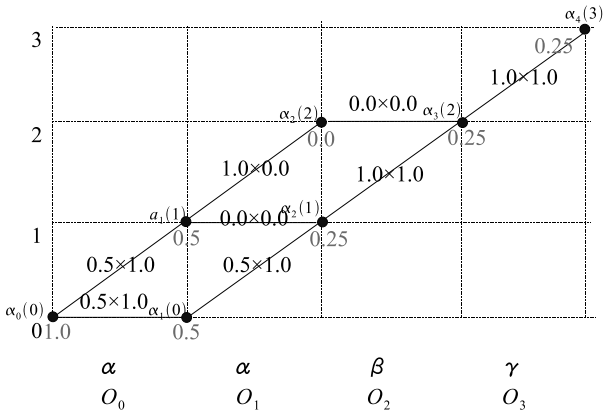


図 8　尤度が最大になるパラメータの尤度計算

尤度は，0.25 になり，これ以上高い値は，存在しない．

5. Baum-Welch アルゴリズム

4.2 節において，尤度が最大となるパラメータを示した．これを解析的に解く方法が Baum-Welch アルゴリズム[2]である．この学習アルゴリズムは山登り法の一種であり，繰り返し計算が基本になる．そのため最大値が保証できず，極大値になることもある．ただし経験的には Left-to-Right HMM では，起こりにくい．以下に計算例を示しながら解説する．

Baum-Welch アルゴリズムの計算のポイントは時刻 t における i から j への遷移確率 $\Gamma_t(i,j)$ の計算にある．この計算を行うためには forward アルゴリズムと backward アルゴリズムを利用する．なお， $\Gamma_t(i,j)$ の計算が期待値ステップになる．

5.1 Forward アルゴリズム

学習用の入力シンボル系列として，4.1 節に記載されている入力シンボル系列を利用する．また，初期値として，表 1 のパラメータを利用する．このときの forward アルゴリズムを図 9 に示す．なお，図 9 は，3.3 節の図 6 と同じである．また，forward アルゴリズムの grid の値を $\alpha_t(i)$ として表記する．

$$\alpha_t(j) = \alpha_{t-1}(j) \times a_{j,j} \times b_j(O_{t-1}) + \alpha_{t-1}(j-1) \times a_{j-1,j} \times b_{j-1}(O_{t-1})$$

$\alpha_0(0) = 1.0$　　$\alpha_1(0) = 0.49$
$\alpha_1(1) = 0.21$　　$\alpha_2(1) = 0.1533$
$\alpha_2(2) = 0.0336$　　$\alpha_3(2) = 0.01873$
$\alpha_4(3) = 0.01349$
尤度 $= \alpha_4(3) = 0.01349$

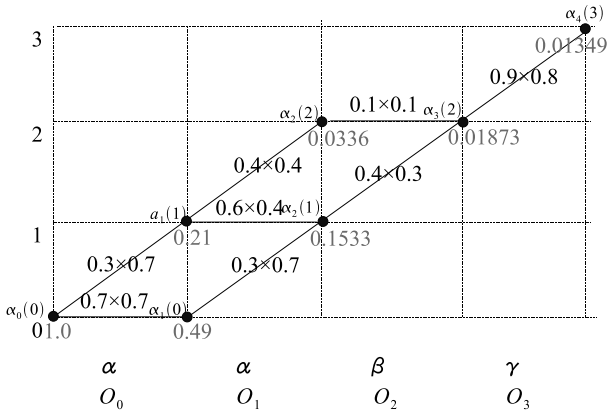


図 9　forward アルゴリズム

5.2 Backword アルゴリズム

Backward アルゴリズム は，入力シンボル系列において時系列の逆方向の尤度である．forward アルゴリズムと逆の方向から計算する．図 10 にこの例を示す．横軸は時系列で縦軸は状態を意味する．最終状態から，初期状態に向かって計算する．

backword アルゴリズムの grid の値は $\beta_t(i)$ として表記する．$\beta_t(i)$ は最後の状態から時刻 t まで状態 i における逆方向の尤度を意味する． forward アルゴリズムの grid の値 $\alpha_t(i)$ と逆方向である．

$\beta_4(3) = 1.0$　$\beta_4(2) = 0.0$　$\beta_4(1) = 0.0$　$\beta_4(0) = 0.0$ から始まる．

backward アルゴリズムの grid の計算式を以下に示す．

$$\beta_t(j) = \beta_{t+1}(j+1) \times a_{j,j+1} \times b_j(O_t) + \beta_{t+1}(j) \times a_{j,j} \times b_j(O_t)$$

各 grid の計算式を以下に示す．

$\beta_4(3) = 1.0$
$\beta_3(2) = \beta_4(3) \times a_{2,3} \times b_2(O_3) = \beta_4(3) \times a_{2,3} \times b_2(\gamma)$
$= 1.0 \times 0.9 \times 0.8 = 0.72$
$\beta_2(2) = \beta_3(2) \times a_{2,2} \times b_2(O_2) = \beta_3(2) \times a_{2,2} \times b_2(\beta)$
$= 0.72 \times 0.1 \times 0.1 = 0.0072$
$\beta_2(1) = \beta_3(2) \times a_{1,2} \times b_1(O_2) = \beta_3(2) \times a_{1,2} \times b_1(\beta)$
$= 0.72 \times 0.4 \times 0.3 = 0.0864$
$\beta_1(1) = \beta_2(2) \times a_{1,2} \times b_1(O_1) + \beta_2(1) \times a_{1,1} \times b_1(O_1)$
$= \beta_2(2) \times a_{1,2} \times b_1(\alpha) + \beta_2(1) \times a_{1,1} \times b_1(\alpha)$
$= 0.0072 \times 0.4 \times 0.4 + 0.0864 \times 0.6 \times 0.4$
$= 0.021888 (\fallingdotseq 0.02189)$
$\beta_1(0) = \beta_2(1) \times a_{0,1} \times b_0(O_1) = \beta_2(1) \times a_{0,1} \times b_0(\alpha)$
$= 0.0864 \times 0.3 \times 0.7 = 0.018144 (\fallingdotseq 0.01814)$
$\beta_0(0) = \beta_1(1) \times a_{0,1} \times b_0(O_0) + \beta_1(0) \times a_{0,0} \times b_0(O_0)$
$= \beta_1(1) \times a_{0,1} \times b_0(\alpha) + \beta_1(0) \times a_{0,0} \times b_0(\alpha)$
$= 0.021888 \times 0.3 \times 0.7 + 0.018144 \times 0.7 \times 0.7$
$= 0.01348704 (\fallingdotseq 0.01349)$

なお　Backword アルゴリズムにおいて最後に計算する grid $\beta_0(0)$ の値と Forward アルゴリズムにおいて最後に計算する grid $\alpha_4(3)$ の値は，同じになる．つまり，以下の式に着目してもらいたい．

尤度 $= \beta_0(0) = \alpha_4(3) = 0.01349$

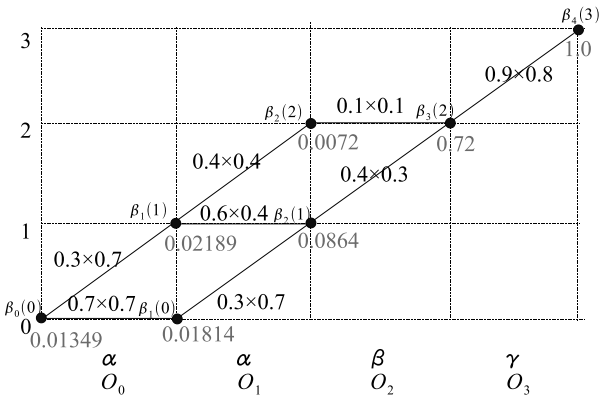


図 10　backward アルゴリズム

5.3 $\Gamma_t(i,j)$ の計算

Baum-Welch 学習のパラメータの再推定にはまず始めに $\Gamma_t(i,j)$ の計算をおこなう．これは，時刻 t において i から j に遷移する確率である．この計算が Baum-Welch アルゴリズムの基本になる．この値は， Forward と Backword の値を利用して計算する． $\Gamma_t(i,j)$ の計算は，期待値ステップになる．

$$\Gamma_t(i,j) = \frac{\alpha_t(i)a_{i,j}b_i(O_t)\beta_{t+1}(j)}{likelihood}$$

$$likelihood = 尤度 = \alpha_4(3)$$

各 $\Gamma_t(i,j)$ の計算式を以下に示す．

$$\Gamma_0(0,0) = \frac{\alpha_0(0)a_{0,0}b_0(O_0)\beta_1(0)}{likelihood} = \frac{\alpha_0(0)a_{0,0}b_0(\alpha)\beta_1(0)}{likelihood} = \frac{1.0\times0.7\times0.7\times0.018144}{0.01348704} = 0.65919$$

$$\Gamma_0(0,1) = \frac{\alpha_0(0)a_{0,1}b_0(O_0)\beta_1(1)}{likelihood} = \frac{\alpha_0(0)a_{0,1}b_0(\alpha)\beta_1(1)}{likelihood} = \frac{1.0\times0.3\times0.7\times0.021888}{0.01348704} = 0.34081$$

$$\Gamma_1(0,1) = \frac{\alpha_1(0)a_{0,1}b_0(O_1)\beta_2(1)}{likelihood} = \frac{\alpha_1(0)a_{0,1}b_0(\alpha)\beta_2(1)}{likelihood} = \frac{0.49\times0.3\times0.7\times0.0864}{0.01348704} = 0.65919$$

$$\Gamma_1(1,1) = \frac{\alpha_1(1)a_{1,1}b_1(O_1)\beta_2(1)}{likelihood} = \frac{\alpha_1(1)a_{1,1}b_1(\alpha)\beta_2(1)}{likelihood} = \frac{0.21\times0.6\times0.4\times0.0864}{0.01348704} = 0.32287$$

$$\Gamma_1(1,2) = \frac{\alpha_1(1)a_{1,2}b_1(O_1)\beta_2(2)}{likelihood} = \frac{\alpha_1(1)a_{1,2}b_1(\alpha)\beta_2(2)}{likelihood} = \frac{0.21\times0.4\times0.4\times0.0072}{0.01348704} = 0.01794$$

$$\Gamma_2(1,2) = \frac{\alpha_2(1)a_{1,2}b_1(O_2)\beta_3(2)}{likelihood} = \frac{\alpha_2(1)a_{1,2}b_1(\beta)\beta_3(2)}{likelihood} = \frac{0.1533\times0.4\times0.3\times0.72}{0.01348704} = 0.98206$$

$$\Gamma_2(2,2) = \frac{\alpha_2(2)a_{2,2}b_2(O_2)\beta_3(2)}{likelihood} = \frac{\alpha_2(2)a_{2,2}b_2(\beta)\beta_3(2)}{likelihood} = \frac{0.0336\times0.1\times0.1\times0.72}{0.01348704} = 0.01794$$

$$\Gamma_3(2,3) = \frac{\alpha_3(2)a_{2,3}b_2(O_3)\beta_4(3)}{likelihood} = \frac{\alpha_3(2)a_{2,3}b_2(\gamma)\beta_4(3)}{likelihood} = \frac{0.018732\times0.9\times0.8\times1.0}{0.01348704} = 1.0$$

なお，以下のように正規化されることに注意してもらいたい．

$\Gamma_0(0,0) + \Gamma_0(0,1) = 1.0$

$\Gamma_1(0,1) + \Gamma_1(1,1) + \Gamma_1(1,2) = 1.0$

$\Gamma_2(1,2) + \Gamma_2(2,2) = 1.0$

$\Gamma_3(2,3) = 1.0$

$\Gamma_t(i,j)$ は，時刻 t において状態 i から状態 j に遷移する確率である．そのためグリッドからグリッドの遷移において，$\Gamma_t(i,j)$ の値を示すと，Baum-Welch アルゴリズムは直感的に理解しやすい． 図 11 に $\Gamma_t(i,j)$ の値を示す．

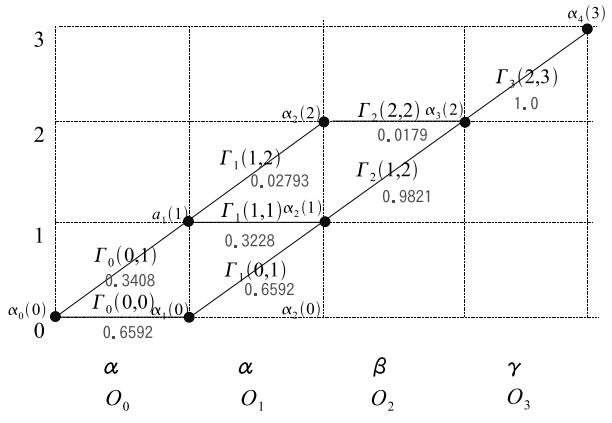


図 11 $\Gamma_t(i,j)$ の表現

5.4 状態遷移確率 $a_{i,j}$ の再計算

次に 状態遷移確率 $a_{i,j}$ を $\Gamma_t(i,j)$ から再計算する．この計算が最大化ステップになる．

$$a_{i,j} = \frac{\sum_t \Gamma_t(i,j)}{\sum_t \sum_j \Gamma_t(i,j)}$$

各 $a_{i,j}$ の計算式を以下に示す．

$$a_{0,0} = \frac{\Gamma_0(0,0)}{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)} = 0.39756$$

$$a_{0,1} = \frac{\Gamma_0(0,1)+\Gamma_1(0,1)}{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)} = 0.60244$$

$$a_{1,1} = \frac{\Gamma_1(1,1)}{\Gamma_1(1,1)+\Gamma_1(1,2)+\Gamma_2(1,2)} = 0.2441$$

$$a_{1,2} = \frac{\Gamma_1(1,2)+\Gamma_2(1,2)}{\Gamma_1(1,1)+\Gamma_1(1,2)+\Gamma_2(1,2)} = 0.7559$$

$$a_{2,2} = \frac{\Gamma_2(2,2)}{\Gamma_2(2,2)+\Gamma_3(2,3)} = 0.01762$$

$$a_{2,3} = \frac{\Gamma_3(2,3)}{\Gamma_2(2,2)+\Gamma_3(2,3)} = 0.98238$$

なお，以下のように正規化されることに注意してもらいたい．

$a_{0,0} + a_{0,1} = 1.0$

$a_{1,1} + a_{1,2} = 1.0$

$a_{2,2} + a_{2,3} = 1.0$

5.5 シンボル出力確率 $b_j(O)$ の再計算

最後にシンボル出力確率 $b_j(O)$ を $\Gamma_t(i,j)$ から再計算する．この計算も最大化ステップになる．

$$b_j(O) = \frac{\sum_{t\in O}\sum_k \Gamma_t(j,k)}{\sum_t \sum_k \Gamma_t(j,k)}$$

この式で $t \in O$ は，時刻 t におけるシンボル O_t が O であることを意味する．

各 $b_j(O)$ の計算式を以下に示す．

$$b_0(\alpha) = \frac{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)}{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)} = 1.0$$

$$b_0(\beta) = \frac{0}{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)} = 0.0$$

$$b_0(\gamma) = \frac{0}{\Gamma_0(0,0)+\Gamma_0(0,1)+\Gamma_1(0,1)} = 0.0$$

$$b_1(\alpha) = \frac{\Gamma_1(1,1)+\Gamma_1(1,2)}{\Gamma_1(1,1)+\Gamma_1(1,2)+\Gamma_2(1,2)} = 0.2576$$

$$b_1(\beta) = \frac{\Gamma_2(1,2)}{\Gamma_1(1,1)+\Gamma_1(1,2)+\Gamma_2(1,2)} = 0.7424$$

$$b_1(\gamma) = \frac{0}{\Gamma_1(1,1)+\Gamma_1(1,2)+\Gamma_2(1,2)} = 0.0$$

$$b_2(\alpha) = \frac{0}{\Gamma_2(2,2)+\Gamma_3(2,3)} = 0.0$$

$$b_2(\beta) = \frac{\Gamma_2(2,2)}{\Gamma_2(2,2)+\Gamma_3(2,3)} = 0.01762$$

$$b_2(\gamma) = \frac{\Gamma_3(2,3)}{\Gamma_2(2,2)+\Gamma_3(2,3)} = 0.98238$$

なお，以下のように正規化されることに注意してもらいたい．

$b_0(\alpha) + b_0(\beta) + b_0(\gamma) = 1.0$

$b_1(\alpha) + b_1(\beta) + b_1(\gamma) = 1.0$

$b_2(\alpha) + b_2(\beta) + b_2(\gamma) = 1.0$

5.6 尤度の再計算

尤度を再計算するため，forward アルゴリズムを再度実行する．この再計算の過程を図 12 に示す．

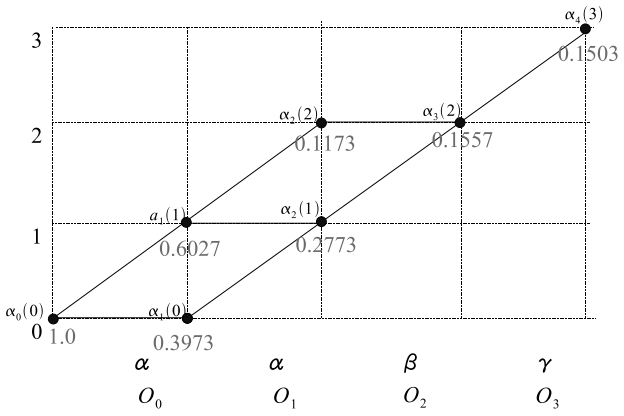


図 12　forward の再計算（尤度の再計算）

尤度 = 0.1503

この尤度 0.1503 は，5.1 節の尤度 0.01349 より，大きく向上していることがわかる．

5.7 繰り返し

Baum-Welch アルゴリズムは，以上の 5.1 節 から 5.5 節のステップを繰り返すことによって、尤度が最大となるパラメータを求める．この例では，２回目の繰り返し計算において，尤度 (likelihood) の値が，4.2 節の値 0.25 に近い値が得られる．終了条件は、尤度が向上しなくなったとき，もしくは事前に決めておいた繰り返し回数が終了したときなどが用いられる．

6. Baum-Welch アルゴリズムの他の応用

言語モデルにおいてネットワーク文法が用いられている．このネットワーク文法に確率を付けた確率付きネットワーク文法と，確率付き有限状態オートマトンは同じパラメータを持つ．そして HMM は確率付き有限状態オートマトンと同じパラメータを持つ．また，確率付きネットワーク文法は，次の状態に遷移しても前の状態に戻ることが必要であるため，Ergodic HMM に相当する[7]．

つまり，言語モデルにおいて利用される確率付きネットワーク文法は Ergodic HMM と同じパラメータを持つ．したがって，入力に単語系列を想定して，Ergodic HMM に対して Baum-Welch アルゴリズムを使うことで，自動的に確率付きネットワーク文法を獲得することが可能である．図 13 に，この例を示す．

このとき，状態遷移は品詞と見なすことができる．つまり単語の自動的に言語のクラスタリングが可能である．また，同じ状態遷移における単語の出現確率から，単語の距離を計算することもできる．この場合，シンボルが単語の種類の数になるため，大量のパラメータになる．そのため，大量の記憶量や計算量が必要になる．しかし，小さいシンボル出力確率や状態遷移確率を 0 と見なして Baum-Welch アルゴリズムを実行することが可能である．このテクニックを利用することで，記憶量や計算量を大幅に削減できる[8]．ほかにも DNA におけるタンパク質導出過程などの計算[9]にも利用されている．

Baum-Welch アルゴリズムは，応用範囲の広いアルゴリズムである．しかし，山登り法の 1 種であり，繰り返し計算によって解を求めるため．得られた解が最大値である保証はできない．そのため初期値が重要になる．特に Ergodic HMM では初期値によって得られるパラメータが大きく異なることに注意してもらいたい．

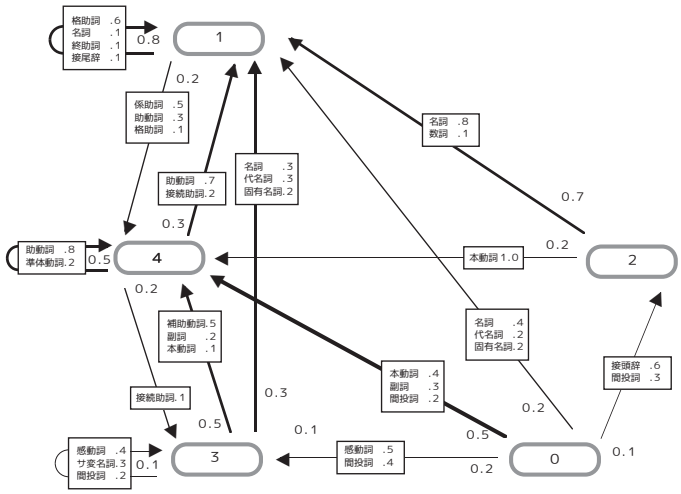


図 13　ネットワーク文法の自動獲得の例

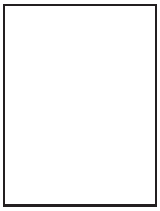
7. おわりに

本論文では，音声認識において使用される Baum-Welch アルゴリズムをステップバイステップで説明した．Baum-Welch アルゴリズムの基本は $\Gamma_t(i,j)$ の計算にある． $\Gamma_t(i,j)$ は，状態遷移確率 $a_{i,j}$ とシンボル出力確率 $b_j(O)$ から計算する．これが期待値ステップになる．状態遷移確率 $a_{i,j}$ は $\Gamma_t(i,j)$ から再計算する．シンボル出力確率 $b_j(O)$ は $\Gamma_t(i,j)$ から再計算する．これが最大化ステップになる．この計算をすることで，尤度が向上する．これを繰り返すことで，尤度は理想的な値に接近する．現実的には，尤度が向上しなくなったとき，繰り返し計算を止める．

HMM には，多くの応用例が考えられる． Baum-Welch アルゴリズムは，これらの応用例に適応できる．この解説論文が，これらの役に立つことを願っている．

文　献

(1) S. Rolf, “Maximum likelihood theory for incomplete data from an exponential family,” Scandinavian Journal of Statistics 1, pp.49-58, 1974.
(2) L.E.Baum and T.Petrie, “Statistical inference for probabilistic functtions of finite state Markov chains,” The Annals of Mathematical Statistics, vol.37, no.6, pp.1554-1563, 1966.
(3) X.D.Huang, Y. Ariki, M.A.Jack, “Hidden Markov Models For Speech Recognition,” 0-7486-0162-7, Edinburgh University Press, Edinburgh, 1990.
(4) S.J. Young, Sj Young, “The HTK Hidden Markov Model Toolkit: Design and Philosophy,” Entropic Cambridge Research Laboratory, vol 2, pp.2-44, 1994. http://htk.eng.cam.ac.uk/.
(5) Kai-Fu Lee, “Automatic Speech Recognition: The Development of the SPHINX Recognition System,” 978-0898382969, Springer, New York, 1988.
(6) F. Jerinek, “Statistical Methods for Speech Recognition”, 0-262-10066-5, The MIT Press Cambride, Massachusetts London, Massachusetts, 1997.
(7) 村上 仁一, 山本 寛樹, 嵯峨山 茂樹, “Ergodic HMM による確率つきネットワーク文法の獲得の可能性について,” 人工知能学会, 研究会資料，SIG-SLUD-9204-3, pp.17-24, 1993.
(8) 村上 仁一, “メモリ量および計算量を削減した Baum-Welch アルゴリズムの提案と言語モデルへの適用,” 信学技報, SP94-97, pp.45-50, 1995.
(9) C.Burge,S. Karlin, “Prediction of complete gene structures in human genomic DNA,” J. Mol. Biol. 268, pp.78-94, 1997.



村上仁一　（正員）

1984 年筑波大学第 3 学群基礎工学類卒．1986 年筑波大学大学院修士課程理工学研究科理工学専攻修了．同年 NTT に入社．NTT 情報通信処理研究所に勤務．1991 年国際通信基礎研究所 (ATR) 自動翻訳電話研究所に出向．1995 年 NTT 情報通信網研究所に復帰．1998 年鳥取大学工学部知能情報工学科に転職．現在に至る．主に音声認識のための言語処理の研究を行う．最近は統計翻訳の研究に従事．電子情報通信学会，日本音響学会，言語処理学会各会員．