数理解析研究所講究録
第1682巻 2010年 193-200

HMMの学習アルゴリズムの並列化に関する一考察

広島大学大学院工学研究科　河合　理恵，岡村　寛之，土肥　正
Rie Kawai, Hiroyuki Okamura and Tadashi Dohi
Graduate School of Engineering
Hiroshima University, Japan

1　はじめに

HMM（Hidden Markov Model; 隠れマルコフモデル）は時系列データを解析する統計モデルの一つであり，パラメータ未知のマルコフ過程を仮定している．確率によって動作する非決定有限オートマトンであり，観測データから内部状態の推移を一意に特定できない．そのために隠れマルコフモデルと呼ばれている．HMMでは観測された時系列データからモデル内部のパラメータを推定する．連続的かつ伸縮しうる信号列のパターン抽出などに適しており，音声認識をはじめとしてDNA配列設計，画像認識にも利用されている [1, 2, 3].

HMMの学習は，観測データから内部のパラメータを推定する作業に対応づけられる．代表的な学習アルゴリズムに，Baumらが提案したBaum-Welchアルゴリズム（BW）がある [4]. BWはEM法（Expectation-Maximization method）を用いてパラメータの最尤推定値を求める学習アルゴリズムであるが，計算量が観測データの大きさに比例して増加する．そのため，長い系列データを扱う場合などに膨大な学習時間を要する場合がある．

一般に数値計算の計算時間短縮は，アルゴリズムの改良による理論的な時間計算量の減少と，コーディングによる工夫やハードウェア化によって処理効率を上げる手法 [5] がある．コーディングによる計算時間の短縮は，BLAS (Basic Linear Algebra Subprograms) などのようなアーキテクチャに最適化されたライブラリを利用することで簡単に計算速度を向上することができるため，本質的なアルゴリズムの改良なしで手軽に利用できる．

一方，近年のCPUにおけるコア数の増加とOpenMPなどのメモリ共有型の並列化ライブラリの整備は「低コストの並列化」と言うこれまでとは異なる計算時間短縮の潮流を生み出している．メモリ共有型の並列化は手軽に利用できるが，既存のアルゴリズムを単純に並列化するだけでは十分な効果を得ることができないことが多い．そこで本稿では，HMMの学習アルゴリズムに対して，一般化EM法（Generalized EM; GEM）[6] と呼ばれる手法を適用する．

本稿では，第2章はHMMの定義について説明し，第3章ではEM法から導出されるBaum-Welchアルゴリズム（BW）を紹介する．第4章ではGEMの原理と，GEMを用いたHMMの学習アルゴリズムについて説明する．第5章では，2つのアルゴリズムを計算時間と学習の効率性の観点から比較する．最後に第6章では，今回得られた数値例から得た結論と今後の課題について述べる．

2　HMMの記述

本稿では離散型HMMについて考える．内部状態数を M，シンボル数を L とする．内部状態の集合を $\boldsymbol{S} = \{s_1, \cdots, s_M\}$，出力シンボルの集合を $\boldsymbol{v} = \{v_1, \cdots, v_L\}$ と表す．次に，HMMに対して，内部状態の推移確率行列 $\boldsymbol{A} = [a_{i,j}]$ を定義する．また，シンボル $v_l, l = 1, \cdots, L$ の出力に関する確率行列を $\boldsymbol{B}(v_l)$ をとする．シンボルの出力は内部状態に依存するため，対角要素に $b_i(v_l)$ を持つ対角行列として $\boldsymbol{B}(v_l)$ を仮定する．$b_i(v_l)$ は内部状態 s_i でシンボル v_l を出力する確率を表している．またHMMの初期状態確率を表現する確率ベクトルを $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_M\}$，吸収状態を表すベクトルを $\boldsymbol{\xi} = \{\xi_1, \cdots, \xi_M\}^T$ とする．このとき，π_i は内部状態 s_i を初期状態として選択する確率，ξ_i は内部状態 s_i が吸収状態であれば1，吸収状態でない場合は0である．

3　Baum-Welchアルゴリズム

3.1　EM法の概要

EM法は期待値最大化法とも呼ばれ，観測データからモデルパラメータの最尤推定値を算出するのに用いられる．一般的に観測データを $\boldsymbol{D}$，未観測データ（潜在変数）を $\boldsymbol{Z}$ とする．このとき，観測データに対する

対数尤度をモデルパラメータベクトル $\boldsymbol{\theta}$ を使って表現すると，

$$\log P(\boldsymbol{D};\boldsymbol{\theta}) = \log \sum_{\boldsymbol{Z}} P(\boldsymbol{D},\boldsymbol{Z};\boldsymbol{\theta}) \tag{1}$$

となる．ここで $P(\cdot)$ は任意の密度または確率関数である．原理的に最尤推定値はこの式を最大化する $\boldsymbol{\theta}$ を求めればよい．しかしながら，式 (1) を最大にする $\boldsymbol{\theta}$ を陽に得ることは困難である．そこで EM 法では完全データ $(\boldsymbol{D},\boldsymbol{Z})$ に対する対数尤度関数の期待値を最大にする $\boldsymbol{\theta}$ を求めることを考える．具体的に未観測データに対する条件付分布 $P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta})$ が得られたとすると，完全データに対する期待対数尤度は，

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}') = \sum_{\boldsymbol{Z}} P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta}') \log P(\boldsymbol{D},\boldsymbol{Z};\boldsymbol{\theta}) \tag{2}$$

となる．ここで $\boldsymbol{\theta}'$ は暫定的に与えられるパラメータベクトルである．式 (2) で表される完全データに対する期待対数尤度は Q 関数と呼ばれる．EM 法では暫定パラメータ $\boldsymbol{\theta}'$ が与えられたもとで Q 関数を算出するステップ（E ステップ）と，それを最大にする $\boldsymbol{\theta}$ を求め，新たなパラメータとして更新するステップ（M ステップ）を繰り返すことで最尤推定値を算出する．

3.2 EM 法を用いた HMM パラメータの推定（Baum-Welch アルゴリズム）

観測データとして，シンボル系列 $\boldsymbol{D} = \{x_1,\cdots,x_K\}$ が得られた時の HMM パラメータ推定を考える．未観測データ（潜在変数）として $\boldsymbol{Z} = \{\boldsymbol{z}_1,\cdots,\boldsymbol{z}_{K+1}\}$, $\boldsymbol{z}_k = (z_{k,1},\cdots,z_{k,M})$ を導入する．$z_{k,i}$ は時刻 k においてシンボル x_k を出力する直前の内部状態が i であるかどうかをあらわす指標であり，内部状態が i ならば 1，それ以外は 0 である．これを用いると完全データ $(\boldsymbol{D},\boldsymbol{Z})$ に対する対数尤度は，

$$\log P(\boldsymbol{D},\boldsymbol{Z}) = \sum_{i=1}^{M} z_{1,i}\log\pi_i + \sum_{k=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M} z_{k,i}z_{k+1,j}\log(I(x_k=v_l)b_i(v_l)a_{i,j}) + \sum_{i=1}^{M} z_{K+1,i}\log\xi_i \tag{3}$$

となる．$P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta}')$ による期待演算子を $E[\cdot|\boldsymbol{D};\boldsymbol{\theta}']$ とすると，式 (2) および (3) より，

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}') &= \sum_{i=1}^{M} E[z_{1,i}|\boldsymbol{D};\boldsymbol{\theta}']\log\pi_i + \sum_{k=1}^{K}\sum_{i=1}^{M}\sum_{j=1}^{M} E[z_{k,i}z_{k+1,j}|\boldsymbol{D};\boldsymbol{\theta}']\log(I(x_k=v_l)b_i(v_l)a_{i,j}) \\ &\quad + \sum_{i=1}^{M} E[z_{K+1,i}|\boldsymbol{D};\boldsymbol{\theta}']\log\xi_i \end{aligned} \tag{4}$$

を得る．ここで $\boldsymbol{\theta} = (\pi,\boldsymbol{A},\boldsymbol{B},\boldsymbol{\xi})$ である．このとき式 (4) を最大化するパラメータはラグランジュ未定乗数法から，

$$a_{i,j} = \frac{\sum_{k=1}^{K} E[z_{k,i}z_{k+1,j}|\boldsymbol{D};\boldsymbol{\theta}']}{\sum_{k=1}^{K}\sum_{j=1}^{M} E[z_{k,i}z_{k+1,j}|\boldsymbol{D};\boldsymbol{\theta}']},\quad b_i(v_l) = \frac{\sum_{k=1}^{K} I(z_k=v_l)E[z_{k,i}|\boldsymbol{D};\boldsymbol{\theta}']}{\sum_{k=1}^{K} E[z_{k,i}|\boldsymbol{D};\boldsymbol{\theta}']},\quad \pi_i = E[z_{1,i}|\boldsymbol{D};\boldsymbol{\theta}'] \tag{5}$$

と陽に得られる（M ステップにおける更新式）．一方，$E[z_{k,i}z_{k+1,j}|\boldsymbol{D};\boldsymbol{\theta}']$ は，

$$\begin{aligned} E[z_{k,i}z_{k+1,j}|\boldsymbol{D};\boldsymbol{\theta}] &= \sum_{\boldsymbol{Z}} z_{k,i}z_{k+1,j}P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta}) \\ &= \frac{1}{P(\boldsymbol{D})}[\pi\boldsymbol{B}(x_1)\boldsymbol{A}\boldsymbol{B}(x_2)\boldsymbol{A}\cdots\boldsymbol{B}(x_{k-1})\boldsymbol{A}]_i b_i(x_k)a_{i,j}[\boldsymbol{B}(x_{k+1})\boldsymbol{A}\cdots\boldsymbol{B}(x_K)\boldsymbol{A}\boldsymbol{\xi}]_j \end{aligned} \tag{6}$$

によって求めることができる．ここで $\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k$ を

$$\boldsymbol{\alpha}_k = \pi\boldsymbol{B}(x_k)\boldsymbol{A}\boldsymbol{B}(x_{k+1})\boldsymbol{A}\cdots\boldsymbol{B}(x_K)\boldsymbol{A},\quad \boldsymbol{\beta}_k = \boldsymbol{B}(x_k)\boldsymbol{A}\boldsymbol{B}(x_{k+1})\boldsymbol{A}\cdots\boldsymbol{B}(x_K)\boldsymbol{A}\boldsymbol{\xi} \tag{7}$$

とすると，$E[z_{k,i}|\boldsymbol{D}]$ と $E[z_{1,i}|\boldsymbol{D}]$ は，

$$E[z_{k,i}|\boldsymbol{D}] = \frac{1}{P(\boldsymbol{D})}[\boldsymbol{\alpha}_{k-1}]_i[\boldsymbol{B}(x_k)\boldsymbol{A}\boldsymbol{\beta}_{k+1}]_i, \tag{8}$$

$$E[z_{1,i}|\boldsymbol{D}] = \frac{\pi_i[\boldsymbol{\beta}_1]_i}{P(\boldsymbol{D})} \tag{9}$$

によって得ることができる．ここで新たな変数とその期待値を導入する，

$$E[N_{i,j}|\boldsymbol{D}] = \sum_{k=1}^{K} E[z_{k,i}z_{k+1,j}|\boldsymbol{D}],\ E[Y_{i,l}|\boldsymbol{D}] = \sum_{k=1}^{K} I(x_k = v_l)E[z_{k,i}|\boldsymbol{D}],\ E[R_i|\boldsymbol{D}] = E[z_{1,i}|\boldsymbol{D}]. \tag{10}$$

$E[N_{ij}|\boldsymbol{D}]$ は時刻 0 から時刻 K までの間に，内部状態 s_i が s_j へ推移した回数の期待値であり，分子を $EN_{i,j}$ で表す．$E[Y_{i,k}|\boldsymbol{D}]$ は時刻 0 から時刻 K までの間に，内部状態 s_i でシンボル v_l が出力された回数の期待値であり，分子を EY_{il} で表す．$E[R_i|\boldsymbol{D}]$ は HMM が内部状態 s_i で開始されるかどうかの期待値であり分子を ER_i で表す．実際の BW によるパラメータ推定では，これらの確率変数を用いて以下の 4 つのステップを繰り返すことで推定値を算出する．

Step1 時刻 0 および各時刻 $k = 1, 2, \cdots, K$ に対して前向き尤度 $\boldsymbol{\alpha}_k$ を求める．

$$\boldsymbol{\alpha}_0 = \boldsymbol{\pi},\ \boldsymbol{\alpha}_k = \boldsymbol{\alpha}_{k-1}\boldsymbol{B}(x_k)\boldsymbol{A}. \tag{11}$$

Step2 時刻 $K+1$ および各時刻 $k = K, K-1, \cdots, 1$ に対して後ろ向き尤度 $\boldsymbol{\beta}_k$ を求める．

$$\boldsymbol{\beta}_{K+1} = \boldsymbol{\xi},\ \boldsymbol{\beta}_k = \boldsymbol{B}(x_k)\boldsymbol{A}\boldsymbol{\beta}_{k+1}. \tag{12}$$

Step3 確率変数の期待値の分子を求める．

$$EN_{i,j} = \sum_{k=1}^{K}[\boldsymbol{\alpha}_{k-1}]_i b_i(x_k)a_{i,j}[\boldsymbol{\beta}_{k+1}]_j, \tag{13}$$

$$EY_{i,l} = \sum_{k=1}^{K} I(x_k = v_l)[\boldsymbol{\alpha}_{k-1}]_i[\boldsymbol{B}(v_l)\boldsymbol{A}\boldsymbol{\beta}_{k+1}]_i, \tag{14}$$

$$ER_i = \pi_i[\boldsymbol{\beta}_1]_i \tag{15}$$

Step4 パラメータを更新する．

$$a_{i,j} = \frac{EN_{i,j}}{\sum_{j=1}^{M} EN_{i,j}},\ b_i(l) = \frac{EY_{i,l}}{\sum_{l=1}^{L} EY_{i,l}},\ \pi_i = \frac{ER_i}{\sum_{i=1}^{M} ER_i}. \tag{16}$$

ここでは，各期待値の分母は Step4 における除算で約分されるので，算出していない．実際には暫定的なパラメータから始めて，Step1 から Step4 をパラメータまたは尤度が収束するまで繰り返す．

最後に BW の並列化について考える．並列化を行うためにはデータ依存の除去が重要な課題である．データ依存とはループ内での計算において，一つ前の反復の値が次の反復に影響を及ぼすような場合を指す．例えば Step1 の計算において $\boldsymbol{\alpha}_{k-1}$ を求めなければ，$\boldsymbol{\alpha}_k$ を求めることができない．この関係を除去することが並列計算には必要不可欠である．

図 1 は BW の並列化の概念図である．BW は前向き確率と後ろ向き確率の計算にデータ依存を含んでいるため Step1 と Step2 は逐次実行する必要があり，この部分では BW は高々2 並列での計算が限界である．Step3 は足し算の繰り返しとなっており，データ依存が存在しない．そのためスレッド数に応じた並列計算が可能となる．ステップ 4 ではステップ 3 の結果を用いて除算を行ってパラメータを更新する．

4 一般化 EM 法による並列学習アルゴリズム

4.1 一般化 EM 法の原理

一般化 EM 法（Genelaized EM,GEM）は EM 法における事後分布に対して近似を適用するものである．パラメータ $\boldsymbol{\psi}$ をもつ新たな分布 $\tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi})$ を導入し $P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta})$ の近似として用いる．このとき，観測データに

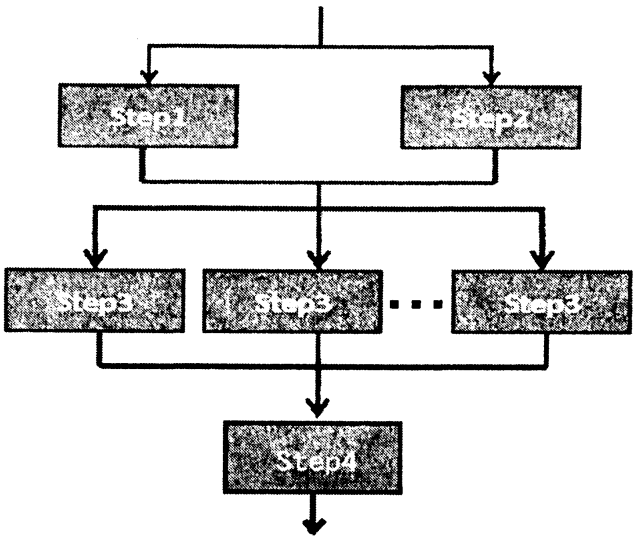

図 1: BW の並列化の概念図.

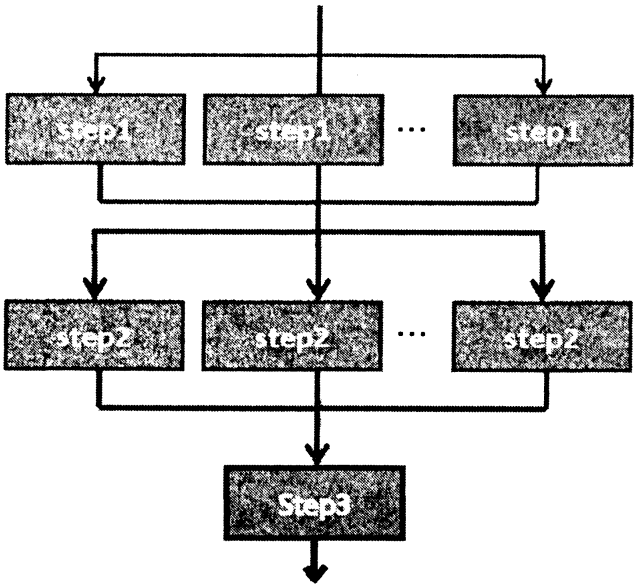

図 2: GEM の並列化の概念図.

対する対数尤度の下限値として以下の結果を得る.

$$\begin{aligned}\log P(\boldsymbol{D};\boldsymbol{\theta}) &= \log \sum_{\boldsymbol{Z}} \tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi}) \frac{P(\boldsymbol{D},\boldsymbol{Z};\boldsymbol{\theta})}{\tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi})} \\ &\geq \sum_{\boldsymbol{Z}} \tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi}) \log \frac{P(\boldsymbol{D},\boldsymbol{Z};\boldsymbol{\theta})}{\tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi})} \equiv F(\tilde{P},\boldsymbol{\theta}). \end{aligned} \tag{17}$$

単純に下限値 $F(\tilde{P},\boldsymbol{\theta})$ と対数尤度の差を考えると,

$$\log P(\boldsymbol{D},\boldsymbol{\theta}) - F(\tilde{P},\boldsymbol{\theta}) = KL(\tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi}), P(\boldsymbol{Z}|\boldsymbol{D};\boldsymbol{\theta})) \tag{18}$$

となる. $KL(\cdot,\cdot)$ は KL ダイバージェンスであり, 二つの分布の距離を表している. この式より $\tilde{P}(\boldsymbol{Z}|\boldsymbol{\psi})$ を最良の近似にするためには $F(\tilde{P},\boldsymbol{\theta})$ を最大化するような $\boldsymbol{\psi}$ を用いれば良いことが分かる.

4.2 一般化 EM 法を用いた HMM パラメータの推定

3.2 と同様にシンボル系列 $\boldsymbol{D} = \{x_1, x_2, \cdots, x_K\}$ が観測されたもとで HMM パラメータを GEM によって推定することを考える. いま, 未観測データを $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots \boldsymbol{z}_{K+1}\}$ とし, その事後分布の近似を $q(\boldsymbol{z})$ で表現する. つまり,

$$P(\boldsymbol{Z}|\boldsymbol{D}) \cong q(\boldsymbol{Z}). \tag{19}$$

また $q(\boldsymbol{z})$ に対して, 因数分解が可能である仮定 $q(\boldsymbol{Z}) = q_1(\boldsymbol{z}_1)q_2(\boldsymbol{z}_2)\cdots q_{K+1}(\boldsymbol{z}_{K+1})$ を設けるとき式 (17) の下限値は,

$$F(q_1,\cdots,q_{K+1};\boldsymbol{\theta}) = \sum_{\boldsymbol{z}_1}\sum_{\boldsymbol{z}_2}\cdots\sum_{\boldsymbol{z}_{K+1}} q_1(\boldsymbol{z}_1)q_2(\boldsymbol{z}_2)\cdots q_{K+1}(\boldsymbol{z}_{K+1}) \log \frac{P(\boldsymbol{D},\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_{K+1};\boldsymbol{\theta})}{q_1(\boldsymbol{z}_1)q_2(\boldsymbol{z}_2),\cdots,q_{K+1}(\boldsymbol{z}_{K+1})} \tag{20}$$

となる. 制約条件 $\sum_{\boldsymbol{z}_k} q_k(\boldsymbol{z}_k) = 1, k = 1,\cdots,K+1$ のもとでオイラーラグランジュ方程式を解くと,

$$\begin{aligned} q_k^*(\boldsymbol{z}_k|\boldsymbol{D}) \propto \exp\{&\sum_{\boldsymbol{z}_1}\cdots\sum_{\boldsymbol{z}_{k-1}}\sum_{\boldsymbol{z}_{k+1}}\cdots\sum_{\boldsymbol{z}_{K+1}} q_1^*(\boldsymbol{z}_1|\boldsymbol{D})\cdots q_{k-1}^*(\boldsymbol{z}_{k-1}|\boldsymbol{D})q_{k+1}^*(\boldsymbol{z}_{k+1}|\boldsymbol{D})\cdots q_{K+1}^*(\boldsymbol{z}_{K+1}|\boldsymbol{D}) \\ &\times \log P(\boldsymbol{D},\boldsymbol{z}_1,\cdots,\boldsymbol{z}_{K+1})\} \end{aligned} \tag{21}$$

を得る．$q_k^*(\boldsymbol{z}_k)$ は下限値を最大にする $q_k(\boldsymbol{z}_k)$ である．具体的に $q_k(\boldsymbol{z}_k)=\prod_{i=1}^{M}\psi_{k,i}^{z_{k,i}}$ とすると

$$\begin{aligned} q_k^*(\boldsymbol{z}_k) \propto \exp\{&\sum_{i=1}^{M}\psi_{1,i}\log\pi_i+\sum_{l=2}^{k-2}\sum_{i=1}\sum_{j=1}\psi_{l,i}\psi_{l+1,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}\\ &+\sum_{i=1}^{M}\sum_{j=1}^{M}\psi_{k-1,i}z_{k,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}+\sum_{i=1}^{M}\sum_{j=1}^{M}z_{k,i}\psi_{k+1,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}\\ &+\sum_{l=k+1}^{K}\sum_{i=1}\sum_{j=1}\psi_{l,i}\psi_{l+1,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}+\sum_{i=1}^{M}\psi_{K+1,i}\log\xi_i\}. \end{aligned} \tag{22}$$

式 (22) で $z_{k,i}$ を含む項に注目すると，

$$q_k^*(\boldsymbol{z}_k)\propto\exp\{\sum_{i=1}^{M}\sum_{j=1}^{M}\psi_{k-1,i}z_{k,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}+\sum_{i=1}^{M}\sum_{j=1}^{M}z_{k,i}\psi_{k+1,j}\log[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}\} \tag{23}$$

となり，$q_k^*(\boldsymbol{z}_k)=\prod_{i=1}^{M}\boldsymbol{\psi}_{k,i}^{z_{k,i}}$ から

$$\psi_{k,i}^*\propto\prod_{i=1}^{M}[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}^{\psi_{k-1,i}^*}\times\prod_{i=1}^{M}[\boldsymbol{B}(x_k)\boldsymbol{A}]_{j,i}^{\psi_{k+1,i}^*} \tag{24}$$

が得られる．同様に $k=1,K+1$ の時はそれぞれ，

$$\psi_{1,i}^*\propto\pi_i\prod_{j=1}^{M}[\boldsymbol{B}(x_1)\boldsymbol{A}]_{i,j}^{\psi_{2,j}^*},\ \ \psi_{K+1,i}^*\propto\prod_{j=1}^{M}[\boldsymbol{B}(x_K)\boldsymbol{A}]_{j,i}^{\psi_{K,j}^*}\xi_i \tag{25}$$

となる．つまり式 (24) および式 (25) を満たす $\boldsymbol{\psi}_{k,i}^*$ は事後分布 $P(\boldsymbol{Z}|\boldsymbol{D})$ に対する最良の近似である．この $\boldsymbol{\psi}_{k,i}^*$ を用いれると GEM における E ステップは $\boldsymbol{\psi}_k$ の算出と期待値 $E[N_{i,j}|\boldsymbol{D}],E[Y_{i,l}|\boldsymbol{D}],E[R_i|\boldsymbol{D}]$ の計算となり M ステップはそれらの期待値から式 (16) でパラメータを更新する手続きとなる．$\boldsymbol{\psi}$ の算出を組み込んだ GEM によるパラメータ推定は以下のようになる．

Step1 時刻 $k=1,K+1$ および $k=2,3,\cdots,K$ について $\psi_{k,i}$ を更新する．

$$\psi_{1,i} \propto \pi_i\prod_{j=1}^{M}[\boldsymbol{B}(x_1)\boldsymbol{A}]_{i,j}^{\psi'_{2,j}}, \tag{26}$$

$$\psi_{K+1,i} \propto \prod_{j=1}^{M}[\boldsymbol{B}(x_K)\boldsymbol{A}]_{j,i}^{\psi'_{K,j}}\xi_i, \tag{27}$$

$$\psi_{k,i} \propto \prod_{i=1}^{M}[\boldsymbol{B}(x_k)\boldsymbol{A}]_{i,j}^{\psi'_{k-1,i}}\times\prod_{i=1}^{M}[\boldsymbol{B}(x_k)\boldsymbol{A}]_{j,i}^{\psi'_{k+1,i}}. \tag{28}$$

Step2 各期待値を求める．

$$E[N_{i,j}|\boldsymbol{D}] = \sum_{k=1}^{K}\psi_{k,i}\psi_{k+1,i}, \tag{29}$$

$$E[Y_{i,l}|\boldsymbol{D}] = \sum_{k=1}^{K}I(x_k=v_l)\psi_{k,i}, \tag{30}$$

$$E[R_i|\boldsymbol{D}] = \psi_{1,i}. \tag{31}$$

Step3 BW と同様に式 (16) を用いてパラメータを更新する．

GEM では Step1 から Step3 をパラメータまたは下限値が収束するまで繰り返す．

ここで $\boldsymbol{\psi}'$ はひとつ前の繰り返しで求めた $\boldsymbol{\psi}$ の値であることを示している．

最後にGEMを適用した学習の並列化について考える．図2はGEMを適用した学習の並列化の概念図である．Step1のψ^*の更新式に注目すると，各$k=1,\ldots,K$に対するψ_kの計算では先の反復におけるψに対するデータ依存のみで，$\psi_k, k=1,\ldots,K$間のデータ依存がない．よって，スレッド数に応じた並列計算が可能となる．ステップ2においても，和を求めるだけであるから，スレッド数に応じた並列計算が可能である．このように，GEMを適用した学習では効率的に並列化を行うことが可能であり，長い系列データを扱う際，大きな効果を生むと考えられる．

5 数値例

5.1 数値実験の環境

ここでは数値実験の環境について記述する．実験に用いた計算機のCPUはQuad Core AMD Opteron(TM) 2384 (2.7GHz/6MB L3キャッシュ）を2つ用いている．各CPUは4つのコアを持つので，原理的に8つのプロセスに対する処理を並列に行うことができる．またメモリは4GBで構成されている．

数値計算用プログラムは，Fortranを使用しており，GEMのGEステップ，BWのEステップ，そしてGEMとBW共通のMステップを実装し，それをC言語で作成したmain関数によって呼び出している．実装において，BLASを用いている．BLASとは，ベクトルと行列に関する基本線代数数操作を実行するライブラリAPIのディファクトスタンダードであり，BLASによって，GEMとBWの行列演算は最適化されている．並列化はOpenMPを用いている．OpenMPはメモリ共有型の並列化を行うためのAPIであり，コメント形式の指示文を挿入することでスレッドが生成され，低コストで並列計算を行えるようになっている．本稿ではプロセス数を踏まえ，最大8スレッドでの計算を行っている．

数値実験の初期値および教師データは乱数により生成した．またBWとGEMを比較する際の初期値は全て同じ値である．各数値実験の状態数nとシンボル数vは$5,10,25,50,75,100$と変化させ，系列長$dsize$は，$10,100,1000,10000$と変化させ数値実験を行った．スレッド数$thread$は，$1,2,4,6,8$と変化させた．

5.2 GEMを適用したHMMの適合性

本稿の以下の数値実験では，HMMの教師データへの適合性を計るために，GEMを適用したHMMの教師データに対する対数尤度の代わりに対数尤度の下限値を用いている．これは，GEMを適用した学習において，直接対数尤度を求めるためには，BWの一部を流用しなければならず，収束の判定を行う際，余計に時間がかかってしまうためである．そこで，ここでは下限値を対数尤度の代用することに対する検証を行う．下限値は式(17)から，完全データの対数尤度と近似分布によって求めることができる．図1は状態数とシ

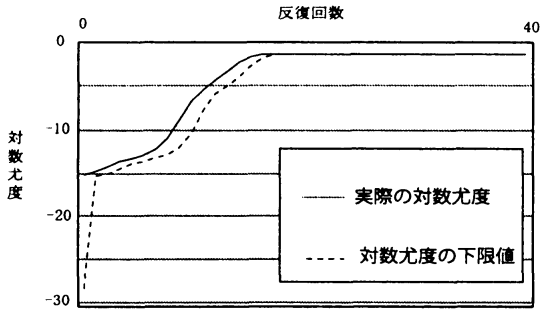

図3: GEMの対数尤度と下限値の収束.

ンボル数が10，教師データの系列長が10の時の対数尤度とその下限値の動きを表している．横軸が反復回数であり，縦軸が対数尤度である．まず，下限値が非減少であり，収束していることがわかる．そして下限値が収束している部分では対数尤度も収束している．

表1は状態数，シンボル数，系列長を変化させながら反復を1000回行った時の対数尤度とその下限値である．表中の値はそれぞれ，対数尤度とその下限値である．よく似た値を出していることがわかる．以上のことより，HMMの教師データへの適合性を計る際，下限値を対数尤度の代わりに近似値として用いることが許容されると言える．

表 1: GEM による学習の対数尤度と下限値.

	dsize=10		dsize=100		dsize=1000		dsize=10000	
	対数尤度	下限値	対数尤度	下限値	対数尤度	下限値	対数尤度	下限値
n=v=5	-6.07	-6.07	-138.95	-139.65	-1591.17	-1591.32	-16086.12	-16087.99
n=v=10	-1.39	-1.39	-168.33	-168.99	-2236.97	-2239.40	-23011.54	-23012.17
n=v=25	0.00	0.00	-142.71	-142.71	-2813.68	-2824.27	-32142.57	-32143.04
n=v=50	0.00	0.00	-95.03	-95.03	-2912.64	-2913.93	-38773.72	-38794.73

5.3 同じ反復回数での実行時間の差

次に GEM による学習と BW が並列化によって計算時間がどのように変化するかを観察する．ここでは計算時間を実行時間と CPU 時間に分けて考える．実行時間は，実際に計算が開始されてから終了するまでの

表 2: HMM 学習の実行時間.

n=v=50 dsize=10

	thread=1		thread=2		thread=4		thread=6		thread=8	
	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間
BW	0.71	0.71	0.56	1.13	0.52	2.09	0.56	3.35	0.59	4.69
GEM	1.18	1.17	0.96	1.92	0.89	3.55	0.92	5.51	0.93	7.39

n=v=50 dsize=10000

	thread=1		thread=2		thread=4		thread=6		thread=8	
	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間	実行時間	CPU 時間
BW	1405.75	1405.47	731.04	1438.45	471.53	1521.19	835.40	1607.81	341.69	1687.30
GEM	603.30	603.18	301.11	602.11	150.33	601.21	100.98	605.74	76.04	608.21

時間である．CPU 時間は全てのコアでのクロック数の合計を，一秒あたりのクロック数で割って計算している．どちらも単位は秒である．

表 2 は状態数及びシンボル数を 50 に固定した時の，実行時間と CPU 時間を表わしている．GEM と BW を短い系列で比較すると，どちらもスレッドの増加に従って実行時間が短縮されるものの，系列長が 10 の場合では，スレッド数が 6 から 8 になった時に，実行時間の値が増加している．また，系列長が 100 の場合でもスレッドの増加による実行時間の減少幅も小さい．この時の CPU 時間を比較すると，BW，GEM ともに，スレッドの増加とともに CPU 時間が増加している．CPU 時間は CPU の仕事量と捉えることもできる．スレッドが増加するにしたがって，CPU 時間が増加しているのはスレッド処理のためのオーバーヘッドが増加しているためと考えられる．短い系列の学習において CPU 時間の増加が大きく，実行時間の短縮幅が小さいのは，オーバーヘッドが占める割合が大きいからである．

次に長い系列の場合を比較する．系列長が 10000 の場合，スレッド数が 1 から 8 に変化すると，実行時間は，BW で 4 分の 1，GEM では 8 分の 1 程度になっている．これは，実行時間に占めるスレッド処理のオーバーヘッドの割合が小さくなり並列化の効果が顕著に現れることを示している．また，BW はスレッドの増加に対して，計算時間の短縮幅は頭打ちになっているように見えるが，GEM ではその傾向は見られない．アムダールの法則により，スレッド数をどんどん増やしていくと，GEM でも実行時間の短縮幅は頭打ちになると考えられるが，8 並列までででは，ほぼスレッド数と実行時間が反比例している．これは，BW では E ステップで前向き確率と後ろ向き確率の計算において，データ依存を含んでおり，高々2 並列での計算しか行うことができないのに対し，GEM では，各時刻ごとに並列化が可能であるためだと考えられる．これらのことから，長い系列を計算する場合，GEM において，並列化効果が特に顕著に現れており，GEM は BW より効率的に並列計算が行えると言える．

5.4 収束速度の比較

次に，収束速度を比較する．またその時の BW と GEM を適用した HMM の教師データに対する，対数尤度，あるいは対数尤度の下限値を見る．これにより GEM が HMM の学習として利用できるかどうかを考察する．実際の計算において，回数で打ち切るよりも，収束の判定によって打ち切ることが多い．収束の判

表 3: BW と GEM による学習の収束速度の比較.

n=v=50 dsize=100

	反復回数	対数尤度	thread=1	thread=2	thread=4	thread=6	thread=8
BW	34	-97.22	0.24	0.14	0.09	0.08	0.07
GEM	40	-95.03	0.26	0.14	0.09	0.07	0.07

n=v=50 dsize=10000

	反復回数	対数尤度	thread=1	thread=2	thread=4	thread=6	thread=8
BW	3202	-36945.78	3830.44	1972.80	1277.28	1047.38	931.44
GEM	2423	-37643.74	1426.17	713.52	357.73	238.50	179.40

定は相対誤差と絶対誤差がそれぞれの基準値より下回った時に，収束したと判定した．今回は相対誤差の基準値は 10^{-7}，絶対誤差の基準値は 10^{-5} としている．表 3 は BW と GEM による学習の収束速度を比較したものである．状態数とシンボル数が 50 の場合の系列長が 100 の時と，10000 の時を示している．反復回数は収束するまでに要した反復の回数であり，それ以外の数値は実行時間を示している．収束するまでの反復回数は BW が多ければ GEM も多く BW が少なければ GEM も少ない傾向がある．計算時間は上で示したように GEM の方が顕著に効果が現れている．GEM と BW の対数尤度には開きがある．これは BW, GEM いずれも収束しきっていないためだと考えられる．収束の判定条件を厳しくすることで，より近い値が出せると考えられる．

6 まとめと今後の課題

HMM に一般化 EM 法を適用することで並列化に適した学習アルゴリズムを構成できることが分かった．並列化を行うことにより GEM は BW より効率的な並列化を実現できる．しかし系列が短い場合は，その効果が現れにくく，その精度は BW に劣るという結論を導ける．また対数尤度の比較からある程度有効な学習が行われていることも分かった．実際の適用において BW を適用した HMM と GEM を適用した HMM でどの程度性能に差があるかを，出力確率に確率密度関数を持つ連続型 HMM での検証やトラフィック解析への適用を通して調べて行く予定である．

参考文献

[1] 中川 聖一,「確率モデルによる音声認識」, 電子情報通信学会, 1988.

[2] 前村 一哉, 小野 廣隆, 定兼 邦彦, 山下 雅史, 隠れマルコフモデルを用いた DNA 配列設計, 電子情報通信学会技術報告, vol. 106, no. 63, pp. 39-46, 2006.

[3] 菊池 洋光, 甲藤 二郎, 改良型 Embedded HMM を用いた顔画像認識の検討, 画像電子学会研究会講演予稿, Vol. 02-07, pp. 85-90, 2003.

[4] L.E. Baum, T. Petrie, G. Soules and N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164-171, 1970.

[5] 松野 裕之, 友利 記昌, 宮崎 崇, 西村 英樹, 神戸 尚志, 音声認識システムのハードウェア化の一手法 —HMM 出力確率計算のハードウェア化, 電子情報通信学会技術研究報告, vol. 104, no. 737, pp. 79-84, 2005.

[6] 汪 金芳, 田栗 正章, 手塚 集, 樺島 洋介, 上田 修功 ,「計算統計 I 確率計算の新しい手法」, 岩波書店, 2003.

[7] Z. Ghahramani and M.I. Jordan, Factorial hidden markov models, *MachineLearning*, vol. 29, pp. 245-275, 1997.