情報処理学会論文誌　データベース　Vol. 4　No. 4　48–62 (Dec. 2011)

隠れマルコフモデルに基づくストリーム処理

松原靖子[†1]　櫻井保志[†2]　吉川正俊[†1]

近年，データストリーム処理に関する研究がさかんに行われている．本論文は，隠れマルコフモデル（HMM）に基づき，与えられた問合せモデルの特徴に類似した部分シーケンスをデータストリームから検出することを目的とする．HMM はシーケンスを確率密度関数に従う生成モデルを有する状態の遷移として表現するデータモデルであり，様々な分野で広く使われているが，主として有限長の蓄積データに用いられてきた．しかし，ネットワーク分析，センサ監視等，データ量が多く緊急性が要求されるような近年のアプリケーションでは，すべてのデータを蓄積してから処理することが困難である．本研究では，このような問題を解決する手法である *StreamScan* を提案する．さらに，理論的な分析を行い，精度を犠牲にしないにもかかわらず計算コストがデータストリームの長さに依存せず一定であることを証明する．様々な実データを用いた実験を行い，*StreamScan* がデータストリームから正確に部分シーケンスを検出し，そして従来の手法と比較して大幅な性能向上を達成していることを明らかにした．

Stream Processing through Hidden Markov Models

Yasuko Matsubara,[†1] Yasushi Sakurai[†2]
and Masatoshi Yoshikawa[†1]

Data stream processing has recently attracted an increasing amount of interest. Our aim is to monitor data streams, and find subsequences that have the characteristics of a given Hidden Markov Model (HMM). A lot of research effort has concentrated on pattern discovery for HMMs, and it has been studied for finite, stored sequence sets. However, in many applications such as sensor monitoring, massive amounts of data arrive continuously and it is infeasible to save all the historical data. We propose *StreamScan*, a novel algorithm that can solve the problem. We provide a theoretical analysis and prove that *StreamScan* guarantees the exactness of the output, while it requires *constant space and time* per time-tick. These are significant improvements over the alternative solution. Our experiments on real data illustrate that *StreamScan* does indeed detect the qualifying subsequences correctly and that it can offer great improvements in speed over the alternative method.

1. はじめに

金融，ネットワーク監視，モバイルサービス，センサネットワーク管理等，データストリーム処理の応用は多岐にわたる．このため，計算理論，データベース，データマイニング，ネットワーク等様々な分野で研究がさかんに行われている[11),12),23),33)]．これらの応用の中で，特に重要な要素技術として時系列データストリームの監視技術があげられ，ストリーム監視の高速化は重要かつ非常に挑戦的な研究課題である．データストリームは高いビットレートで送信され，そのデータサイズは限りなく大きく，実用においてはすべてのデータをメモリ空間に格納できない．そこでデータストリームにおいては，1 度データを処理すると破棄し，2 度とそのデータを使う必要がないようなアルゴリズムが求められる．

ストリーム監視では，部分シーケンスマッチングのメカニズムが必要とされるが，それはノイズに対してロバストでなければならない．また，各々のデータストリームのサンプリングレートが異なる場合や周期が変化する場合もあるため，シーケンスマッチングにおいては時間軸方向の調整を可能とすることが望ましい．

隠れマルコフモデル（HMM: Hidden Markov Model）は与えられたシーケンスを確率密度関数に従う生成モデルを有する状態の遷移として扱うデータモデルであり，ノイズに強いという特長がある．このため HMM は医療データ解析[25)]や遺伝子解析[21)]等，多くのアプリケーションにおいて使用されている．

応　用　例

本論文では，HMM を用いたデータストリーム監視の問題を扱う．HMM は時系列パターンの統計的特徴を表現するための最も有用な技術の 1 つである．**図 1** を用いて本論文で扱う問題の説明を行う．図 1 は，モーションキャプチャデータにおける連続するモーションの動きの様子を示している．ここで，これらの 5 つのモーションの組合せで表現される 1 つのシーケンスを考える．本研究で解決したい問題は，このシーケンスの中から，特定のモーション（たとえば walking）のみを発見することである．提案するアルゴリズムは，与えられた HMM のモデル（walking を表現するモデル等）に対して，そのモデルの特徴に類似

†1 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University
†2 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories, NTT Corporation

図 1　モーションキャプチャデータを対象としたパターンの発見
Fig. 1　Pattern discovery from the motion capture dataset.

した部分シーケンスを検出することができる．ここで重要となるのが，我々の提案するアルゴリズムは，HMM の従来の研究とは異なり，データストリーム監視の出力結果の厳密性を保証することができる点である．すなわち提案アルゴリズムは，精度を犠牲にすることなく，適切な部分シーケンスを正しくすべて出力することができる．我々のアルゴリズムは，ストリーム状況下において連続的に動作するだけでなく，計算コストにおいても従来手法と比較し大幅な性能向上を実現している．提案手法の計算コストは，データストリームの長さに依存せず一定である．

HMM に基づくデータストリームの監視は，様々なアプリケーションに適応できる．本節では，*StreamScan* の有用性を示すために，いくつかの領域とそのアプリケーションの例について紹介する．

- **マルチメディア**：マルチメディア分野におけるマルチモーダルデータのマイニングは，様々な形で研究がさかんに行われている[13),19)]．マルチメディアデータは，デジタル画像，オーディオ，ビデオ，テキストデータ等，様々なものがあり，これらのデータのマイニングは非常に有益である．マルチメディア分野における分布データ探索の例として，モーションキャプチャにおけるヒトの動きの検出が考えられる．体の各所の時間ごとの運動エネルギーの値は，多次元の時系列データに変換することができ，これらの時系列データを用いて類似探索を行うと，アノテーションやそのほかのメタデータを用意することなく類似モーションを発見することができる．他の例として，ヒトの声の特徴を抽出することでユーザの判別問題を行う話者認識があげられる．テレフォンバンキングを初めとするセキュアなシステムにおいて，音声パターンを利用した個人認証等を行うことは，非常に重要なタスクである．
- **医療データ解析**：大量の医療データの中から有益な情報を抽出することは，多くの医療領域において研究の中心的な課題となっている[25),26)]．たとえば，脳波（EEG）のデータを用いたメンタルタスクの探索問題は，ヒトの脳機能の理解に非常に有益であり，様々な取り組みがなされている[32)]．
- **遺伝子解析**：遺伝子情報の解析において最も重要な貢献は，異なる有機体の遺伝子配列に関連性があることを発見したことである．異なる種において類似した遺伝子が保たれており，それらの遺伝子が非常によく似た機能を持っていることが分かっている[21)]．このことにより，生物情報学において遺伝子配列におけるシーケンスマッチング問題は最も重要なタスクの 1 つである[5)]．また，判別問題も遺伝子の機能を知るうえで大変有用なタスクとなっている．未知の遺伝子配列を他の既知の配列と比較し，類似した機能を持つ遺伝子を発見することで，その未知の遺伝子の機能や構造，生化学的な活動を推定することができる．

提案手法の貢献

本研究の貢献は次のとおりである．

(1)　本論文では，データストリームにおいて部分シーケンスマッチングを行うための新手法である *StreamScan* を提案する．本手法は，高速，正確，省メモリでデータストリームを処理することができ，ストリームの過去のデータにさかのぼることなく逐次的に解析を行う．また，メモリ消費量と単位時間あたりの計算コストは，データストリームの長さに依存せず一定である．

(2)　理論的な分析を行い，*StreamScan* が厳密性を保証しながら（つまり，精度を犠牲にすることなく），最適な部分シーケンスを検出することを証明する．

(3)　実データを用いた実験を行い，*StreamScan* が最適な部分シーケンスを正しく発見し，また，従来の手法と比較し約 500,000 倍まで性能向上を達成していることを示した．

本論文の構成は次のとおりである．2 章では，関連研究をあげる．3 章では，本論文における問題設定を示した後に，データストリームの監視をするための提案手法について説明する．4 章では，*StreamScan* の制度と計算量について議論する．5 章では実験結果について検証し，*StreamScan* の有効性を示す．6 章は本論文のまとめである．

2. 関 連 研 究

関連研究は大きく 2 種類に分類することができる．第 1 のグループは HMM に関するものであり，第 2 のグループはデータストリームに関するものである．

2.1 隠れマルコフモデル（HMM: Hidden Markov Model）および時系列マッチング

隠れマルコフモデル（HMM）は，遺伝子解析[5] や医療データ解析[32] をはじめとする様々な分野で利用されている技術である．さらに音声認識の分野において，HMM の探索問題は大きな研究テーマの 1 つである[10]．連続型 HMM の状態は典型的に 8〜64 個のガウス関数で構成され，尤度計算をするにはそれぞれのガウス関数を別々に計算しなければならない．そのため計算コストは高くなる．計算コストを落とすために Hunt らは線形判別分析によりガウス関数の数を減らす手法を提案した[16]．また尤度がすでに計算されたガウス関数の部分セットのみを用いる手法も示されている[4]．Sagayama らは連続 HMM を離散型 HMM に置き換える手法を提案した[27]．離散型 HMM の尤度はスカラ量子化された確率のテーブルを引くことで計算できる．これらの研究は本論文において提案する手法と併用することにより，より効果的かつ高速な探索が可能となる．

Beam search アルゴリズムは，Viterbi アルゴリズムをはじめとする動的計画法を用いた手法の高速化を実現した技術であり，様々な研究が行われている[7),22]．ここで，Viterbi アルゴリズムにおける Beam search の基本的なアイディアを説明する．最も尤度の高いパスと比較して，尤度が十分に低い状態変数へのパスは，最適なパスになりえないという性質を利用し，このような尤度の低いパスを不要のものとして枝刈りする．しかしここで問題となるのが，この手法は厳密性を保証するものではないため，最適解を失う可能性があるということである．

HMM の応用として，音声認識[1),2),8)]，遺伝子解析[15]，情報抽出[9),30),31)]，不正検出[3]，手書き文字認識[14] 等があげられる．特に，HMM を用いた部分シーケンスマッチング手法は，動作認識の分野でも取り組まれている[6),17),20)]．しかしこれらの分野におけるすべての研究は，我々の提案するストリーミング処理を行うものではない．これに対して提案手法は，尤度だけでなく，部分シーケンスの位置についても，精度をいっさい犠牲にすることなく，そしてストリームの過去のデータにさかのぼることなく検出する．また，部分シーケンスマッチングに関しては，主として DTW に基づく音声処理のために，連続 DP（Dynamic Programming）と呼ばれる手法が提案されている．これは過去にさかのぼらず，入力フレームに対する計算のみで累積距離を出力することを可能とする[34),35)]．これに対して本論文では，HMM に基づいてシーケンスマッチングを行うために，長さで正規化した尤度を閾値として用い，部分シーケンスを検出する．またその際，上記で述べたように，尤度計算の誤差がないことを保証するだけでなく，過去にさかのぼることなしに部分シーケンスの位置についても厳密に特定することができる．

2.2 データストリームによるパターン発見

HMM に基づくデータストリームの監視に関するマイニング技術はあまり取り組まれていないが，ここでは関連するトピックとして，データストリームのパターン発見，要約，データ圧縮等の技術について言及する．データストリームにおけるパターン検出の研究としては，L^p 距離に基づくストリームのトレンド検出や相関検出，予測をはじめとする様々な手法が提案されている．たとえば，Gilbert らはウェーブレット変換のためのインクリメンタルなアルゴリズムを提案した[11]．Zhu らは，リアルタイムでのストリーム監視に焦点を合わせ，StatStream を提案している[33]．SPIRIT はデータストリームから相関とトレンドに相当する隠れ値を検出する問題に取り組んだものである[24]．BRAID はデータストリーム間の遅延相関を検出するための手法である[29]．文献 28) では DTW に基づき，複数数値ストリームを効率的にモニタリングする手法を提案しており，ウィンドウサイズを指定することなく一定の計算コストで部分シーケンスを特定する．しかしながら，これらの手法のどれも，我々が述べた問題に取り組んでいるものではない．

3. 提 案 手 法

本章では，シーケンスマッチングを行うための HMM について説明し，その後本論文で扱う問題を定義し，さらにその解決法を述べる．本論文においては，例をあげる際に離散出力の全結合型 HMM を示しているが，提案手法は連続出力，そして left to right モデルに対しても適用可能である．

本論文における主な記号の定義は**表 1** のとおりである．

3.1 隠れマルコフモデル

隠れマルコフモデル（HMM: Hidden Markov Model）は不確定な時系列のデータをモデル化するための有効な統計的手法である．HMM はノイズにロバストであり，話者認識や自然言語処理，たんぱく質や DNA を含む遺伝子列解析等多くのアプリケーションにおいて使用されている．HMM は，初期状態確率 $\boldsymbol{\pi}=\{\pi_i\}$，状態遷移確率 $\mathbf{A}=\{a_{ij}\}$，シンボル出力確率 $\mathbf{B}=\{b_i(v)\}$ から構成される．

表 1　主な記号とその定義
Table 1　Symbols and definitions.

Symbol	Definition
X	長さ n のデータシーケンス/ストリーム
x_t	X の t 番目の値/要素（$t=1,\ldots,n$）
$X[t_s:t_e]$	t_s から t_e までの X の部分シーケンス
k	モデルの状態数
$\boldsymbol{\Theta}$	モデルのパラメータ集合
$\boldsymbol{\pi}=\{\pi_i\}$	状態 i の初期状態確率
$\mathbf{A}=\{a_{ij}\}$	状態 i から状態 j への状態遷移確率
$\mathbf{B}=\{b_i(v)\}$	状態 i におけるシンボル v の出力確率
ϵ	適合する部分シーケンスを発見するための閾値
μ	適合する部分シーケンスの長さの閾値
$P(X,\boldsymbol{\Theta})$	$\boldsymbol{\Theta}$ のもとでの X の尤度関数
$p_i(t)$	要素 (t,i) における尤度/確率（i.e., 時刻 t における状態 i）
$c_i(t)$	要素 (t,i) における累積尤度
$s_i(t)$	要素 (t,i) における開始点

HMM において重要な処理は，モデルの中の隠れ状態から最も確率が高くなるような状態遷移の列を取り出すことであり，これは Viterbi アルゴリズムによって行われる．Viterbi アルゴリズムは動的計画法によってシーケンス X の尤度を推定する．ここで，確率を最大化するような状態推移の列は，Viterbi パスと呼ばれる．モデル $\boldsymbol{\Theta}$ とシーケンス X が与えられたとき，これらの尤度 $P(X,\boldsymbol{\Theta})$ は Viterbi アルゴリズムにより次のように計算できる．

$$P(X,\boldsymbol{\Theta})=\max_{1\le i\le k}\{p_i(n)\} \tag{1}$$

$$p_i(t)=\begin{cases}\pi_i b_i(x_1) & (t=1)\\ \max_{1\le j\le k}\{p_j(t-1)a_{ji}\}b_i(x_t) & (2\le t\le n)\end{cases}$$

ここで，$p_i(t)$ は，時刻 t における状態 i の最大確率を示す．

Viterbi アルゴリズムでは，**図 2** のように，状態を縦軸に，時間を横軸に並べたときに構成されるトレリス構造に基づき尤度を計算する．ここで，尤度は，各状態における確率の最大値を動的計画法によって求められる．図 2 において，太線で示した状態の遷移は Viterbi パスである．

例 1　次にあげるモデルとシーケンスを考える．

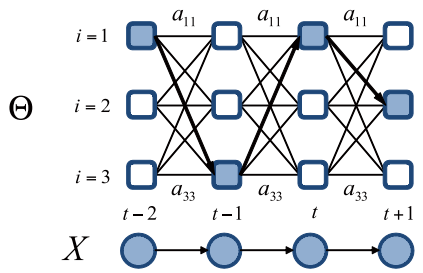


図 2　トレリス構造の様子
Fig. 2　Illustration of the trellis structure.

$$\pi=\begin{bmatrix}1\\0\\0\end{bmatrix},\mathbf{A}=\begin{bmatrix}0.5 & 0.5 & 0\\ 0.5 & 0.25 & 0.25\\ 0 & 0 & 1\end{bmatrix},$$

$$\mathbf{B}=\begin{bmatrix}1 & 0 & 0\\ 0.75 & 0.25 & 0\\ 0 & 0 & 1\end{bmatrix},X=(1,1,2,3).$$

Viterbi アルゴリズムは以下のように計算される．

$$\begin{array}{llll}p_1(1)=1, & p_1(2)=0.5, & p_1(3)=0, & p_1(4)=0\\ p_2(1)=0, & p_2(2)=0.75\cdot 0.5, & p_2(3)=(0.5)^2\cdot 0.25, & p_2(4)=0\\ p_3(1)=0, & p_3(2)=0, & p_3(3)=0, & p_3(4)=(0.5)^2\cdot(0.25)^2.\end{array}$$

確率を最大化する状態推移は (u_1,u_1,u_2,u_3) であり，結果として尤度は

$$P(X,\boldsymbol{\Theta})=(0.5)^2\cdot(0.25)^2$$

となる．

Viterbi アルゴリズムは一般に，$O(nk^2)$ の計算量がかかる．これはトレリス構造が nk 個の要素で構成されており，さらに k 個のすべての状態において，前の時刻の状態すべてから遷移する確率を計算するからである．ただしここで，メモリ使用量は $O(k)$ である．これは，アルゴリズムがトレリス構造の中の 2 列（すなわち，現在の列と直前の時刻の列）のみを用いて尤度計算を行うためである．

3.2 問題定義

データストリーム X は，$x_1, x_2, \ldots, x_n, \ldots$ の値からなる半無限長のシーケンスである．ここで x_n は最も新しい値であり，時刻が進むごとに n は増加する．$X[t_s : t_e]$ を t_s から t_e までの部分シーケンスとする（$1 \leq t_s \leq t_e \leq n$）．本論文の目的は，与えられたモデル $\mathbf{\Theta}$ の特徴と高い類似性を有する（つまり，尤度 $P(X[t_s : t_e], \mathbf{\Theta})$ が高い値となる）部分シーケンス $X[t_s : t_e]$ を発見することである．より具体的には，次の条件を満たす部分シーケンスを検出する．

$$P(X[t_s : t_e], \mathbf{\Theta}) \geq \epsilon^l \tag{2}$$

ここで，l は $X[t_s : t_e]$ の長さを示す（つまり $l = t_e - t_s + 1$）．尤度は，トレリス構造内の状態確率の積で表現されるため，部分シーケンスの長さが長くなるにつれて，その値は減少する．そのため，類似判定の閾値も部分シーケンス長に従って指数関数的に減少させることが望ましいといえ，ϵ^l を用いて判定を行う．

しかしながら，実用上の問題として，オリジナルの Viterbi アルゴリズムでは，ノイズ等によってシーケンス長の短い，意味のない部分シーケンスを適合解として検出する可能性がある．そこで新たに，部分シーケンスの長さの閾値という概念を導入する．部分シーケンスの長さの閾値は，ユーザによって与えられる値であり，これにより，ユーザの真の要求を満たす最適な部分シーケンスを発見することができる．具体的には，我々の解決したい部分問題は次のように定義される．

問題 1（Subsequence query） 長さ n のシーケンス X，k 個の状態からなるモデル $\mathbf{\Theta}$，尤度の閾値 ϵ，そして部分シーケンスの長さの閾値 μ が与えられたとき，次の条件を満たすような部分シーケンス $X[t_s : t_e]$ を発見する．

$$P(X[t_s : t_e], \mathbf{\Theta}) \geq \epsilon^{l-\mu}. \tag{3}$$

ここで，適合する部分シーケンスの長さの最小値 μ はユーザによって与えられる．直観的に問題 1 は，部分シーケンスの長さの概念も含んでいる．つまり，我々は，μ 以上の長さの部分シーケンスのみを検出するとともに，μ によって，より長いシーケンスが出力されるような制約を加えることができる．

しかしここで，部分シーケンスの検出を問題設定に導入するにあたり考慮すべき問題がある．問合せ $\mathbf{\Theta}$ が X の部分シーケンスに適合する場合，極大値をとる区間と重複（overlap）する他の多くの部分シーケンスも適合してしまう．このような余分な部分シーケンスは，次のように 2 重に害を及ぼす．(a) 利用者に冗長な情報を与えて悩ませることになる．(b) 不必要な結果についても報告させるために，アルゴリズムの処理速度を低下させる．したがって本論文では，このような余分な部分シーケンスを除外するために，標準的な範囲問合せにもう 1 つの条件を追加することを提案する．

具体的には，Viterbi パスが重なっているような部分シーケンスを，重複するマッチングであると定義する．つまり，トレリス構造の中の少なくとも 1 つの要素が共有されていれば，それらの部分シーケンスは重複であるとする．これ以降，極大値をとる部分シーケンスについて言及する際，これを最適な部分シーケンスと表現する．本論文において最終的に解決したい問題を以下のように定義する．

問題 2（Optimal-match query） データストリーム X（つまり，長さ n の時系列データ）と，k 個の状態からなるモデル $\mathbf{\Theta}$，閾値 ϵ，μ が与えられたとき，次の条件を満たす部分シーケンス $X[t_s : t_e]$ をすべて検出する．

(1) 部分シーケンスがモデル $\mathbf{\Theta}$ に対して十分に適切である．すなわち，$P(X[t_s : t_e], \mathbf{\Theta}) \geq \epsilon^{l-\mu}$，

(2) 複数の重複するマッチングの中から，極大値のみを報告する，すなわち，(1) の条件を満たす部分シーケンスのうち，$P(X[t_s : t_e], \mathbf{\Theta}) \cdot \epsilon^{\mu-l}$ が最大となるもののみを報告する．

さらに，本研究ではストリーム処理を指向している．最適な部分シーケンスを検出することを保証し，かつ可能な限り早くそれを出力する．

3.2.1 Standard-Viterbi

シーケンス X とモデルパラメータ $\mathbf{\Theta} = \{\pi, \mathbf{A}, \mathbf{B}\}$ が与えられたとき，我々の目的は，$\mathbf{\Theta}$ の特徴を持つような X の部分シーケンスを発見することである．最適な問合せマッチングの問題（問題 2）において，ユーザが理論的な保証をしながら（精度を犠牲にせずに）最適解を出したいと要求した場合，最も素直な方法は，すべての部分シーケンス $X[t_s : t_e]$（$1 \leq t_s \leq t_e \leq n$）を考えて，一般的な尤度計算をそれぞれの部分シーケンスに対し適用することであり，これには $O(n^2)$ 個のトレリス構造を必要とする．時間計算量は $O(n^3k^2)$（もしくは，単位時間あたり $O(n^2k^2)$）となる．この方法では，計算コストが高いだけではなく，ストリーム環境に適用することができない．

最善ではないが，これよりも好ましい方法を以下に述べる．最適な部分シーケンス $X[t_s : t_e]$ を見つけるために，**図 3** のように，すべての時刻 t から始まるような $O(n)$ 個のトレリス構造を用意し，これらに対して Viterbi アルゴリズムを適用する．これにより，$P(X[t_s : t_e], \mathbf{\Theta}) \cdot \epsilon^{\mu-l}$

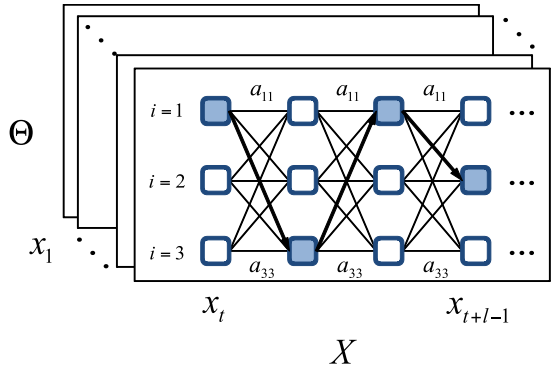


図 3　HMM における部分シーケンスの検出の様子．色のついた状態は，トレリス構造における Viterbi パスを示す．*Standard-Viterbi* は毎時刻生成されるトレリス構造をすべて更新する必要がある

Fig. 3　Illustration of subsequence identification with HMMs. The shaded states denote the Viterbi path in the trellis structure. *Standard-Viterbi* has to maintain the structures starting from every time-tick.

を最大化するような部分シーケンスを見つけ出す．この方法は，探索もれがないことを保証することができ，我々はこの手法を今後 *Standard-Viterbi* と表現する．

s 番目のトレリス構造（すなわち時刻 s から始まるトレリス構造）において，時刻 t の状態 i の確率を $p_{s,i}(t)$ とする．X と $\mathbf{\Theta}$ が与えられたうえでの部分シーケンスマッチングの最大尤度は，以下のように求めることができる．

$$P(X[t_s:t_e],\mathbf{\Theta}) = \max_{1\le i\le k}\{p_{t_s,i}(t_e - t_s + 1)\} \tag{4}$$

$$p_{s,i}(t) = \begin{cases} \pi_i b_i(x_t) & (t=s) \\ \max_{1\le j\le k}\{p_{s,j}(t-1)a_{ji}\}b_i(x_t) & (s<t\le t_e) \end{cases}$$

$$(s=1,\ldots,n; t=1,\ldots,n-s+1; i=1,\ldots,k).$$

この手法は $O(n)$ 個のトレリス構造を必要とするため，毎時刻 $O(nk^2)$ 個の確率を更新する必要がある．次からの節では，精度をまったく犠牲にすることなくこれを劇的に性能向上させる方法を示す．

3.3　基本的なアイディア

我々の提案手法は，次にあげる 2 つのアイディアに基づいている．

3.3.1　累積尤度関数

Standard-Viterbi は尤度の計算のために，新たなトレリス構造を毎時刻作成する．このため全体で $O(n)$ 個のトレリス構造が必要となる．そこで我々はこの手法を用いる代わりに，累積尤度の概念を導入する，これを用いることにより，単一のトレリス構造のみを用いての尤度計算を実現する．

アプローチ 1　X の最適な部分シーケンスを発見するために，尤度の累積値を計算する累積尤度関数を導入する．このとき尤度計算には，1 つのトレリス構造しか必要としない．

Standard-Viterbi が最大尤度を求めるために単位時間あたり $O(nk^2)$ 個の尤度を更新するのに対し，累積尤度は $O(k^2)$ 個の値しか更新しないため，時間とメモリ量の大幅な低減化につながる．さらに，後に述べるように（補助定理 1），この累積尤度関数は X の最適部分シーケンスを正しく出力することが保証されている．

3.3.2　拡張トレリス構造

部分シーケンスマッチングの最大尤度を得るために，累積尤度関数を提案したが，このアイディアのみでは問題を解決することはできない．ここで別の問題がある．単に累積尤度関数のみを用いた場合，最適な部分シーケンスの開始点に関する情報を失ってしまう．すなわち，1 つのトレリス構造をスキャンすることによって最大尤度を得ることができるが，スキャンの後，どの部分シーケンスが最大尤度を出力したのかを判断することができなくなる．

そこで，第 2 のアイディアである拡張トレリス構造を用いる．これは，各々の候補部分シーケンスの開始点に関する情報を扱うことができるようにトレリス構造を改良したものである．ここで，トレリス構造の時刻 t における状態 i の値を $c_i(t)$ であるとする．すなわち，$c_i(t)$ は，X の t 番目の値とモデル $\mathbf{\Theta}$ の状態 i との間の最大累積尤度を示すものとする（i.e., $t=1,\ldots,n$; $i=1,\ldots,k$）．そしてさらに，我々の提案する拡張トレリス構造は，$c_i(t)$ に対応する開始点である $s_i(t)$ も記録する．いい換えれば，$s_i(t)$ から t までの部分シーケンスの累積尤度が $c_i(t)$ に対応する．拡張トレリス構造の計算の具体例は後に示す（図 6）．

アプローチ 2　拡張トレリス構造は，各々の部分シーケンスの確率/尤度の値と，開始点を保持する．したがって適合する部分シーケンスをストリーム処理によって認識することができる．

拡張トレリス構造は，累積尤度の値だけでなく，尤度に付随する開始点も更新する．我々の手法は次のような特長を持つ．(1) 拡張トレリス構造を用いることにより，どの部分シーケンスが最大尤度を出力したのかをストリーム処理の間も認識することができる．さらに重要なこととして，(2) 累積尤度関数は，最適な部分シーケンスの尤度関数に関して可逆で

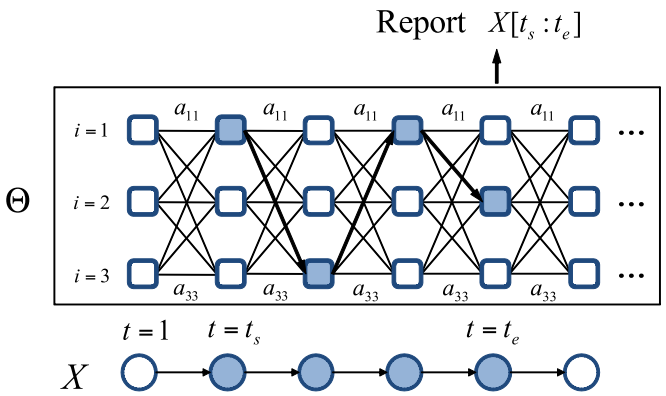


図 4　*StreamScan* による問合せ処理．*StreamScan* は単一のトレリス構造のみを用いて最適な部分シーケンスを検出する

Fig. 4　Illustration of *StreamScan*. *StreamScan* uses only a single trellis structure to capture all qualifying subsequences.

ある．

3.4　StreamScan

本節では，3.2 節で述べた問題を解決するためのアルゴリズムを提案する．*StreamScan* は，問合せモデルに対して適切な部分シーケンスをデータストリームから検出する手法である．**図 4** は，どのように *StreamScan* が適切な部分シーケンスを検出するのかを示している．図のように，*StreamScan* は，X と $\mathbf{\Theta}$ の拡張トレリス構造を扱う．この構造の中で，各要素 (t, i)（つまり X の t 番目の値と $\mathbf{\Theta}$ の状態 i）は累積尤度と開始点の両方を保持している．最適解の問い合わせの際には，*StreamScan* は適合するすべての部分シーケンスを出力する．アルゴリズムを示す前に，ここで累積尤度関数と拡張トレリス構造の詳細について述べる．

長さ n のシーケンス X が与えられたとき，$X[t_s : t_e]$ の累積尤度 $C(X[t_s : t_e], \mathbf{\Theta})$ は次のようにインクリメンタルに計算される．

$$C(X[t_s : t_e], \mathbf{\Theta}) = c_{max}(t_e) = \max_{1 \le i \le k} \{c_i(t_e)\} \tag{5}$$

$$c_i(t) = \max \begin{cases} \pi_i b_i(x_t) \cdot \epsilon^{-1} \\ \max_{1 \le j \le k} \{c_j(t-1) a_{ji}\} b_i(x_t) \cdot \epsilon^{-1} \end{cases}$$

$(t = 1, \ldots, n; i = 1, \ldots, k)$.

同様に，部分シーケンスの開始点もインクリメンタルに更新することができる．

$$s_i(t) = \begin{cases} (t, i) & (c_i(t) = \pi_i b_i(x_t) \cdot \epsilon^{-1}) \\ s_j(t-1) & (c_i(t) \neq \pi_i b_i(x_t) \cdot \epsilon^{-1} \\ & \quad \wedge c_j(t-1) = c_{max}(t-1)). \end{cases} \tag{6}$$

最適な Viterbi パスは，尤度計算によって得ることができ，最適な部分シーケンスの開始点は Viteri パス上を伝播していく．また，累積尤度関数は最適な部分シーケンスの尤度に関して可逆である．最適な部分シーケンスの尤度は，累積尤度の値と部分シーケンスの長さから，次のようにして得ることができる．

$$P(X[t_s : t_e], \mathbf{\Theta}) = C(X[t_s : t_e], \mathbf{\Theta}) \cdot \epsilon^l \tag{7}$$

ここで l は部分シーケンスの長さである（つまり，$l = t_e - t_s + 1$）．

3.4.1　アルゴリズム

StreamScan は，次にあげる 2 つの特長を持つ．(a) 問題 2 の 2 つ目の条件を満たす部分シーケンスをもれることなくすべて検出することを保証する．(b) すべての適切な解を，可能な限り早く報告する．我々の提案する *StreamScan* を**図 5** に示す．図 5 に示すように，時刻 t において x_t が到着するたびに，累積尤度 $c_i(t)$ をインクリメンタルに更新し，開始点 $s_i(t)$ を $c_i(t)$ に基づいて決定する．

候補集合 $\mathcal{S}$ は，異なる開始点を持つような最適部分シーケンスの解の候補が複数個格納されており，すべての解候補の情報を保持している．ここで，解候補 (c, s, e) とは最適部分シーケンスの候補を示しており，累積尤度 c，開始点 s，終点 e の情報で構成される．区間が重複する部分シーケンスが存在する場合，累積尤度はその最大値である c のみが格納される．*StreamScan* は，s が次の条件を満たしたとき，その部分シーケンスを最適解として報告する．

$$\forall_i, s \neq s_i(t) \tag{8}$$

これは，候補部分シーケンスが今後出現する部分シーケンスによって置き換わることがないことを意味する．つまり，今後出現する候補部分シーケンスは必ず，現在報告された候補部分シーケンスと重複することがないことを示す．図 5 に示しているように，出力する部分シーケンスの尤度 p は，累積尤度とシーケンス長から簡単に求めることができ，最終的に解をストリーム状況下で出力する．

ここで**図 6** を用いてアルゴリズムがどのように動作するのかについて具体的に説明する．

Algorithm StreamScan (x_t)
for $i = 1$ to k **do**
  compute $c_i(t)$; // cumulative likelihood derived by Eq. (5)
  compute $s_i(t)$; // starting position derived by Eq. (6)
  $e_i(t) := (t, i)$; // end position
  if $c_i(t) \geq \epsilon^{-\mu}$ **then**
    if $s_i(t) \in \mathcal{S}$ **then**
      pick up an entry (c, s, e) such that $s = s_i(t)$, from the candidate set $\mathcal{S}$;
      // update the maximum cumulative likelihood and end position
      if $c_i(t) \geq c$ **then**
        $c := c_i(t)$;
        $e := e_i(t)$;
      end if
    else
      // add the subsequence into $\mathcal{S}$
      add $(c_i(t), s_i(t), e_i(t))$ to $\mathcal{S}$;
    end if
  end if
end for
// report the optimal subsequence
for each entry $(c, s, e) \in \mathcal{S}$ **do**
  if $\forall_i, s \neq s_i(t)$ **then**
    // compute the likelihood of the subsequence
    $p = c \cdot \epsilon^l$;
    report (p, s, e);
    remove the entry from $\mathcal{S}$;
  end if
end for

図 5　ストリーム処理アルゴリズム *StreamScan*
Fig. 5　Streaming algorithm, *StreamScan*.

例 2　閾値を $\epsilon = 0.1$，$\mu = 3$ と設定し，次のモデルとシーケンスを考える．

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0.25 & 0.25 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0.75 & 0.25 & 0 \\ 0 & 0 & 1 \end{bmatrix}, X = (1, 1, 1, 2, 3, 3, 3, 1).$$

要素 (t, i) には，累積尤度 $c_i(t)$ と開始点 $s_i(t)$ が含まれている．色のついた要素は最適なパスを示す．時刻 $t = 6$ において，尤度 $c_3(6) = 1562.5$ となる部分シーケンス $X[2:6]$ を検出する．ここで，尤度は $\epsilon^{-\mu}$ よりも大きな値であるが，$X[2:6]$ を最適な部分シーケンスとして出力しない．なぜなら，これから出現する部分シーケンスが最適な部分シーケンスとして置き換わる可能性があるためである．時刻 $t = 7$ において，最適な部分シーケンス $X[2:7]$ を検出する．そして，$X[2:7]$ を時刻 $t = 8$ において出力する．それは，これから出現するものは最適な部分シーケンスになる可能性がないことを確認できたためである．

4. 理論的な分析

本章では，理論的な分析を行い，*StreamScan* の精度と計算量について述べる．

4.1 精　　度

補助定理 1　シーケンス X とモデル $\mathbf{\Theta}$ が与えられたとき，問題 2 は次の条件と等価である．

(1)　$C(X[t_s : t_e], \mathbf{\Theta}) \geq \epsilon^{-\mu}$

(2)　Viterbi パスが交差する部分シーケンスのグループの中で，$C(X[t_s : t_e], \mathbf{\Theta}) \cdot \epsilon^{\mu}$ が最大値をとる．

証明 1　$X[t_s : t_e]$ の Viterbi パスが，時刻 t_s，状態 m（つまりトレリス構造内の要素 (t_s, m)）から開始されるとする．式 (4)，(5) より，

$$\pi_m b_m(x_{t_s}) = p_{t_s,m}(t_s) = c_m(t_s) \cdot \epsilon.$$

さらに，もし Viterbi パスが，要素 $(t_e - 1, j)$ と (t_e, i) も含むならば，

$$a_{ji}\ b_i(x_{t_e}) = p_{t_s,i}(t_e) / p_{t_s,j}(t_e - 1) = c_i(t_e) \cdot \epsilon / c_j(t_e - 1).$$

時刻 t_e における状態 i において，

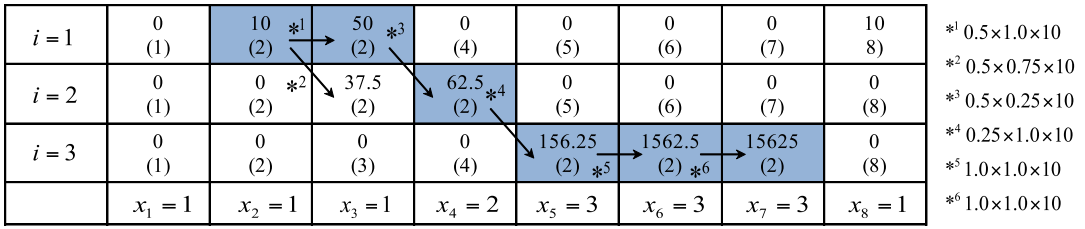


図 6　*StreamScan* の部分シーケンス検出の例．行列の各要素について，上の値は累積尤度の値を，括弧内の値は開始点を示している．色のついた要素は最適なパスを示している
Fig. 6　Illustration of proposed algorithm. The upper number shows the cumulative likelihood value in each element of the matrix. The number in parentheses shows the starting position. The shaded elements denote the optimal path.

$$p_{t_s,i}(t_e) = c_i(t_e) \cdot \epsilon^l.$$

これにより，

$$c_i(t_e) \geq \epsilon^{-\mu}.$$

問題 2 の第 2 の条件から，トレリス構造内の最適 Viterbi パスは，それぞれの部分シーケンスのグループにおける最大累積尤度を与えることが明らかである．よって，問題 2 と補助定理 1 の 2 つの条件は等価となる．□

補助定理 2　*StreamScan* は，探索漏れを発生させないことを保証する．

証明 2　C_s を，最適な部分シーケンス $X[t_s : t_e]$ の開始点とする．拡張トレリス構造において，Viterbi パスが重なるような最適な部分シーケンス群は，必ず同じ要素を共有しているため，開始点 C_s も一致する．もし $s_i(t) \neq C_s$ であれば，その部分シーケンスの Viterbi パスは最適なパスと重複しない．同様に，以下の条件を満たす場合，これから出現する部分シーケンスは候補集合の中の部分シーケンスと重複することはない．

$$\forall_i, C_s \neq s_i(t).$$

StreamScan は，上記の条件を満たすときのみ，部分シーケンス $X[t_s : t_e]$ を最適な解として報告する．よって，*StreamScan* が最適な部分シーケンスを見落とすことはない．補助定理 1 は，最適部分シーケンスの尤度が累積尤度の値から復元することができることを示している．これにより，*StreamScan* は，解の厳密性を保証する．□

4.2 計　算　量

X を伸長している長さ n のシーケンス，k をモデルの状態数とする．ここでは，ストリーム状況下において，トレリス構造を扱う際の計算量について議論する．

補助定理 3　*Standard-Viterbi* は，$O(nk)$ のメモリ量と単位時間あたり $O(nk^2)$ の計算時間を要する．

証明 3　*Standard-Viterbi* は，$O(n)$ 個のトレリス構造を扱い，単位時間あたり $O(nk^2)$ 個の値を更新する．したがって，$O(nk^2)$ の時間を要する．各トレリス構造において，k 個の状態の配列を 2 つ保持するため，全体では $O(nk)$ のメモリ空間が必要となる．□

補助定理 4　*StreamScan* は $O(k)$ のメモリ量と単位時間あたり $O(k^2)$ の計算時間を要する．

証明 4　*StreamScan* は，トレリス構造を 1 つしか保持しないため，毎時刻 $O(k^2)$ 個の値を更新する．したがって，*StreamScan* は単位時間あたり $O(k^2)$ の時間を要する．*StreamScan* は，単一のトレリス構造において，k 個の状態の配列を 2 つ保持するため，最終的に $O(k)$ のメモリ空間が必要となる．□

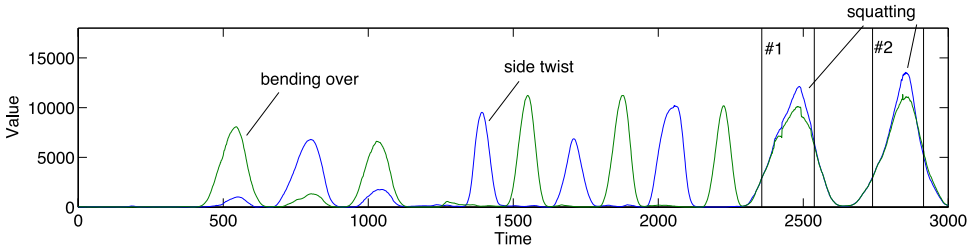
5. 評 価 実 験

StreamScan の有効性を検証するため，実データを用いた実験を行った．実験は 4 GB のメモリ，Intel Core 2 Duo 1.86 GHz の CPU を搭載した Linux のマシン上で実施した．なお，本実験では $k = 20$ に設定し，ϵ については 10^{-1} から 10^{-4} の 4 つの値の中で最も適切なものを選択した．本実験は，以下の諸問題に取り組む．

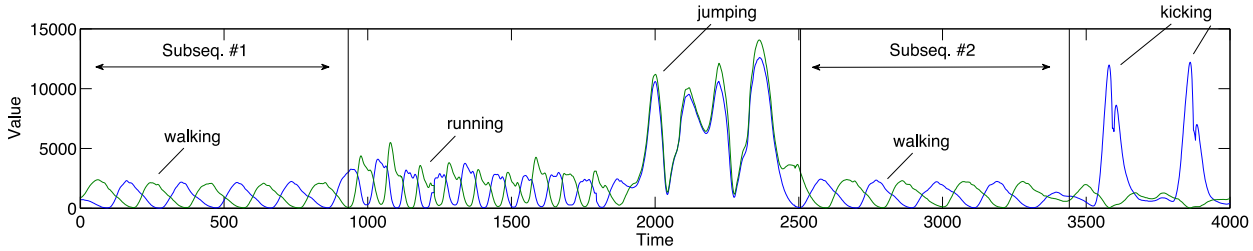
(1)　シーケンスのパターン発見における本手法の有効性
(2)　データストリーム監視における計算時間とメモリ使用量の検証

5.1 シーケンスパターンの発見

本節では，部分シーケンスのパターン発見における *StreamScan* の有効性を立証するた



(a) *MoCap* #1 (squatting: 2280-2520, 2640-2880)



(b) *MoCap* #2 (walking: 1-900, 2460-3360)

図 7　*MoCap* データにおけるシーケンスパターンの発見（1）
Fig. 7　Discovery of sequence patterns in *MoCap* (1).

め，実データを用いたケーススタディを示す．それぞれのデータセットにおいて，問合せ用のモデルは，複数のシーケンスから Baum-Welch アルゴリズム[18] を用いて学習を行い作成した．図 7〜図 11 は，*StreamScan* が適合する部分シーケンスを検出している様子を示している．ここで，適合する部分シーケンスが複数存在した場合は，それらすべてを図に示しており，すべてのデータセットに対する本手法による探索結果の再現率と適合率は 100%である．**表 2** は実験結果の詳細であり，2 列目はデータセットのシーケンス長，4 および 5 列目は各々のデータセットから提案手法によって検出した部分シーケンスの開始点と長さを示している．

MoCap

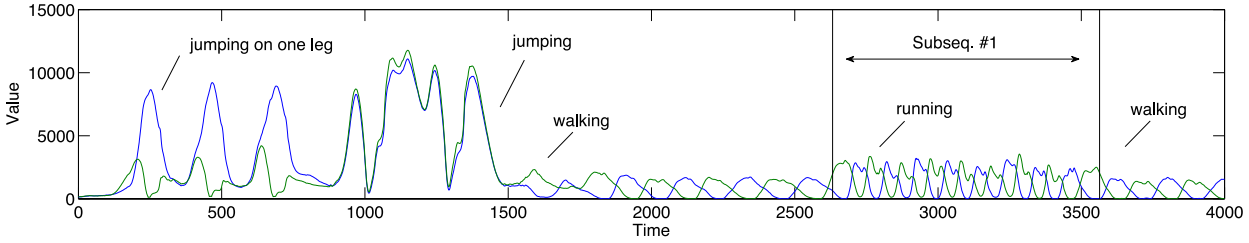
MoCap は，1 秒 120 フレームでヒトの動きを計測したモーションキャプチャのデータセットである[⋆1]．実験で利用したデータは，ヒトの典型的な動き（walking，running，squatting，jumping 等）がランダムに繰り返されるというものであり，それぞれ，各フレームごとの両足の運動エネルギーの値を 2 次元のシーケンスとして表現している．実験で利用したデータは，2 次元のデータを 100 のグリッドに分割し離散データとして扱った．問合せのモデルには，4 種類の動き（squatting，walking，running，climbing）を使用した．

図 7，**図 8** は，*MoCap* データにおけるヒトの動きの検出を行った様子を示しており，キャプションの括弧内の数値は 4 種類の動き（squatting，walking，running，climbing）の時間区間を表している．実験より，*StreamScan* は，ノイズに強く，すべての適切な部分シーケンスを検出できることが分かる．実際に，図 7 (a) はシーケンスの中から squatting モーションを正しく検出していることを示し，図 7 (b) は walking モーションの検出に成功している．同様に，図 8 (c) と (d) はそれぞれ，running モーションと climbing ladder モーションを検出している．表 2 は実験の詳細である．

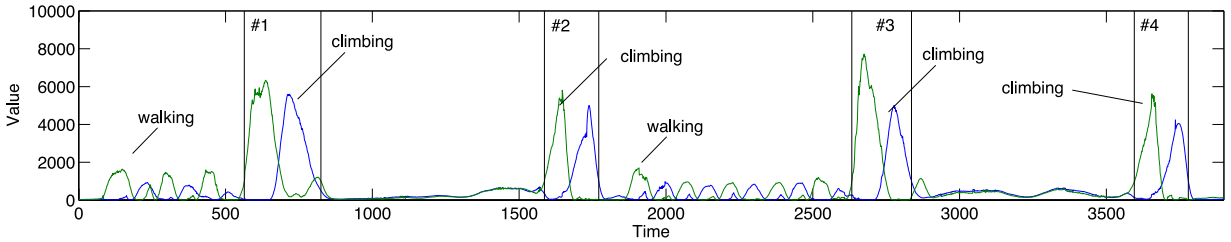
EEG

このデータセットは，アルコール依存症の遺伝的素因と EEG の関連性の大規模調査の結

⋆1 http://mocap.cs.cmu.edu/



(c) *MoCap* #3 (running: 2600-3560)



(d) *MoCap* #4 (climbing: 540-840, 1560-1800, 2640-2880, 3560-3780)

図 8　*MoCap* データにおけるシーケンスパターンの発見（2）
Fig. 8　Discovery of sequence patterns in *MoCap* (2).

果をまとめたものであり，UCI Repository によって公開されている[⋆1]．各データは，1 秒間に 256 Hz（3.9-msec epoch）で 64 カ所の電極の測定値を持ち，そのシーケンス長は 256 である．本実験では，任意の 2 つの電極を選び，それぞれに対し 10 分割して使用した．なお，すべてのデータはアルコール依存傾向（alcoholic），非アルコール依存傾向（control）の 2 つのグループに分けられる．アルコール依存傾向のシーケンスは，いくつかの大きなスパイクを有しており，ここではこれらのアルコール依存傾向の検出を行う．

図 9 に示すとおり，アルコール依存傾向のパターンは，20〜100 ボルトの間で変動しており，*StreamScan* は，これらのパターンをシーケンスの中から正確に検出することができた．

Voice

UCI のウェブサイトで公開されているデータであり，複数名の男性話者からの日本語の母音の発音データを，640 個の時系列データとして蓄積している．各データは 12 次元の時系列データとして表現され，その長さは 7 から 29 である．ここでの実験では，任意の 5 つの次元を選び，それぞれに対し 25 分割して使用した．これらのデータから，特定の発話者の検出を行う．**図 10** は，*Voice* データのシーケンスを示しており，3 人の話者の声が含まれている（話者 A：0-500，話者 B：500-901，話者 C：901-1202）．問合せモデルには，話者 B を用いた．図に示すとおり，我々のアルゴリズムは *Voice* データにおいても適切なシーケンスの検出に成功した．

Gene

UCI Repository で公開されている primate splice-junction 遺伝子（DNA）のデータセットである．各データは，8 つの文字（A，G，T，C，D，N，S，R）で構成された文字列として表現される．このデータ集合から，exon/intron 境界（EI）と，intron/exon 境界（IE）の認識を行うことが本データの課題である．本実験では，EI のグループに属する複数のシーケンスから，モデルを学習し，問合せデータとして使用した．

図 11 は，ランダムに作成した DNA のシーケンスデータである．このシーケンスは 5 つの部分シーケンスから構成されており，それぞれの部分シーケンスは，EI，IE，それ以外の 3 つのクラスのうちいずれかに属している．*StreamScan* は，このシーケンスの中から，正

⋆1 http://archive.ics.uci.edu/ml/

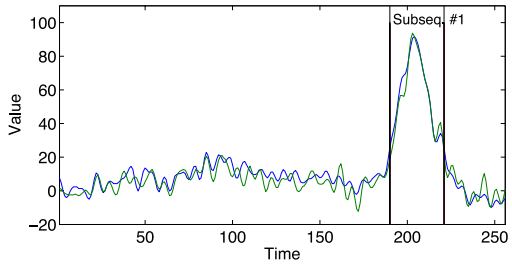


図 9　*EEG* データにおけるシーケンスパターンの発見
Fig. 9　Discovery of sequence patterns from the *EEG* data.

表 2　*StreamScan* の実験結果
Table 2　Results of optimal-match queries.

Datasets	Sequence length	Threshold ϵ	Matching subsequences	
			Starting position	Length
MoCap #1	3000	10^{-4}	2357	181
			2739	176
MoCap #2	4000	10^{-4}	1	931
			2506	935
MoCap #3	4000	10^{-4}	2632	932
MoCap #4	4000	10^{-4}	564	261
			1586	185
			2633	203
			3595	184
EEG	256	10^{-4}	190	31
Voice	1202	10^{-1}	500	397
Gene	300	10^{-2}	0	61
			111	66

しく特定の部分シーケンスを発見することができた．具体的には，図 11 において，正解となる部分シーケンスはは太字で示されているが，*StreamScan* は 2 つの部分シーケンス（下線部）を最適解として出力している．これにより，*StreamScan* は，複数のパターンを持つ遺伝子配列の中から，EI のパターンのみを正しく検出することができることが示された．

5.2　性　　能

4.2 節では，*StreamScan* の計算量について議論したが，本研究では実際的な状況でそれを

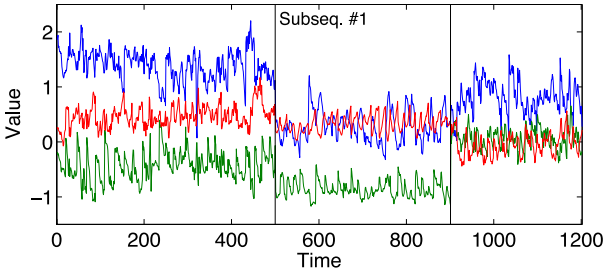


図 10　*Voice* データにおけるシーケンスパターンの発見
Fig. 10　Discovery of sequence patterns from the *Voice* data.

GCCCTGGCACCCAGCACAATGAAGATCAAGGTGGGTGTCTTTCCTGCCT
GAGCTGACCTGTAAATGTGTTTCCTGCATACAGTCAAAGTTGCCACTTCTT
TTTCTTCATATCATCGATCT**CCCTCCATCGTGGGGCGCCCCAGGCACCAGG**
TAGGGGAGCTGGCTGGGTGGGGCAGCCCCAATATCTTTCGTTGGCTTCC
AGGTTACAGAAAAATAATTTGTAACAAAGTTTAAAGGTCAGCTCAGGGCT
TCTTGTCCTTTCCTTCCCAGGGCGTGATGGTGGGCATGGGTCAGAAGGAT

図 11　*Gene* データにおけるシーケンスパターンの発見
Fig. 11　Discovery of sequence patterns in *Gene*.

確認するために実験を行った．**図 12** は，*StreamScan* と，従来手法である *Standard-Viterbi* との計算時間の比較である．データ集合には *MoCap* を用い，シーケンス長 n を変化させている．図では時間ごとのトレリス構造の更新と部分シーケンスの検出の合計時間を平均し，計算時間として示している．

実験結果から，*StreamScan* の性能が *Standard-Viterbi* と比較し非常に高いことが分かる．この傾向は，4.2 節における理論的な議論と合致する．従来手法の計算量が $O(nk^2)$ であることと比較し，*StreamScan* は計算コストの大幅な低減化を達成しており，それは n に依存せず一定である．本実験では，最大で 500,000 倍の高速化を達成している．

図 13 はトレリス構造の更新に必要なメモリ使用量を示したものである．本論文では適切な部分シーケンスの位置情報のみを出力することを想定してきたが，*StreamScan* は最適な部分シーケンスの Viterbi パスに関する情報についても出力することが可能である．図において，*StreamScan* は前者の場合，*StreamScan* （path）は後者の場合のメモリ使用量を示している．*StreamScan* （path）のメモリ使用量は検出した部分シーケンスの長さに依存するが，従来手法よりも大幅に低減させている．また *StreamScan* はストリームの長さに依存

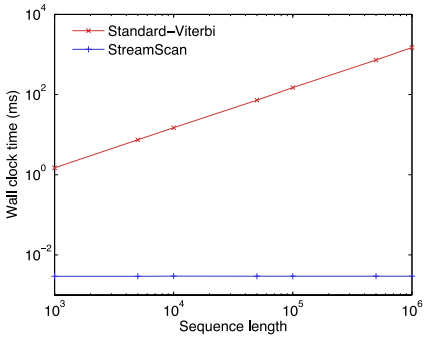


図 12　データシーケンス長に対する計算時間．*StreamScan* は最大で 500,000 倍の高速化を達成している
Fig. 12　Wall clock time as a function of sequence length. *StreamScan* is up to 500,000 times faster.

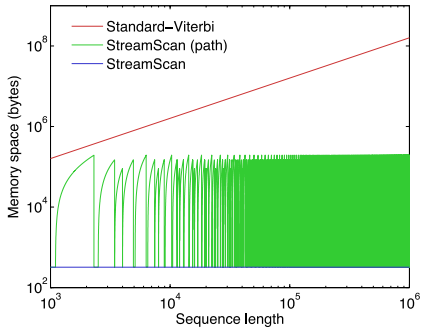


図 13　データシーケンス長に対するメモリ使用量．*StreamScan* はシーケンス長に依存せず一定のメモリ使用量を示している
Fig. 13　Memory space consumption as a function of sequence length. *StreamScan* can handle data streams with a small constant memory space.

せず一定のメモリ使用量を示している．

6. おわりに

本論文では，データストリームのための HMM に基づく部分シーケンスの検出の問題を扱い，その問題を解決するための手法である *StreamScan* を提案した．従来手法と異なり，*StreamScan* はたった 1 つのトレリス構造を用いて最適な部分シーケンスを検出することが可能である．*StreamScan* は検出漏れがないことを保証しながらも，計算時間とメモリ使用量を大幅に削減しており，それはシーケンス長に依存せず一定である．本論文では，理論的な検証だけでなく，実験でもこの優位性を確認しており，従来手法に比べて最大で 500,000 倍の高速化を達成している．

参 考 文 献

1) Bahl, L.R. and Jelinek, F.: Decoding for channels with insertoins, deletions and substitutions with applications to speech recognition, *IEEE Trans. Informat. Theory*, Vol.IT-21, pp.404–411 (1975).
2) Bahl, L.R., Jelinek, F. and Mercer, R.L.: A maximum likelihood approach to continuous speech recognition, *IEEE TPAMI*, Vol.PAMI-5, pp.179–190 (1983).
3) Barbará, D., Goel, R. and Jajodia, S.: Mining Malicious Corruption of Data with Hidden Markov Models, *DBSec*, pp.175–189 (2002).
4) Bocchieri, E.: Vector quantization for the efficient computation of continuous density likelihoods, *ICASSP*, pp.692–695 (1993).
5) Durbin, R., Eddy, S.R., Krogh, A. and Mitchison, G.: *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge University Press (1999).
6) Elmezain, M., Al-Hamadi, A. and Michaelis, B.: Hand trajectory-based gesture spotting and recognition using HMM, *Int. Conf. on Image Processing*, pp.3577–3580 (2009).
7) Jelinek, F.: *Statistical methods for speech recognition*, The MIT Press (1999).
8) Jelinek, F.: A fast sequential decoding algorithm using a stack, *IBM J. Res. Develop.*, Vol.13, pp.675–685 (1969).
9) Freitag, D. and McCallum, A.: Information Extraction with HMM Structures Learned by Stochastic Optimization, *AAAI/IAAI*, pp.584–589 (2000).
10) Gales, M., Knill, K. and Young, S.: State-based Gaussian selection in large vocabulary continuous speech recognition using HMMs, *TSAP*, pp.152–161 (1999).
11) Gilbert, A.C., Kotidis, Y., Muthukrishnan, S. and Strauss, M.: Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries, *Proc. VLDB*, Rome, Italy, pp.79–88 (2001).
12) Guha, S., Meyerson, A., Mishra, N., Motwani, R. and O'Callaghan, L.: Clustering Data Streams: Theory and Practice, *IEEE TKDE*, Vol.15, No.3, pp.515–528 (2003).
13) Guo, Z., Zhang, Z., Xing, E.P. and Faloutsos, C.: Enhanced max margin learning on multimodal data mining in a multimedia database, *KDD*, pp.340–349 (2007).
14) Hu, J., Brown, M.K. and Turin, W.: HMM Based On-Line Handwriting Recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol.18, No.10, pp.1039–1045 (1996).

15) Hughey, R. and Krogh, A.: Hidden Markov models for sequence analysis: extension and analysis of the basic method, *Computer Applications in the Biosciences*, Vol.12, No.2, pp.95–107 (1996).
16) Hunt, M. and Lefebvre, C.: A comparison of several acoustic representations for speech recognition with degraded and undegraded speech, *ICASSP*, pp.262–265 (1989).
17) Lee, H.-K. and Kim, J.H.: An hmm-based threshold model approach for gesture recognition, *IEEE TPAMI*, Vol.21, pp.961–973 (1999).
18) Levinson, S.E., Rabiner, L.R. and Sondhi, M.M.: An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition, *Bell Syst. Tech. J.*, Vol.62, pp.1035–1074 (1982).
19) Li, C., Zhai, P., Zheng, S.-Q. and Prabhakaran, B.: Segmentation and recognition of multi-attribute motion sequences, *ACM Multimedia*, pp.836–843 (2004).
20) Mori, T., Segawa, Y., Shimosaka, M. and Sato, T.: Hierarchical Recognition of Daily Human Actions Based on Continuous Hidden Markov Models, *Int. Conf. on Automatic Face and Gesture Recognition*, pp.779–784 (2004).
21) Mount, D.W.: *Bioinformatics: sequence and genome analysis*, Cold Spring Harbor Laboratory Press (2001).
22) Ney, H., Mergel, D., Noll, A. and Paesler, A.: Data driven search organization for continuous speech recognition, *IEEE Trans. Signal Processing.*, Vol.40, No.2, pp.272–281 (1992).
23) Papadimitriou, S., Brockwell, A. and Faloutsos, C.: Adaptive, Hands-Off Stream Mining, *Proc. VLDB*, Berlin, Germany, pp.560–571 (2003).
24) Papadimitriou, S., Sun, J. and Faloutsos, C.: Streaming Pattern Discovery in Multiple Time-Series, *Proc. VLDB*, Trondheim, Norway, pp.697–708 (2005).
25) Pfurtscheller, G., Flotzinger, D. and Neuper, C.: Differentiation between finger, toe and tongue movement in man based on 40 Hz EEG, *Electroencephalography and Clinical Neurophysiology*, pp.456–460 (1994).
26) Raymer, M.L., Doom, T.E., Kuhn, L.A. and Punch, W.F.: Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm, *IEEE Trans. Systems*, pp.802–813 (2003).
27) Sagayama, S., Knill, K. and Takahashi, S.: On the use of scalar quantization for fast HMM computation, *ICASSP*, pp.213–216 (1995).
28) Sakurai, Y., Faloutsos, C. and Yamamuro, M.: Stream Monitoring under the Time Warping Distance, *Proc. ICDE*, Istanbul, Turkey, pp.1046–1055 (2007).
29) Sakurai, Y., Papadimitriou, S. and Faloutsos, C.: BRAID: Stream Mining through Group Lag Correlations, *Proc. ACM SIGMOD*, Baltimore, Maryland, pp.599–610 (2005).
30) Scheffer, T., Decomain, C. and Wrobel, S.: Mining the Web with Active Hidden Markov Models, *ICDM*, pp.645–646 (2001).
31) Skounakis, M., Craven, M. and Ray, S.: Hierarchical Hidden Markov Models for Information Extraction, *IJCAI*, pp.427–433 (2003).
32) Sykacek, P. and Roberts, S.J.: Adaptive Classification by Variational Kalman Filtering, *NIPS*, pp.737–744 (2002).
33) Zhu, Y. and Shasha, D.: Statistical Monitoring of Thousands of Data Streams in Real Time, *Proc. VLDB*, Hong Kong, China, pp.358–369 (2002).
34) 岡　隆一：連続 DP を用いた連続音声認識，音響学会音声研資料 S78-20，pp.145–152 (1978).
35) 高橋勝彦，関　進，小島　浩，岡　隆一：ジェスチャー動画像のスポッティング認識，電子情報通信学会論文誌 D，Vol.77, No.8, pp.1552–1561 (1994).

(平成 23 年 6 月 20 日受付)
(平成 23 年 9 月 20 日採録)

(担当編集委員　手塚 太郎)

松原　靖子

2006 年お茶の水女子大学理学部情報科学科卒業．2009 年同大学院人間文化創成科学研究科理学専攻博士前期課程修了．2009 年より京都大学情報学研究科社会学情報専攻博士後期課程に在籍．データストリーム処理，大規模データマイニングに関する研究に従事．日本データベース学会学生会員．

櫻井　保志（正会員）

1991 年同志社大学工学部電気工学科卒業．1991 年日本電信電話（株）入社．1999 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．2004〜2005 年カーネギーメロン大学客員研究員．本会平成 18 年度長尾真記念特別賞，本会平成 16 年度および平成 19 年度論文賞，電子情報通信学会平成 19 年度論文賞，日本データベース学会上林奨励賞，ACM KDD best paper awards（2008，2010）等受賞．索引技術，データストリーム処理，センサーデータ処理技術の研究に従事．ACM，電子情報通信学会，日本データベース学会各会員．

吉川　正俊（正会員）

京都大学大学院工学研究科博士後期課程修了．工学博士．京都産業大学，奈良先端科学技術大学院大学，名古屋大学を経て 2006 年より京都大学大学院情報学研究科教授．この間，南カリフォルニア大学客員研究員，ウォータルー大学客員准教授．The VLDB Journal および Information Systems（Elsevier/Pergamon）の編集委員．XML データベース，異種情報源の統合等の研究に従事．電子情報通信学会，ACM，IEEE Computer Society 各会員．