制限付き隠れマルコフモデルによる系列モデルの高速推定

加瀬雄一朗† 白井 匡人†† 三浦 孝夫†

† 法政大学理工学研究科システム理工学専攻 〒 184–8584 東京都小金井市梶野町 3-7-2
†† 島根大学大学院総合理工学研究科情報システム学領域 〒 690–8504 島根県松江市西川津町 1060
E-mail: †yuichiro.kase.7n@stu.hosei.ac.jp, ††shirai@cis.shimane-u.ac.jp, †††miurat@hosei.ac.jp

あらまし 本研究では系列データのモデル推定の高速化手法を提案する．具体的には，隠れマルコフモデル (Hidden Markov Model, HMM) の潜在状態の構造 (トポロジ) に対して，「状態遷移では後戻りしない」という制限を加えることで，モデルの学習を効率的に行う手法を提案する．実験では，完全グラフを仮定した隠れマルコフモデル (Ergodic-HMM) との比較を示し，また性能についての考察・分析を行う．

キーワード 隠れマルコフモデル，系列データ，分類

1. 前 書 き

近年，スマートフォン等の情報機器やインターネットの普及を背景として，様々な分野のタスクが電子化されており，それに伴い電子的に表現されたデータの量が増大している．例えば，新聞社の多くは紙版のみならず電子版の新聞も扱っており，現在，ウェブ上には大量の新聞記事のデータが存在する．電子化された情報を保持することは，紙媒体の情報などと比べると，維持・管理や検索などが比較的容易であり，実際，世の中の電子的なデータの量は増え続けている．そのような大量のデータを機械で効率的かつ自動処理することの重要性はますます高まっている．例えば，多くの企業は，このような大量のデータを “ビッグデータ” と称し，それらを利用して，顧客の満足度を向上させることを試みている．

情報処理では，想定するデータ領域に対応したモデルを構築する．パラメタをもつモデルを構築し，そのパラメタを具体的なデータに合わせて調整（学習）することで，そのデータ領域に対応した特徴を汎用的に表現することができる．例えば，自動分類では，各カテゴリに対応したモデルをあらかじめ構築・学習しておき，分類対象事例の特徴が，どのモデルの特徴に一番近いかを判断することで分類を行う．

各分野に依存した情報には，テキスト，音声，画像など多様なデータ表現がある．データは表現に応じて異なる性質を有し，それぞれに応じたモデル構築が必要となる．

データの性質の一つに “系列” がある．系列データとは，データを構成する要素が連続し，その要素の出現が互いに関連しあうような性質をもつデータである．系列データの例としてテキストデータがある．例えば，{ 吾輩，は，猫，で，ある } という，形態素に分解された日本語の文章があるとする．この場合，“吾輩” と “は”，“は” と “猫” など，それぞれの要素の間には関連があり，その順番が意味を持つ．順番を入れ替えて，“猫吾輩ではある”，などとすると日本語として意味をなさなくなる．

本研究では系列の性質を持つデータのモデル推定の高速化手法を提案する．具体的には，隠れマルコフモデル (Hidden Markov Model, HMM) の状態構造 (トポロジ) に対して，系列中で「状態遷移では後戻りしない」という制限を加えることで，モデルの学習を効率的に行う手法を提案する．

本研究の貢献は以下の通りである．

- 系列中で同じ状態は繰り返さない場合のモデルの提案
- 提案モデルの高速推定方法の提案
- 実験による提案モデルの妥当性の確認

2 章以降ではこの論文は以下のように構成される．2 章では系列データのモデル化について述べる．特に系列データのモデル化の代表的手法である HMM について述べ，その関連研究について述べる．3 章では本研究で提案する制限付き隠れマルコフモデルについて述べ，4 章ではその推定方法について述べる．5 章では実験の結果より提案手法が系列データのモデル化において有効なことを，構造の分析などから示す．そして 6 章で結びとする．

2. 系列データのモデル化

2.1 マルコフモデル

系列データの確率推定にモデルを適用するには，要素間の関連をモデルに組み込む必要がある．しかし，系列データに含まれる全要素の関連を一度に考えることは困難である．なぜなら，要素同士の順序が複雑になり，すべての組み合わせを考慮することが不可能だからである．

系列的な事象を表現するにはマルコフモデルがよく用いられる．マルコフモデルでは現在の状態は一つ前の状態からのみ影響を受け，系列データ中の各要素 (観測値) が現在の状態にのみ依存し，観測値が現在の状態に対応して確率的に決定すると単純化する．すなわち，現在の状態は一つ前の状態に影響して決定するため要素同士の順序の全ての組み合わせを考える必要がなくなる．そのようにモデルを単純化して考えることで，系列データの確率推定を容易に適用できる．例えば，音声認識や形態素解析などでマルコフモデルはよく用いられる．

そのように観測値とそれが生成される状態を分けて単純なモデルを構築した場合，そのモデルの推定 (学習) には，学習データとして観測値とそれに対応する状態が組合わせて必要となる．しかし，全ての場合で状態が観測されるとは限らないため，

多量の学習データを構築するには，各系列中の要素に対応した状態を人手でラベル付けする必要がある．そのため，学習データに含まれる系列の要素一つ一つにラベル付けを行うため膨大な作業コストがかかる．また，ラベル付けに主観が入ることは避けられず，そのラベル付けに客観性が欠けるという問題点もある．

2.2 隠れマルコフモデル

状態が未観測の場合，系列要素に対してラベル付けを行わず，代わりに，事前に状態に関して何の仮定も置かないで，尤度の最大化による状態推定で系列をモデル化する手法がある．そのような手法の一つとして隠れマルコフモデル (Hidden Markov Model, 以下 HMM) [1] [2] [3] がある．HMM は次の 5 つのパラメタによって定義される [13].

(1)$Q = \{q_1, \cdots, q_N\}$：状態集合 (状態数 N)
(2)$\Sigma = \{o_1, \cdots, o_M\}$：出力記号集合 (記号の種類 M)
(3)$A = \{a_{ij}\}$：状態遷移確率分布 (状態番号 i, j)
(4)$B = \{b_i(o_t)\}$：記号出力確率分布 (時刻 t の記号 o_t)
(5)$\pi = \{\pi_i\}$：初期状態確率分布

HMM は確率的に遷移する未観測の状態と各状態に対応した記号出力確率分布を持つ確率オートマトンである．前述のマルコフモデル同様，ある状態は一つ前の状態のみから影響を受けて決定される．また，記号の出力はその時点での状態のみに影響を受けて決定される．ただし，パラメタ (3),(4),(5) は明示的には与えられない．

潜在状態の構造 (トポロジ) に対して，あらかじめ何の仮定も置かず，事前に与えられるのは状態数 N，記号の種類 M のみとした HMM を Ergodic-HMM と呼ぶ．これは，潜在状態の構造に完全グラフを仮定することに等しい．言い換えれば，すべての状態は自分を含む全て状態へのリンクをもつことを意味する．Ergodic-HMM の推定すべきパラメタは A，B，π である．それらのパラメタの推定方法としては，EM アルゴリズムを用いた方法である，Baum-Welch アルゴリズム [3] がよく知られている．Baum-Welch アルゴリズムでは前向き確率，後ろ向き確率と呼ばれる確率値を計算し，それらの値に基づき，ある状態から別の状態へ遷移する回数の期待値を計算することで，各パラメタの値を更新する．

しかし，前向き確率，後ろ向き確率の計算に必要な計算量はそれぞれ，状態数 N，系列長 T とした時，$O(N^2T)$ であり [13]，状態数が増えるにつれ，計算量が状態数の 2 乗に比例して増える．しかも，それらの値に基づき遷移回数の期待値計算，パラメタの更新のステップを行う手順が，EM アルゴリズムの各繰り返し計算ごとに必要となるため，計算量は膨大となり，特に，状態数が比較的多い場合における学習は困難となる．

2.3 関 連 研 究

ここでは本研究に関連する研究を述べる．トポロジを極限まで単純化し，状態遷移を一直線のパスに制限した HMM を Left to Right HMM [5] [6] という．Left to Right HMM では潜在状態の分岐が途中の状態で存在せず，各モデルが初期状態と最終状態を一つずつ持つという特徴がある．Left to Right HMM のパラメタ A，B も Ergodic-HMM 同様，Baum-Welch アルゴリズムで推定される [14]．Left to Right HMM は音声認識の分野でよく用いられる．Left to Right HMM は Ergodic-HMM と比べると単純なモデルであり，そのモデルの表現能力も大きく制限されるが，この Left to Right HMM のトポロジは音声データの特徴をモデル構築時に考慮し，モデル学習を単純化させているといえ，実際，音声の分野では代表的なモデルである．

HMM 以外に系列のモデル化をする有名な手法として，線形動的システム (Linear Dynamical System, LDS) [7] がある．これは状態遷移，観測値の出力がともに正規分布であると仮定した系列モデルである．LDS は前述の HMM の状態が連続値で表されるとした HMM であると解釈できる [9]．LDS の推定には HMM 同様 EM アルゴリズムが用いられる．また，Martens [8] は LDS を，収束の保証された近似的に推定する手法を提案している．この手法により，通常の EM アルゴリズムを使った場合よりも高速に LDS の推定ができる．

HMM の推定を高速に行う手法として，Matsuyama [11] の方法がある．これは通常よりも収束が速いとされる α-EM [10] アルゴリズムと呼ばれる手法を用いて Ergodic-HMM の推定を行う手法である．実験では実際に，通常の EM アルゴリズムを使った HMM よりも収束が速いことが示されている．しかし，HMM の推定における計算上のボトルネックは前向き確率・後ろ向き確率の計算であるが，その点は解決されていない．

HMM と同様に事象間の関係を確率的なグラフ構造で表現したモデルにベイジアンネットワーク [12] がある．ベイジアンネットワークではグラフにより変数の関係を表現し，変数間の因果関係を確率的に記述する [15]．またそのグラフ構造を，ノードを確率変数，リンクを依存関係とした有向非巡回グラフ (Directed Acyclic Graph, DAG) に制限することでモデルを簡易化している．本研究との大きな違いは，確率変数，つまり，観測値同士の関連をベイズ理論により直接記述したモデルであるという点である．

3. 制限付き隠れマルコフモデル

Ergodic-HMM では潜在状態の構造に完全グラフを仮定する．このモデルは，ある系列長に対し，与えられた全ての状態の組み合わせ (パス) を表すことができるモデルである．しかし，各系列の要素に対して，あらゆる状態が割り当てられることを許容しているので，計算量が大きいという問題がある．したがって，本研究ではこの学習時の計算量を減らすことを考える．

ここで，潜在状態列で，すでに到達した状態には戻らないような場合を考える．つまり，一旦ある状態に遷移したら再びその状態には戻らないような場合である．例えば，状態を英文センテンスの構成要素である “SVOC” と見た場合，一旦，状態 “S” から “V” に遷移したら，再び状態 “S” に戻ってくることはない．また，文書構造が “起承転結” に従うとする場合，一旦状態 “起” から状態 “承” に遷移したら，再び状態 “起” に戻ってくることはない．このような場合に完全グラフを仮定した HMM を用いて系列のモデル化を行うことは，計算量の点から

無駄が多いといえる．なぜなら，完全グラフでは，以前に滞在した状態に再び戻る遷移を許容しており，その場合，その “再遷移” の場合における確率値も計算する必要があるからである．

本研究では，「状態列中で，すでに到達した状態は繰り返さない」という場合をモデル化する．ベイジアンネットワークにも用いられるグラフ構造である有向非巡回グラフ (Directed Acyclic Graph, 以下 DAG) に自己サイクル (同じ状態での繰り返し) を追加した構造を HMM の潜在状態構造に用いて，系列データのモデル化を試みる．DAG 構造を用いることで，系列データ中に状態の繰り返しが無いという状況を表現することができる．また，Left to Right HMM と同様に初期状態と最終状態を一つずつ持つと仮定する．以上の潜在状態の構造に対する制限により，前向き確率，後ろ向き確率の計算時に必要な計算量を大幅に削減できる (詳細は後述する)．

4. 制限付き隠れマルコフモデルの高速推定

ここでは，制限付き隠れマルコフモデルの推定方法を提案する．まず，モデル推定に必要となる値，前向き確率，後ろ向き確率を示す．前向き確率，後ろ向き確率はそれぞれ以下のように定義される．

$$\alpha_t(i) = P(o_1, \cdots, o_t, q_t = i) \tag{1}$$

$$\beta_t(i) = P(o_{t+1}, \cdots, o_T | q_t = i) \tag{2}$$

これらの値を使い，遷移確率を計算する．

$$\begin{aligned}\xi_t(i,j) &= P(Q_t = i, Q_{t+1} = j | o_1, \cdots, o_T) \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_T(i)}\end{aligned} \tag{3}$$

ただし，T は最終時刻，言い換えれば，ある系列の最終要素の位置を表し，$Q_t = i$ は時刻 t での状態が i であることを表す．式 (3) は全時刻において，状態 i から状態 j に遷移する確率を表している．また，滞在確率を計算する．

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{4}$$

式 (4) は全時刻において，状態 i に滞在する確率を表している．

Ergodic-HMM における前向き確率，後ろ向き確率の計算量は，それぞれともに $O(N^2T)$(状態数 N，系列長 T) である．HMM の計算においては前向き確率・後ろ向き確率が計算上のボトルネックになっており，逆に言えば，この部分が高速に実行できれば HMM の推定にかかる計算コストを大幅に削減できる．制限付き HMM ではその計算量が削減される．これは，以下に述べる 2 点によるものである．

(1) 各時刻にありうる状態数の減少

例えば，ある HMM の潜在状態の構造が，N 個の状態からなる完全グラフとすると，各時刻 t に割り当てられる状態は，すべての時刻で N 通りである．一方，ある制限付き HMM において，状態 i から状態 j へのリンクが存在しないとする．その時，時刻 $t-1$ で状態 i である場合，時刻 t では，状態 j である可能性を考えなくてよい．このため，各時刻にありうる状態数を削減することができる．

(2) 最終状態までの遷移回数が残りの出力列長より小さい

制限付き HMM では最終状態が決まっているため，最終時刻 T において，必ず最終状態に到達していなければならない．このため，ある系列に対して，「最終状態までの遷移回数が残りの出力列長より小さい」の関係が成り立つ．例えば，最終状態までたどりつくために必要な遷移回数が状態 i で 3 回であるとする．そして，残りの系列長，つまり最終時刻 (系列要素の最終位置番号) と現在の時刻 (現在の系列要素の位置番号) の差 $T-t$ が 4 であるとすると，この後，状態 $i \to$ 状態 $i \to \cdots$ と遷移することはあり得ない．なぜなら，2 回目に状態 i に遷移した時点で，最終状態までの遷移回数 $= 3$，残りの出力系列長 $= 3$ となり，「最終状態までの遷移回数 $<$ 残りの出力系列長」の関係が成り立たず，残りの出力記号長では最終状態に到達できないからである．

以上の 2 点を前向き確率・後ろ向き確率を計算する際に，自動的に判定する．以下に前向き確率を計算するアルゴリズムの概要を示す．

```
1:  procedure FORWARD-ALGORITHM(sequence)
2:      for all sequence.length : t do
3:          for all t にありうる状態集合 : j do
4:              for all t − 1 にありうる状態集合 : i do
5:                  α_t(j) = α_{t−1}(i) a_ij
6:              end for
7:              α_t(j) = α_t(j) b_j(o_t)
8:          end for
9:      end for
10:     return { 前向き確率の集合 }
11: end procedure
```

実際は，アルゴリズム中の 3 行目と 4 行目の部分で前述の 2 点を判断しながら for 文を実行する．この部分により，状態 i から状態 j への遷移の組み合わせを減らすことができるため，前向き確率の計算量を大幅に削減できる．後ろ向き確率の計算では，初期状態と最終状態をそれぞれ逆に読みかえることで同様の手続きで計算でき，実際，計算量も前向き確率の場合と同様になる (注1)．

そして，以上のようなアルゴリズムによって計算した，前向き確率・後ろ向き確率を用いて，式 (3)，式 (4) で表される遷移確率，滞在確率が計算される．本研究では，これらの値を用いて，EM アルゴリズムを利用することで，ある系列に関して，与えられたパラメタに対し，それらの値を更新していくことで，制限付き HMM のパラメタ推定を行う [4].

(注1)：計算量は潜在状態の構造によって異なるため一般的な議論をすることはできない

5. 実　　験

5.1 準　　備

ここでは提案手法を分類問題に適用することで，そのモデル化の妥当性を示す．分類は，各クラスに対応したモデルを構築し，各事例がどのモデルに一番近いかを，尤度によって判断し，その最大尤度となるクラスを選ぶ．本研究では，モデル化の妥当性を分類の正解率 (Accuracy) で評価し，実行時間と抽出された構造の考察を行う．また，実験結果を主に実行時間において比較検討するため，同じ状態数を持つ Ergodic-HMM での実験も併せて行う．本実験では状態数を 11 とする．

本研究では The 20 Newsgroups Data Set(注2) を実験に用いる．これは文書データが，単語を要素とした系列データであり，単語がそれぞれ人間が判別できる意味を持つため，推定された各状態に対する考察がしやすいからである．The 20 Newsgroups Data Set は全部で 18846 件の文書を含み，学習データが 11314 件，テストデータが 7532 件で構成される．これらの各文書はすべて，一つのカテゴリ (クラス) に排他的に所属しており，クラスの種類は 20 種類ある．表 1 に各クラスとそれに対応する番号を示す．本研究では一つのセンテンス (ピリオドからピリオドまで) を一つの系列データとして扱う．つまり，一つの文書には複数の系列が含まれる．学習データに含まれる総系列数は 146611 系列である．テストデータとしては，文書中に 5 系列，つまり 5 センテンス以上含まれる文書のみを使用する．そのような文書は 5268 件である．

各文書は，scikit-learn (注3) を使用し，各文書ごとに付随する “footers” と “headers” を削除して使用する．また，Natural Language Toolkit(NLTK) (注4) を使用し，小文字化，センテンス分割，トークン化，ステミングを行う．さらに，数字列は記号 `d*`に変形してすべて同じものとみなす．

クラス	クラス名
1	alt.atheism
2	comp.graphics
3	comp.os.ms-windows.misc
4	comp.sys.ibm.pc.hardware
5	comp.sys.mac.hardware
6	comp.windows.x
7	misc.forsale
8	rec.autos
9	rec.motorcycles
10	rec.sport.baseball
11	rec.sport.hockey
12	sci.crypt
13	sci.electronics
14	sci.med
15	sci.space
16	soc.religion.christian
17	talk.politics.guns
18	talk.politics.mideast
19	talk.politics.misc
20	talk.religion.misc

表 1　20 Newsgroups のクラス

(注2)：http://qwone.com/∼jason/20Newsgroups/
(注3)：http://scikit-learn.org/stable/
(注4)：http://www.nltk.org/

5.2 実 験 結 果

表 2 にクラスごとに一回パラメタを更新するのにかかる学習時間を示す．Ergodic-HMM では合計で 4987.12ms，平均で 249.356ms であるのに対して，提案手法では合計で 1478.34ms，平均で 73.917ms であり，平均 29.6%の実行時間で学習を行っている．

表 3 に提案手法の正解率と Ergodic-HMM との正解率の比較を示す．提案手法でテストデータが元々所属するクラスと分類の結果のクラスが一致した事例数は 4020 件なので，平均正解率は 0.7631 であり，Ergodic-HMM との比較で，平均 0.0118 の改善である．クラスごとに比較すると，10 種類のクラスでは正解率は改善し，その他の 10 種類のクラスでは悪化している．特に，alt.atheism が 0.2868 改善，soc.religion.christian が 0.5062 改善と顕著な改善が見られる．一方，comp.os.ms-windows.misc では 0.6627 悪化，talk.religion.misc では 0.3267 悪化と大きく正解率が悪化している．

表 4，表 5 にそれぞれ提案手法，Ergodic-HMM において各事例がそれぞれどのようなクラスと推定されたかを示す．表 4 より，実際に所属しているクラスと推定されたクラスがどのように異なるかがわかる．例えば，表 4 の行を見ると，クラス alt.atheism に実際に所属する事例 265 件のうち，alt.atheism と推定されたものには，実際に alt.atheism に所属している事例 215 件の他に，sci.crypt が 1 件，sci.med が 2 件，sci.space が 4 件，soc.religion.christian が 10 件，talk.politics.guns が 5 件，talk.politics.mideast が 4 件，talk.politics.misc が 4 件，talk.religion.misc が 20 件含まれる．また，表 4 の列を見ると，クラス comp.os.ms-windows.misc と推定された事例 5 件のうち，実際に comp.os.ms-windows.misc に所属している 2 件のほかに，comp.graphics が 1 件，comp.sys.mac.hardware が 1 件，comp.windows.x が 1 件が含まれる．

クラス番号	提案手法実行時間 (ms)	Ergodic-HMM 実行時間 (ms)
1	75.78	240.4
2	54.37	182.79
3	135.11	418.33
4	47.86	159.84
5	42.24	138.46
6	83.59	263.2
7	33.24	106.46
8	56.12	182.8
9	48.93	158.25
10	55.62	181.85
11	83.59	263.08
12	99.72	326.92
13	49.52	164.83
14	77.97	254.75
15	77.04	251.04
16	94.67	320.53
17	88.52	372.75
18	123.44	444.15
19	93.96	366.93
20	57.05	189.76
合計	1478.34	4987.12
平均	73.917	249.356

表 2　実 行 時 間

5.3 評価と考察

前述のとおり，クラス soc.religion.christian では正解率が 0.5062 改善と顕著な上昇が見られる一方，comp.os.ms-windows.misc では 0.6627 悪化と大きく正解率が下がってい

クラス番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	合計
1	215	0	0	0	0	0	0	0	0	0	0	1	0	2	4	10	5	4	4	20	265
2	2	164	1	6	11	1	0	1	1	0	0	11	6	1	8	1	0	0	0	0	214
3	1	53	2	86	32	35	2	3	2	0	0	20	3	4	1	4	1	0	5	1	255
4	0	9	0	185	28	1	5	3	1	0	0	4	25	0	0	0	0	0	0	0	261
5	0	5	1	13	188	0	2	6	2	0	0	3	15	3	2	0	0	0	0	0	240
6	0	30	1	6	5	193	1	0	0	0	0	4	0	3	2	0	0	0	1	0	246
7	0	1	0	10	9	1	109	8	6	1	0	0	5	1	4	0	2	0	1	0	158
8	0	2	0	1	0	0	0	267	8	0	0	1	3	0	4	0	1	0	3	0	290
9	0	1	0	0	0	0	2	15	245	0	0	0	2	0	0	0	3	0	3	0	271
10	5	1	0	0	1	0	0	4	0	258	2	0	0	0	1	2	2	1	11	0	288
11	5	1	0	0	0	0	0	3	6	10	246	0	1	0	0	0	3	0	2	0	277
12	1	1	0	0	0	2	0	1	0	0	0	256	6	2	0	1	10	2	6	1	289
13	0	6	0	14	8	0	2	9	10	0	0	22	185	1	2	1	0	0	0	0	260
14	12	2	0	0	0	0	0	3	4	0	0	0	2	236	4	4	1	1	10	2	281
15	3	10	0	0	0	0	0	3	1	0	0	1	6	5	240	1	1	0	19	1	291
16	31	1	0	0	0	0	0	0	0	0	0	1	0	1	0	255	4	3	3	23	322
17	1	0	0	0	0	0	0	2	2	0	0	4	0	1	1	0	252	1	19	10	293
18	16	0	0	0	0	0	0	0	1	0	0	2	0	0	0	2	0	267	21	4	313
19	7	0	0	0	0	0	1	0	0	0	1	2	0	2	8	1	71	3	148	8	252
20	48	1	0	0	0	0	0	0	1	0	0	0	0	2	5	14	16	3	3	109	202
合計	347	288	5	321	282	233	124	328	290	269	249	332	259	264	286	296	372	285	259	179	5268

表 4　提案手法 micro precision

クラス番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	合計
1	139	0	0	0	0	0	0	1	1	0	0	1	0	6	5	1	1	1	6	103	265
2	0	165	9	6	14	1	0	1	0	0	0	0	2	8	6	0	0	0	0	2	214
3	0	18	171	25	15	3	3	0	1	0	0	0	2	7	2	0	0	0	2	6	255
4	0	7	18	180	35	1	4	1	0	0	0	0	14	1	0	0	0	0	0	0	261
5	0	3	8	13	198	0	2	1	1	0	1	0	6	4	3	0	0	0	0	0	240
6	0	26	17	6	3	188	1	1	0	0	1	0	0	2	0	0	0	0	1	0	246
7	0	0	2	10	5	0	126	5	4	0	3	0	3	0	0	0	0	0	0	0	158
8	0	1	0	1	0	0	2	269	5	0	2	0	5	1	3	0	0	0	0	1	290
9	0	1	2	0	0	0	2	14	247	1	0	0	1	0	0	0	0	0	2	1	271
10	0	0	1	1	0	0	0	1	0	257	22	0	0	0	1	0	0	0	1	4	288
11	0	0	1	0	0	1	0	1	0	2	269	0	0	0	0	0	1	0	0	2	277
12	0	5	2	1	5	2	0	2	1	0	1	212	13	9	2	0	13	0	6	15	289
13	0	7	15	18	10	0	2	6	5	0	0	1	183	9	4	0	0	0	0	0	260
14	0	0	0	0	0	0	0	1	2	0	1	0	3	256	4	0	0	1	4	9	281
15	0	10	0	0	2	0	0	1	3	0	0	0	4	5	253	0	1	0	6	6	291
16	1	1	2	1	0	0	0	1	0	0	1	0	0	6	0	92	2	0	1	214	322
17	0	0	0	0	0	0	0	1	0	0	0	1	2	5	1	0	226	1	12	44	293
18	6	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	3	226	24	51	313
19	3	1	0	0	0	0	1	1	1	0	1	1	0	6	7	0	54	1	126	49	252
20	5	1	0	0	0	0	0	0	1	0	0	0	0	7	5	0	5	1	2	175	202
合計	154	246	248	262	287	196	144	308	273	260	302	216	238	333	296	93	306	231	193	682	5268

表 5　Ergodic-HMM micro precision

クラス	件数	一致数	正解率	vs. Ergodic-HMM
alt.atheism	265	215	0.8113	+0.2868
comp.graphics	214	164	0.7664	-0.0047
comp.os.ms-windows.misc	255	2	0.0078	-0.6627
comp.sys.ibm.pc.hardware	261	185	0.7088	+0.0192
comp.sys.mac.hardware	240	188	0.7833	-0.0417
comp.windows.x	246	193	0.7846	+0.0203
misc.forsale	158	109	0.6899	-0.1076
rec.autos	290	267	0.9207	-0.0069
rec.motorcycles	271	245	0.9041	-0.0074
rec.sport.baseball	288	258	0.8958	+0.0035
rec.sport.hockey	277	246	0.8881	-0.0830
sci.crypt	289	256	0.8858	+0.1522
sci.electronics	260	185	0.7115	+0.0077
sci.med	281	236	0.8399	-0.0712
sci.space	291	240	0.8247	-0.0447
soc.religion.christian	322	255	0.7919	+0.5062
talk.politics.guns	293	252	0.8601	+0.0887
talk.politics.mideast	313	267	0.8530	+0.1310
talk.politics.misc	252	148	0.5873	+0.0873
talk.religion.misc	202	109	0.5396	-0.3267
合計	5268	4020	0.7631	+0.0118

表 3　正解率 (Accuracy)

る (表 3)．ここではこれらの点について考察しモデル化の妥当性を判断する．

まず，図 1 にクラス soc.religion.christian で抽出された構造を示す．各ノード，リンクが，それぞれ状態，状態間の遷移を表す．各リンクの上に表示される数値は状態遷移確率を表す．大まかな構造としては，状態 1 を初期状態として，その後，3 つの別々のパスに分かれ，最後に最終状態 11 で，再び分かれたパスが一つになるという構造である．ここで，初期状態 1 から初めて別の状態に遷移するリンクを見ると，それぞれ，状態 2，状態 3，状態 4 への 3 通りの遷移がある．そして，それぞれのパスへ遷移する確率を見ると，状態 3 への遷移が 0.805195 となっており，ほとんどの場合，状態 1 から状態 3 へ遷移し，その後，状態 6→ 状態 9→ 状態 11 とたどっていくことがわかる．そのほかのパスは，初期状態からの遷移確率が低いため，系列中にほとんど起こらない構造や記号が集まる．

表 6 にクラス soc.religion.christian の各状態での記号出力確率分布を示す．表 6 は各状態で現れる確率の高い上位 20 記号 (単語) を表す．状態 1 では “i”，“the”，“it”，“thi” など (注5)，英文構造における S になれる単語が多く表れていることがわかる．また状態 3→ 状態 6→ 状態 9→ 状態 11 と遷移するにしたがって，英文構造における V になれる単語数が徐々に減少していることがわかる．状態 3 では “is”，“wa”，“are” など，上位 20 単語中に 12 単語現れる．続く状態 6 では 6 単語，状態 9 では 4 単語，状態 11 では 3 単語現れる．したがって，抽出された構造は，英文において，各要素が S→V→O/C→ · · · と遷移する英文の構造がモデル化されていることがわかる．

(注5)：表の記号部分はステミング後の単語であるため，表記が一般的に使われる形と異なる

状態 1		状態 3		状態 6		状態 9		状態 11	
記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率
i	0.078109	is	0.037266	the	0.023011	the	0.032611	the	0.050645
the	0.04302	the	0.024884	not	0.022838	that	0.021676	of	0.031227
it	0.029	you	0.015616	that	0.020576	to	0.021034	to	0.030018
in	0.025211	s	0.01524	is	0.019637	d*	0.018819	and	0.022712
if	0.024272	articl	0.012673	t	0.017323	a	0.01616	that	0.019057
thi	0.018136	wa	0.010889	a	0.015694	is	0.012182	a	0.017437
but	0.016047	are	0.010436	of	0.014724	of	0.011863	in	0.01705
he	0.013039	don	0.009729	to	0.013668	i	0.010578	d*	0.017029
and	0.012054	have	0.009387	d*	0.0103	be	0.007417	is	0.016144
we	0.010994	can	0.009019	apr	0.009572	thi	0.006333	i	0.010759
there	0.010511	would	0.008701	are	0.008169	in	0.006117	be	0.01031
that	0.009457	d*	0.007731	be	0.007979	have	0.005328	it	0.010064
you	0.00932	do	0.007477	have	0.007891	you	0.005186	not	0.008614
as	0.009141	am	0.007062	it	0.007655	and	0.005112	for	0.007793
so	0.007051	of	0.007046	in	0.006886	it	0.005003	as	0.007455
what	0.006897	think	0.006693	you	0.005664	not	0.00454	you	0.007304
a	0.006865	i	0.00653	say	0.005363	we	0.004245	are	0.007076
d*	0.00663	m	0.006312	wa	0.005268	believ	0.004178	god	0.007029
howev	0.006381	we	0.005769	i	0.004301	with	0.003989	thi	0.006285
when	0.006324	doe	0.005692	thi	0.004234	for	0.003987	with	0.006185

表 6　soc.religion.christian の記号出力分布

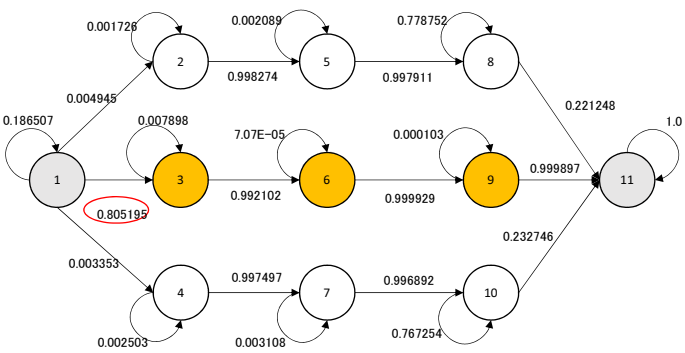

図 1　soc.religion.christian で抽出された構造

同様に，図 2 にクラス comp.os.ms-windows.misc で抽出された構造を示す．大まかな構造は soc.religion.christian の場合と同様である．それぞれのパスへ遷移する確率を見ると，初期状態 1 から状態 2 への遷移確率が 0.412144，初期状態 1 から状態 4 への遷移確率が 0.568374 となっており，ほとんどの場合，状態 1→ 状態 2→ 状態 5→ 状態 8→ 状態 11，および状態 1→ 状態 4→ 状態 7→ 状態 10→ 状態 11 の 2 通りのパスをたどることがわかる．その場合の各状態での記号出力確率分布を表 7 に示す．表 7 は表 6 と同様に，各状態で現れる確率の高い上位 20 記号 (単語) を表す．

まず，状態 1→ 状態 4→ 状態 7→ 状態 10→ 状態 11 のパスを考察する．このパスでは，前述の soc.religion.christian における遷移，状態 1→ 状態 3→ 状態 6→ 状態 9→ 状態 11 と同様の特徴を有する．実際，状態 1 では “i”，“the”，“it”，“you” など，英文構造における S になれる単語が多く表れている．また状態 4→ 状態 7→ 状態 10→ 状態 11 と遷移するにしたがって，英文構造における V になれる単語数が徐々に減少していることも同様である．状態 4 では “have”，“is”，“can” など，上位 20 単語中に 8 単語現われ，続く状態 7 では 7 単語，状態 10 では 3 単語，状態 11 では 0 単語現れる．この場合の特筆すべき特徴は，状態 11 で英単語ではない記号，例えば，“ax”，“q”，“p”，“r” などが現れていることである．状態 11 は最終状態であるため，系列の長い事例において，最後の長さの「つじつま合わせ」になる場合が多く，そのため，英文の構造に関係なく，最終状態には各文書において多く表れる単語が集まりやすい．したがって，クラス comp.os.ms-windows.misc の文書には，前述のような英単語でない記号が多く表れることを意味する．

次に，状態 1→ 状態 2→ 状態 5→ 状態 8→ 状態 11 のパスを考察する．このパスでの特徴は，最終状態 11 だけでなく，状態 2，状態 5，状態 8 の中間の状態においても，前述のような英単語でない記号が多く出現することである．したがって，このパスでは，初期状態以外では英単語でない記号が多く表れるといえる．以下にこのような事例を示す．以下は The 20 Newsgroups Data Set のクラス comp.os.ms-windows.misc を持つ学習データに含まれる，9976 と名前の付けられたファイル内の文書を前述のように前処理した結果である．ファイル上部から，しばらくは通常の英文センテンスが現れた後，このコーパスは特殊記号列を大量に含む．クラス comp.os.ms-windows.misc にはこのような事例が含まれており，この種の事例が英文のセンテンス構造をモデル化することを阻み，結果として分類正解率を大きく悪化させたと考えられる．

due to the resolut and size it is in 14 part thi is a uuencod bitmap

… (中略)

part 1 of 14 begin 644 roman bmp h p 8 h g 4 8 m ht t a k 1 77 rljjun p ui e 3 m u u 6 kl jsoh mb ht 7 y 1 ko mkh 89 hl m w 2 u i 0 dt 896 h x 7 i 8 mi h 2 g iyd g 504 59 h w lk a i xp oi 2 f d 44 g f e hp 5 885 7 ql vxo y 4 ua 84 m eu k i 441 d 84 e d t x e 887 f 84 v m 5 i d 4 8 pl d mh xhc d 79 rlk 5685 hl t e em i 4 d 1 6 lo zzzoh x e 4 dt m nnhlc kjzzzo 43 5 7 o d 1 7 pl 3 f mokx 2 d k h h 8 hd 55896 kk h mb 0 6 oi 4 2 m fdz 1 i m i k b lhzv hz z zrck d s mrd g z w s t mv a rg s as k isrmamem mv 8 sq k rlisecqmv v am u i w m mrmc g v w t 6 im a 7 zrma w rd 8 v k mv ho b 8 hzv ho sk b ov m vtov b z lk lhz 8 c m b zrcho hz hz lhz lk rlhz w ck v im m as d 9 rli as w s w srg s w m m 1 as v qm us p 6 qs 71 w m 1 1 p ww mw uwt v w t w 0 e w w w w w w 2 c w ca 22 wxte 0 0 x

x c mxte x cxtgcx c cx cbdf bn bn cx xxkcxwlr mx cbhj
goc s c hi xs gn gn 6 bc r j s bdxrbhi o y bn hj 6 g 8 ty

… (以下略)

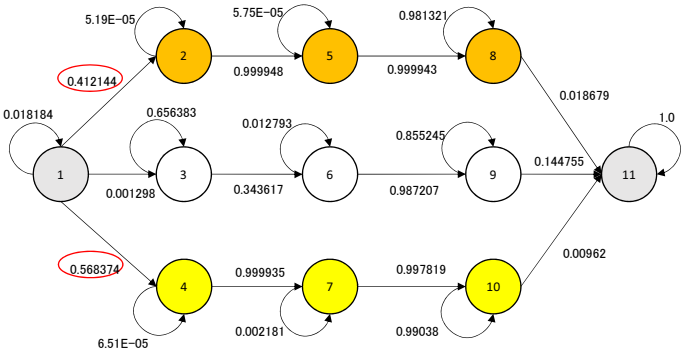

図 2　comp.os.ms-windows.misc で抽出された構造

6. 結　　論

本研究では系列データのモデル化を高速に行う方法を提案した．具体的には，隠れマルコフモデル (Hidden Markov Model, HMM) の状態構造 (トポロジ) に対して，系列中で同じ状態は繰り返さないという仮定の下に制限を加え，モデルの学習を効率的に行った．実験結果より，提案手法は，Ergodic-HMM を用いてモデル化を行った特よりも，平均で 0.296 倍の時間で学習を行えることを確かめた．また，分類では正解率で 1.0157 倍の性能改善が見られた．さらに，実際に抽出された状態の遷移構造を考察・分析することにより，英文のセンテンスの構造を潜在状態によりモデル化できたことを確認した．

文　　献

[1] L. R. Rabiner and B. H. Juang, “An introduction to hidden Markov models,” IEEE Acoustics, Speech and Signal Processing Magazine, 3:416, 1986.

[2] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” The Annals of Mathematical Statistics, vol.37, pp.1554-1563 1966.

[3] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A Maximization Technique Occurring In The Statistical Analysis of Probablistic Functions of Markov Chains,” The Annals of Mathematical Statistics, vol.41, no.1, pp.164-171, 1970.

[4] J. A. Bilmes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,” U.C. Berkeley, TR-97-021, 1998.

[5] F. Jelinek, “Continuous speech recognition by statistical methods,” Proc. IEEE, vol. 64, pp. 532-536, Apr. 1976.

[6] R. Bakis, “Continuous speech word recognition via centisecond acoustic states,” in Proc. ASA Meeting (Washington, DC), Apr. 1976.

[7] Z. Ghahramani and G. E. Hinton, “Parameter estimation for linear dynamical systems,” Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[8] J. Martens, “Learning the linear dynamical system with ASOS,” In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 743750, 2010.

[9] T. Minka, “From Hidden Markov Models to Linear Dynamical Systems,” Technical report, MIT, 1999.

[10] Y. Matsuyama, “The alpha-EM algorithm: Surrogate likelihood maximization using alpha-logarithmic information measures,” IEEE Trans. on Inform. Theory, vol. 49, pp. 692-706, 2003.

[11] Y. Matsuyama, “Hidden markov model estimation based on alpha-EM algorithm: Discrete and continuous alpha-HMMs.” International Joint Conference on Neural Network, San Jose, Vol. 7, No. 5, pp. 808-816, 2011.

[12] J. Pearl, “Probabilistic Reasoning in Intelligent Systems,” Morgan Kaufmann, CA, 1988.

[13] 北 研二, “確率的言語モデル”, 言語と計算機-4, 東京大学出版会, 1990.

[14] 村上仁一, “Baum-Welch アルゴリズムの動作と応用例,” 電子情報通信学会 基礎・境界ソサイエティ Fundamentals Review 4.1, p48-56, 2010.

[15] 木村陽一, “ベイジアンネットワーク：入門からヒューマンモデリングへの応用まで,” 日本行動計量学会第 7 回春のセミナー, 2004.

状態 1		状態 2		状態 4		状態 5		状態 7		状態 8		状態 10		状態 11	
記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率	記号	出力確率
i	0.062762	d*	0.060416	i	0.018346	d*	0.057469	d*	0.016496	d*	0.175784	the	0.040809	ax	0.569711
d*	0.049058	m	0.01977	have	0.015955	m	0.016603	the	0.015919	m	0.052729	d*	0.030839	d*	0.079578
the	0.01858	ax	0.007514	articl	0.01339	ax	0.012733	a	0.015248	w	0.037476	to	0.026202	max	0.040861
in	0.015478	w	0.00734	is	0.012814	q	0.006919	is	0.011258	q	0.025821	a	0.022156	q	0.022929
m	0.01527	p	0.006746	the	0.011544	p	0.006531	i	0.009895	g	0.025756	and	0.018478	m	0.012824
it	0.014036	i	0.006585	you	0.008856	s	0.005989	to	0.008536	r	0.025511	i	0.017097	p	0.011636
if	0.011893	q	0.00646	can	0.008438	w	0.005982	have	0.007819	p	0.020051	of	0.014625	r	0.008002
you	0.006778	z	0.0061	d*	0.007841	l	0.005977	t	0.006844	s	0.019718	is	0.013357	g	0.007285
thi	0.006742	u	0.006085	it	0.007213	i	0.005489	it	0.005726	v	0.018015	it	0.01244	n	0.006767
q	0.006584	a	0.005868	anyon	0.006669	b	0.005225	use	0.00534	z	0.016533	for	0.011847	a	0.005092
w	0.006365	c	0.005712	am	0.006455	n	0.005137	that	0.005038	t	0.016399	in	0.011748	pl	0.004636
a	0.005351	x	0.005683	s	0.005971	r	0.004743	of	0.004508	u	0.015549	window	0.011153	l	0.003586
is	0.005011	r	0.004913	there	0.005664	h	0.004667	you	0.004354	k	0.015364	that	0.01079	v	0.003574
l	0.004353	l	0.004864	a	0.005382	a	0.004647	know	0.004179	i	0.015156	with	0.008393	w	0.003441
doe	0.004253	h	0.00474	are	0.004635	c	0.004551	not	0.003671	b	0.015123	on	0.008205	s	0.002993
and	0.003846	e	0.004666	m	0.004517	y	0.004542	be	0.003512	x	0.014816	use	0.00736	b	0.002958
h	0.003822	d	0.004375	don	0.004358	e	0.004357	ani	0.00342	c	0.014417	have	0.00728	i	0.002852
there	0.003792	v	0.004176	thi	0.003967	z	0.004332	like	0.003326	o	0.012645	you	0.007212	z	0.002802
p	0.003699	o	0.004092	use	0.003948	x	0.004179	in	0.003146	n	0.011885	but	0.006426	f	0.002776
r	0.00364	k	0.003997	will	0.003906	o	0.004076	can	0.002874	a	0.011595	file	0.006359	u	0.00251

表 7　comp.os.ms-windows.misc の記号出力分布