社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－DBS－143（7）
2007／7／2

タイムワーピングを考慮したトレンド検出

豊田 真智子† 市川 俊一† 櫻井 保志††

† 日本電信電話株式会社 NTT 情報流通プラットフォーム研究所
〒 180–8585 東京都武蔵野市緑町 3–9–11
†† 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
〒 619–0237 京都府「けいはんな学研都市」精華町光台 2-4
E-mail: †{toyoda.machiko,ichikawa.toshikazu}@lab.ntt.co.jp, ††yasushi@cslab.kecl.ntt.co.jp

あらまし データストリームは，金融，センサネットワーク管理，製造，ネットワーク監視等の様々な分野で注目されているデータモデルである．これらのアプリケーションで重要とされるストリーム監視においては，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．本稿では，2つのシーケンスの類似度を測定するための新たな関数を提案し，データストリームのための効率的なシーケンスマッチングアルゴリズムを述べる．人工データと実データを用いた実験により，シーケンスのトレンドを効率的に検出できることを示す．

キーワード データストリーム，タイムワーピング，トレンド検出，部分マッチング

Trend Detection under the Time Warping

Machiko TOYODA†, Toshikazu ICHIKAWA†, and Yasushi SAKURAI††

† NTT Information Sharing Platform laboratories, NTT Corpration
9–11, Midori-cho 3-Chome Musashino-shi, Tokyo, 180–8585 Japan
†† NTT Communication Science Laboratories, NTT Corporation
2–4, Hikaridai, Seika-cho, "Keihanna Science City", Kyoto, 619–0237 Japan
E-mail: †{toyoda.machiko,ichikawa.toshikazu}@lab.ntt.co.jp, ††yasushi@cslab.kecl.ntt.co.jp

Abstract Data stream is becoming increasingly important in several domains such as finance, sensor network environment, manufacture, network monitoring. The most fundamental support needed in these applications is efficient monitoring of time series data streams, and a subsequece-matching mechanism is required. In this paper, we propose a new function to recognize the similarity between subsequences, and then present an efficient sequence matching algorithm for data streams. Several experiments with synthetic and real data sets show that our algorithm works well as expected; it can efficiently detect sequence trends in the data sets.

Key words Data Stream, Time Warping, Trend Detection, Subsequence Matching

1. まえがき

計算機システムの性能向上と共に大量のデータが生み出され、増加の一途をたどっている．金融，センサネットワーク管理，製造，ネットワーク監視などの分野においては，時間の経過と共に増加するデータストリームを処理することが求められている．データストリーム処理における要素技術として，ストリームの監視が挙げられる．しかし，高いビットレートで送信され大きくなり続けるデータを，限られたメモリで処理しなければならないという課題のため，容易に実現することが難しい．

ストリーム監視においては，あらかじめ用意されている問い合わせパターンに類似する部分シーケンスを検索したり，頻出するパターンを分析したりすることが求められる．そのため，部分シーケンスマッチング技術が必要とされる．各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があるため，これらに柔軟に対応するよう，タイムワーピングも考慮することが重要となる．

本稿では，データストリームからアプリケーションの要求を考慮したトレンド検出の問題を扱う．これは，事前に問い合わせを用意することなく，データストリームの受信を開始した時点からアプリケーションが注目するスケールのトレンドを継続して検出することを意味する．これにより，頻出するトレンドの監視や類似トレンドの検索を行うことが可能となる．

また，トレンド検出のための新手法のみでなく，部分シーケンスマッチングのための類似判定手法を提案する．これは，従来のように，問い合わせパターンの全域に類似する部分シーケ

ンスを探索するだけでなく，問い合わせパターンの一部に類似する部分シーケンスも探索することが可能となる．

本稿は以下のように構成される．まず2節において，本稿で解決したい問題の定義を行い，3節で関連研究を述べる．4節で提案手法の概要を紹介し，5節で提案手法の評価実験を行う．最後に6節でまとめを述べる．

2. 問題定義

データストリーム X は時刻 $T = t_1, t_2, ..., t_n, ...$ で収集される $x_1, x_2, ..., x_n, ...$ の値からなる半無限長のシーケンスである．x_n は t_n における最新のデータであり，時間の経過と共に 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であるとする．一方，Y は，$y_1, y_2, ..., y_m$ の値からなるシーケンスであり，$Y[i_s : i_e]$ を i_s から i_e までの部分シーケンスと表す．

本稿ではデータストリームからのトレンド検出問題を扱う．ここでトレンドとは，データストリーム X において高い頻度で出現する部分シーケンスのパターンを指す．また，検出する部分シーケンスの長さを制限することも可能である．すなわち，部分シーケンスの長さの下限値を N，上限値を N' (例えば $N' = 2N$) とするとき，$X[t_s : t_e]$ において $N < t_e - t_s \leqq N'$ である．

よりフォーマルには，解決したい問題を二つに分けることができる．まず部分問題として，部分シーケンスマッチングの問題を最初に定義する．

[問題1]（ローカルアライメント） データストリーム X とシーケンス Y が与えられた時，類似する X の部分シーケンス $X[t_s : t_e]$ と Y の部分シーケンス $Y[i_s : i_e]$ を検出する．

この問題は，バイオインフォマティックスではローカルアライメントと呼ばれている[3]．バイオインフォマティックスにおけるローカルアライメントは，記号シーケンスを対象としているが，本稿では数値シーケンスのためのローカルアライメントを対象とする．

ローカルアライメントのためのスコアリング関数が得られるとトレンド検出に適用することができる．本稿で解決したい問題は以下の通りである．

[問題2]（トレンド検出） データストリーム X，シーケンス長の下限値 N，閾値 ε が与えられた時，以下の条件を満たす部分シーケンス Y を頻出パターンとして X から検出する．

（1） Y との間の類似スコアが ε 以上であり，かつ Y と重複しない部分シーケンスが X の中に k 個以上存在する．

（2） それら部分シーケンスおよび Y の部分シーケンスの長さは N 以上，N' 以下である．

ここで，k は類似部分シーケンスの頻度であり，典型的には k は n に比例する．本稿では Y を X に対する比較パターンと呼ぶ．また Y と類似する部分シーケンスが k 個以上存在するとき，特に Y を頻出パターンと呼ぶ．

3. 関連研究

本稿に関連する研究は，大きく2つに分けることができる．1つは時系列データからのパターン検出に関する研究，もう1つはデータストリームに関する研究である．

パターン検出に関する研究は，[1] や [7] において取り組まれている．これらの文献では，与えられた時系列データの特徴を表し，繰り返し現れるパターンを"モチーフ"として定義し，このモチーフを高速に検出ことを目的としている．しかし，これらの手法は蓄積されたデータに対するものであり，本目的とは異なる．

データストリームに対する取り組みは，従来データマイニングで行われてきた内容をデータストリームに拡張するための手法が数多く提案されている．Manku らは，データストリームの頻出値を計算するアルゴリズムを提案し[2]，Zhu らは複数のデータストリームをモニタリングすることにより，ストリーム間の相関関係を高速に検出する手法を提案している[8]．また，櫻井らは文献[5]において複数のデータストリーム間の遅延相関を検出するアルゴリズムを提案し，文献[4]において，ダイナミックタイムワーピング (DTW: Dynamic Time Warping) を用いたストリームの高速なシーケンスマッチング手法を提案している．しかし，これらの文献はストリームのトレンド検出について扱われているものではない．

4. 提案手法

2節で述べた問題を解決するために検討したアルゴリズムは，以下のように動作する．

（1） データストリームの先頭から $2N$ の長さのシーケンスを初期パターンとして検出し，比較パターン A とする．

（2） 検出した比較パターン A と残りのデータストリームをデータが到着する度に1データずつ比較し，類似する部分シーケンスであると判断できた場合は類似パターン a_1 として検出する．

（3） もし，N 個のデータを比較した時点で類似する部分シーケンスではないと判断した場合，この部分シーケンスを新たなパターンとして検出し，比較パターン B とする．

この処理を繰り返し，比較パターンと類似パターンを検出し続けることにより，データストリームをパターン化する．本節では，このアルゴリズムを，ストリーム処理としてのリアルタイム性を維持しつつ，精度の高いパターンを検出するために検討した工夫について述べる．

4.1 類似判定

4.1.1 従来手法の問題点

2つのシーケンス間の類似度を測定する手法としては，ダイナミックタイムワーピング (DTW: Dynamic Time Warping) が代表的である．DTW は，2つのシーケンス間の距離を最小化するように時間軸方向に伸長を行い，各要素同士をマッチングさせた計算により距離値を求め，類似度を判定する．この距離値は DTW 距離と呼ばれ，最適にシーケンス長を調整した後の距離の合計値で表される．動的計画法により，図1の右図のようなマトリックスを用いて計算され，黒く示されたパスは DTW により対応付けられた要素を示しており，マトリックスの中で左下から右上へと伸びる最適なワーピングパスである．この値が小さいほど2つのシーケンスは類似度が高く，0の場

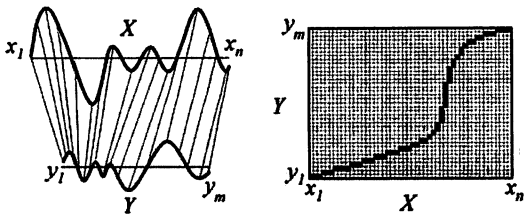

図 1　DTW を用いたシーケンスマッチング

合は完全に一致していることを意味する．DTW を用いることにより，同じ長さのシーケンスペアだけでなく，長さの異なるシーケンスペアやサンプリングレートの異なるシーケンスペアの距離求めることができるため，柔軟に類似度を判定することが可能となる．

DTW は固定長の X と Y のシーケンスマッチングに利用され，バイオインフォマティックスではローカルアライメントに対してグローバルアライメントという問題として扱われている．文献 [4] の手法は，与えられたシーケンス X と Y に対し，類似する X の部分シーケンス $X[t_s:t_e]$ を DTW に基づいた手法により検出するように DTW を改良している．もしこれらの手法を用いてローカルアライメントを行おうとした場合，単位時間当たりに $O(n^2)$ の時間を要することになる．また，比較パターン全体にマッチングする部分シーケンスを検出するものであるため，類似パターンを見逃してしまう場合もある．これについては，5.1 節で詳しく述べる．

一方，2 つのシーケンス間のローカルアライメントを行うために，バイオインフォマティクス分野では Smith-Waterman アルゴリズムが利用されている [6]．最適にシーケンス長を調整した後，類似度をスコアとして算出するものであり，要素間の比較において類似度が高ければスコアが加算され，低ければスコアは減算される仕組みとなっている．このアルゴリズムでは，用いた場合の単位時間当たりの計算時間は $O(n)$ であり，DTW より大幅に削減する．

しかし，Smith-Waterman アルゴリズムは記号シーケンスに対して用いられるアルゴリズムであるため，本稿で対象とするデータストリームに適用することが難しい．そこで文献 [4] のアイディアを基に Smith-Watarman アルゴリズムを改良することで，データストリームに対応した独自のスコア計算関数を考案した．

4.1.2　スコア計算関数

本稿で考案したスコア計算関数は，データストリーム $X=(x_1,x_2,..,x_n,...)$ から比較パターン $Y=(y_1,y_2,...,y_m)$ に類似する部分シーケンス $X[t_s:t_e]$ を類似パターンとして検出するために使用されるものであり，データが 1 つ到着する度に比較パターン Y と部分シーケンス $X[t_s:t_e]$ の類似度をスコアとして計算する．スコアはマトリックスを用いて，比較パターンと部分シーケンスの各要素同士を対応させて計算される．部分シーケンス $X[t_s:t_e]$ と Y の部分シーケンス $Y[i_s:i_e]$ が類似する時，(t_s,i_s) をこのマッチングにおける開始位置，(t_e,i_e) を終了位置と呼ぶ．開始位置の情報を保持することにより，新しく t_n に到着したデータとどの開始位置 t_s から始まる部分シーケンス $X[t_s:t_n]$ が比較パターン Y と類似しているのかを，過去にさかのぼることなく把握することができる．これにより t_1 から t_n までの全ての情報をマトリックスに保持しておく必要がなくなり，ストリーム処理に対応しうるメモリ使用量で計算することができるようになる．

比較パターン $Y=(y_1,y_2,...,y_m)$ に類似する部分シーケンス $X[t_s:t_e]$ のスコア $S(X[t_s:t_e],Y)$ は，以下のように計算される．

$$S(X[t_s:t_e],Y)=s(t_e,i_e)=max(s(t,i))$$

$$s(t,i)=max\begin{cases}0\\ \frac{\epsilon}{N}-\|x_t-y_i\|+s_{best}\end{cases}$$

$$s_{best}=max\begin{cases}s(t,i-1)\\ s(t-1,i)\\ s(t-1,i-1)\end{cases} \tag{1}$$

$$s(t,0)=0, s(0,i)=0$$

ここで，ϵ は閾値，N は部分シーケンスの長さの下限値を意味し，$t=1,...,n,\quad i=1,...,m$ である．$\|x_t-y_i\|$ は 2 つの数値の距離を求める計算式であり，本実験では $(x_t-y_i)^2$ を用いているが，L1 距離 ($|x_t-y_i|$) などの他の距離計算式を用いることも可能である．

スコアマトリックスで保持される $X[t_s:t_e]$ の開始位置情報 $p(t,i)$ は以下のように求められる．

$$p(t,i)=(t_s,i_s)$$

$$(t_s,i_s)=\begin{cases}p(t,i-1)\\ \quad (s_{best}=s(t,i-1))\\ p(t-1,i)\\ \quad (s_{best}=s(t-1,i))\\ p(t-1,i-1)\\ \quad (s_{best}=s(t-1,i-1))\\ (t,i)\\ \quad (s_{best}=0)\end{cases} \tag{2}$$

$$p(t,0)=(t,0),\quad p(0,i)=(0,i)$$

また，$S(X[t_s:t_e],Y)$ の開始位置 t_s は，次のように得られる．

$$t_s=p(t_e,i_e) \tag{3}$$

ここで，閾値 ϵ 以上となる部分シーケンスは，ほぼ同じ区間に重複して存在する可能性があることを考慮する必要がある．これは，$X[t_s-1:t_e]$，$X[t_s-2:t_e-1]$，$X[t_s:t_e]$，$X[t_s+1:t_e+1]$ など，わずかに区間が異なる部分シーケンスも ϵ 以上となる場合があることを意味する．これらすべてをパ

$y_4=4$	57 (5, 2)	18 (5, 2)	86 (5, 2)	78 (5, 2)	128 (5, 2)	150 (5, 2)	166 (5, 2)	110 (5, 2)	0 (6, 1)	81 (6, 1)	6 (6, 1)	0 (6, 1)	29 (17, 2)
$y_3=9$	93 (5, 2)	49 (5, 2)	85 (5, 2)	89 (5, 2)	105 (5, 2)	116 (6, 1)	77 (6, 1)	116 (6, 1)	60 (6, 1)	36 (6, 1)	36 (6, 1)	0 (6, 1)	5 (17, 2)
$y_2=6$	24 (5, 2)	13 (5, 2)	49 (6, 1)	58 (6, 1)	83 (6, 1)	107 (6, 1)	107 (6, 1)	83 (6, 1)	0 (6, 1)	9 (14, 2)	0 (15, 2)	0 (15, 1)	16 (17, 2)
$y_1=11$	0 (5, 1)	24 (6, 1)	24 (6, 1)	48 (6, 1)	48 (6, 1)	37 (6, 1)	0 (6, 1)	21 (12, 1)	0 (12, 1)	0 (14, 1)	16 (15, 1)	0 (15, 1)	0 (17, 1)
x_i	5	12	6	10	6	5	1	13	18	2	14	18	3
t	5	6	7	8	9	10	11	12	13	14	15	16	17

図 2 マトリックスの例

ターンとして検出することはあまり意味がなく，ユーザに冗長な情報を与えることになる．そこで，重複する部分シーケンスが存在する場合は，比較パターンと最も類似度が高いシーケンス 1 つを最適部分シーケンスとして検出することで余分な部分シーケンスを除外することとする．

4.2 パターン検出のタイミング

図 2 は，$N=2,\quad \varepsilon=50,$
$X=(11,6,9,4,5,12,6,10,6,5,1,13,18,2,14,18,3),$
が与えられた場合に作成されるマトリックスを示しており，色付けされた部分シーケンスが，類似パターンとして検出される部分シーケンスのパスを意味する．この例を用いてどのように類似パターンが検出されるかを具体的に示す．

まず，長さ $4(=2N)$ の部分シーケンス $X[1:4]=(11,6,9,4)$ が最初の比較パターンとして検出され，要素が 1 つ到着する度にスコアが計算される．$t=7$ において，$s(7,4)=86 \geqq \varepsilon$，部分シーケンス長 $\geqq N$ となる $X[5,7]$ を出力候補の部分シーケンスとして発見するがまだ検出は行わず，$t=11$ において，最適部分シーケンス $X[5,11]$ を発見する．$t=12$ における最大スコアは $s(12,3)=116$ と低下するが，この時点でもまだ検出しない．なぜなら，$s(12,3)$ の開始位置 $t_s=6$ は部分シーケンス $X[5,11]$ と重複しており，今後もスコアが上昇する可能性があるためである．これらの最終的な報告は，$t=17$ で行われる．これは，$t=17$ のすべての要素の開始位置 t_s が，$X[5,11]$ の部分シーケンスの終了地点である $t=11$ 以降のシーケンスを開始位置としたスコアであり，これ以降に現れる最適部分シーケンスは，これらのシーケンスと重複しないことが確認されるためである．

なお実際のプログラムにおいては，図 2 で示したような 4×13 のマトリックスを作成する必要はなく，現在の要素と 1 つ前の要素の 4×2 のマトリックスを作成し，これらを更新し続けることによりスコアを計算できる仕組みとなっている．

4.3 類似パターンのグループ化

前節で提案したスコア計算関数により，比較パターンに類似する部分シーケンスを類似パターンとして検出することで，類似しない部分シーケンスを異なるパターンと定義し，継続してパターン化を行うことができるようになる．しかし，検出されたこれらパターンすべてを残りのデータストリームとマッチングさせることは，大幅な計算コスト増加につながり効率的であるとは言えない．そこで，比較パターンとパターンに類似して検出された類似パターンを 1 つのグループとして定義する．そして，グループ内のパターンから代表を選択し，データストリームとのマッチングに使用する比較パターンとする．このグループ数は比較パターンの数と等しくなる．グループ化を行うことにより検出されたパターンすべてを使用する必要がなくなるため，大幅な計算コスト削減にもつながる．

比較パターンとして選択するパターンは，グループの特徴を最もよく表すパターン，すなわち，そのグループの最も重心に近いものを選択することが望ましい．そこで，グループ内のすべてのパターン同士のスコアを計算し，これらの合計値が最も大きなパターンを代表とし，比較パターンとして選択する．比較パターン選択処理は，1 グループ内に 3 以上パターンが存在する場合に行うものとする．パターン同士のスコアが保存されていること，パターン選定の際の計算コストを削減することができる．また，1 グループに含まれるパターンに上限を設定することも可能であり，この場合には合計スコアの最も小さなものを新しく検出されたパターンと入れ替えることにより行うものとする．

5. 評価実験

本節では，提案手法の有効性を検証するため，人工データと実データを用いた実験を行う．人工データを用いた実験では，提案したスコア計算関数と DTW それぞれの類似判定関数を使用した場合について評価し，ローカルアライメントの影響について考察を行う．実データを用いた実験では，モーションキャプチャデータを使用し，より詳細にパターン検出の精度を考察する．

5.1 人工データを用いた実験

図 3 で示される人工データは，異なる 2 つのパターンが交互に現れるデータであり，なだらかな半円の周期はそれぞれ異なる．本実験では，提案手法における類似判定関数に DTW を使用した場合についても評価し，スコア計算関数を使用した場合との結果の違いを考察する．実験では，$N=1500$ に設定した．

実験の結果，DTW を用いた場合に検出されたパターンは 6 パターン，スコア計算関数を用いた場合に検出されたパターンは 2 パターンであり，図 4，5 のようになった．これらのパターンはすべて，実験終了後に最終的に代表パターンとなったパターンであり，スコア計算関数を用いた場合の結果のみ，代表パターンの入れ替えが発生し，図 3 における 3 番目のパターンとなっている．

DTW を用いた場合 (図 4)，比較パターンの全体と類似する

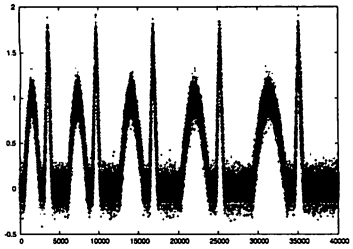

図 3　異なるパターンを含む人工データ

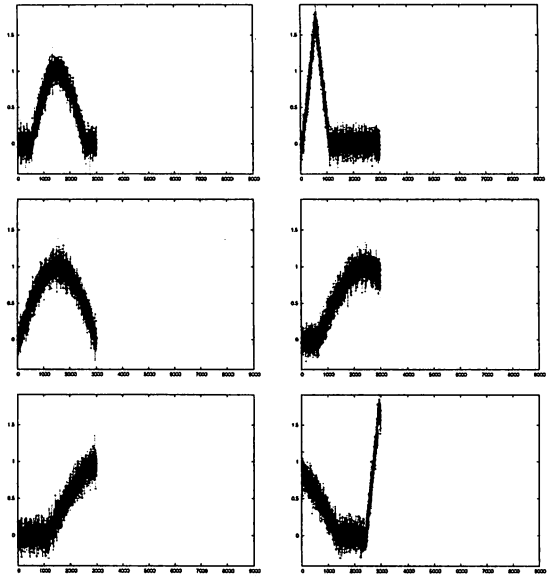

図 4　DTW を用いて検出されたパターン

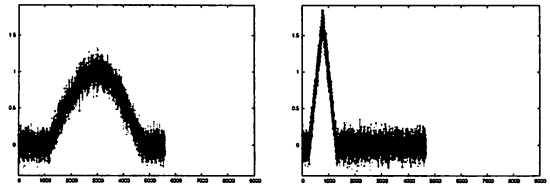

図 5　スコア計算関数を用いて検出されたパターン

部分シーケンスを探すため，周期が変化する半円を類似パターンとして検出することが難しくなる．その結果，それらがすべて新たなパターンとして検出されてしまう結果となった．一方，スコア計算関数を用いた場合 (図 5)，比較パターンの全体だけでなく，部分的に類似する部分シーケンスもマッチングが可能となるため 2 つのパターンのみが検出され，残りの部分シーケンスは類似パターンとして検出することに成功している．

5.2　実データを用いた実験

実データを用いた実験には，モーションキャプチャのデータを用いた．これは，被験者の各部位にマーカーを取り付け，取り付けた部位の角速度を 1 秒間に 120 回というサンプリング周期で測定したデータであり，カーネギーメロン大学によって作

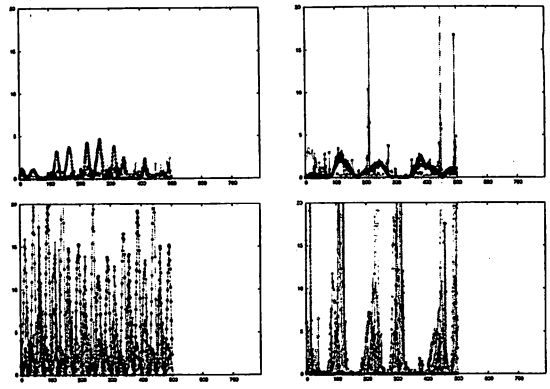

図 6　検出パターン例:左上) "walking"，右上) "squats"，左下) "running"，右下) "punches"

成され，提供されているものである．本実験では，測定されたデータの中から，二の腕・肘下・太もも・ふくらはぎの各部位に左右対称に取り付けられたマーカーから取得された 16 個の角速度データを，16 次元データとして使用した．

使用したモーションキャプチャデータは，8 つのモーションが"walking - squats - running - standing - stretching - walking - jumping - drinking - qunches - walking" という順に現れ，総サンプルリングポイント数が 10616，モーション時間は約 90 秒のデータである．

今回の実験では，各モーションのデータそれぞれを異なる部分シーケンスと考え，1) 異なるモーションを異なるパターンとして検出できるか，2) 同じモーションを類似パターンとして検出できるか，という点に注目して検証した．この検証は，検出されたパターンの時間と，その時間に対応したモーションが何であるかを対応付けることにより行う．

なお，本実験では $N = 250$ に設定した．この値は時間にすると約 2 秒に相当する数値である．

モーションキャプチャを入力データとして本手法を動作させた結果，11 のパターンが検出された．正しく検出できたモーションは，"walking"，"running"，"punches"の各モーション，1 種類のモーションから複数パターンが検出されたモーションは，"squates"，"stretch"，"jumping"の各モーション，異なるモーションが同一パターンとして認識されたものは，"standing"－"drinking"，"stretches"－"drinking"という結果であった．以下で結果の考察を述べる．

考察 1　異なるモーションの識別

図 6 は"walking"，"squats"，"running"，"punches"の各モーションを表すパターンである．モーション毎に各マーカーから取得される角速度が大きく異なるため，これらのモーションを含む 7 モーションを適切に識別し，パターンとして検出することに成功した．検出できなかったモーションは"drinking"であり，これについては後で考察を述べる．

考察 2　"walking"モーション

"walking"モーションは，0-1000，4680-5880，9480-10560 の

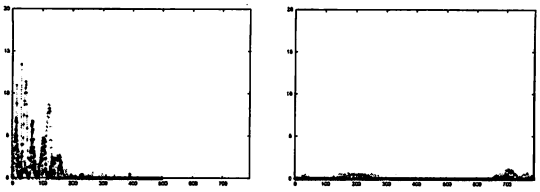

図 7 類似パターンとして検出された異なるモーションのパターン：左) "standing"，右) "drinking"

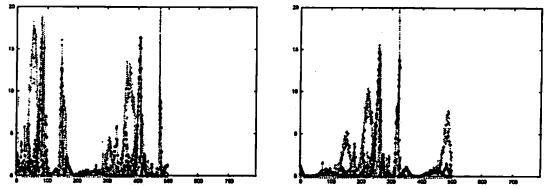

図 8 別パターンとして検出された"jumping"モーションのパターン

区間に現れる．このモーションでは，被験者が一定のリズムで歩き続けるという動作が行われている．図 6 に示したパターンは，0-499 の区間から検出されたパターンであり，700-1001，4701-4963，5332-5747，9634-10120 の 4 つの区間の部分シーケンスが類似パターンとして検出された．

検出された区間すべてが，"walking"モーションに対応する区間とほぼ一致しており，"walking"モーションを正確に検出できていることが確認される．"running"，"punches"の各モーションについても同様の結果が得られた．

考察 3 "drinking"モーション

"drinking"モーションは，7200-8760 の区間に現れるモーションであり，被験者が立ち姿勢で数回に分けて飲み物を飲むという動作が行われている．今回の実験では，"drinking"モーションを 1 つのパターンとして検出することができず，"standing"モーション，"stretches"モーションの類似パターンとして検出された．

図 7 は，"standing"モーションを表すパターンから類似パターンとして検出された"drinking"モーションを表すパターンである．この図から，"standing"モーションのパターンと"drinking"モーションのパターンは非常に類似していることが確認される．これは，"standing"モーションと"drinking"モーションの違いが片手の変化のみとなるためである．これらのモーションの識別を行うためには，被験者に取り付けるマーカーの選定が重要であると言える．

なお，"standing"の最初の区間のデータが大きくばらついているのは，"running"モーションから"standing"モーションへの切り替わり地点のデータであり，実際の"standing"モーションはデータにほとんど変化がない区間に相当する．

考察 4 "jumping"モーション

"jumping"モーションは，5880-7200 の区間に現れるモーションであり，被験者が 4 回に分けて繰り返しジャンプを行う．結果として，図 8 のように 2 つの異なるパターンが検出された．左のパターンが 1 回目と 2 回目のジャンプを，右のパターンが 3 回目と 4 回目のジャンプの最初の動作を意味している．1 回目と 2 回目のジャンプデータには違いが見られるが，2 回目と 3 回目のジャンプデータは比較的類似しているようにも見える．厳密にこれらが異なっているのか，スコア計算関数の類似パターン検出精度の問題なのか，どちらが影響しているのかは今後詳細に分析するつもりである．

6. まとめと今後の計画

本稿では，データストリームから頻出パターンや類似パターンを発見するためのアルゴリズムを提案し，その評価を行った．本アルゴリズムでは，事前に問い合わせパターンを用意することなく，到着するデータストリームから順次パターンを検出する．類似パターン検出のために，従来の DTW の欠点を補い，部分的なマッチングに対応した独自の類似判定関数であるスコア計算関数を考案した．人工データと実データを用いた実験により提案手法を評価し，DTW よりも高い精度でパターン検出を行えることを確認した．

今後は，スコア計算関数の処理についてより詳細な分析を行い，パターン検出の精度向上を目指す．また，提案手法の性能についても評価を行う予定である．

文 献

[1] B. Chiu, E. Keogh, and S. Lonardi: " Probabilistic discovery of time series motifs, " *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pp.493-498, Washington, DC, August 2003.

[2] G. S. Manku, and R. Motwani: "Approximate Frequency Counts over Data Stream," *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, pp.346-357, Hong Kong, China, August 2002.

[3] David W. Mount: "Bioinfomatics: Sequence and Genome Analysis", Cold Spring Harbor, New York, 2000.

[4] Y. Sakurai, C. Faloutsos, and M. Yamamuro: "Stream Monitoring under the Time Warping Distance, " *Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE 2007)*, Istanbul, Turkey, pp.1046-1055, Aplil 2007.

[5] Y.Sakurai, S.Papadimitriou, and, C. Faloutsos: "Braid: Stream Mining through Group Lag Correlations," *Proceedings of ACM SIGMOD*, pp.599-610, Baltimore, Maryland, June 2005.

[6] Smith TF, and Waterman MS: "Identification of Common Molecular Subsequences, " Journal of Molecular Biology 147, pp.195-197, 1981.

[7] Y. Tanaka, and K. Uehara: "Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle, " *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'03)*, pp.252-265, Springer, 2003.

[8] Y. Zhu, and D. Shasha: "StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time", *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, pp.358-369, Hong Kong, China, August 2002.