The Institute of Electronics, Information and Communication Engineers

論　文

部分シーケンスマッチングのためのストリームアルゴリズム*

豊田真智子[†,††]　櫻井　保志[†]　石川　佳治[†††]

A Stream Algorithm for Subsequence Matching*

Machiko TOYODA[†,††], Yasushi SAKURAI[†], and Yoshiharu ISHIKAWA[†††]

あらまし　本論文では，データストリームにおける cross-similarity の問題を定義する．本論文の目的は，データストリームから類似する部分シーケンスペアを検出することである．シーケンス間の類似度を測定する距離尺度には，時間軸上でのスケーリングを考慮することができるダイナミックタイムワーピング (DTW：Dynamic Time Warping) 距離を利用する．我々の提案する CrossMatch は，厳密に DTW に基づいた手法であり，データストリーム処理に適したワンパスアルゴリズムである．DTW を用いた純粋なアルゴリズムと比べて，CrossMatch は計算コストとメモリ使用量の大幅な低減化を実現する．理論的な分析を行い，提案アルゴリズムが精度を犠牲にすることなく類似する部分シーケンスペアを検出することを示す．また，実データと人工データを用いた実験から，CrossMatch がインクリメンタルに cross-similarity を検出することが確認された．

キーワード　データストリーム，部分シーケンスマッチング，ダイナミックタイムワーピング

1. まえがき

データストリームは，金融やセンサネットワーク管理，移動体追跡，ネットワークトラフィックモニタリング，Web クリックストリーム分析など，様々な分野で重要となっている．データストリームは，無限長時系列データであり，高速かつ連続的にオンラインで到着するため，すべてのデータを保持することは困難である．そのため，データストリームを効率的に処理するためのマイニング技術に注目が集まっている．

データストリーム処理が必要となるアプリケーションにおいて求められるのは，連続的にデータストリームをモニタリングする技術であり，その重要な要素技術の一つが部分シーケンスマッチングである．部分シーケンスマッチングに関する先行研究の多くは，問合せシーケンスに類似する部分シーケンスをデータストリームから検出する問題に焦点を当てている [1], [2]．本論文では，データストリーム間で類似する部分シーケンスペアを検出する問題に焦点を当てる．この問題は，頻出パターン検出やルール抽出，クラスタリング，異常検出などに適用される重要な問題である．

オフラインの環境と異なり，データストリームは連続的にデータが到着する．ストリームのサンプリングレートが異なる場合や時間周期が変化する場合があるため，これらに対応する距離尺度が必要となる．この問題を解決するために，距離尺度としてダイナミックタイムワーピング (DTW：Dynamic Time Warping) 距離 [3], [4] を利用する．DTW は様々な分野で幅広く利用されており [5]～[8]，時間軸上でのスケーリングを考慮することが可能なため，データストリームにおける部分シーケンスマッチングに適した距離尺度である．

我々の目標は，データストリームをモニタリングし，すべての類似する部分シーケンスのペアを自動的に検出することである．本論文で解決したい問題は，次のようなものである．

二つのデータストリームが与えられたとき，DTW に基づき，類似する部分シーケンスペアを検出する．

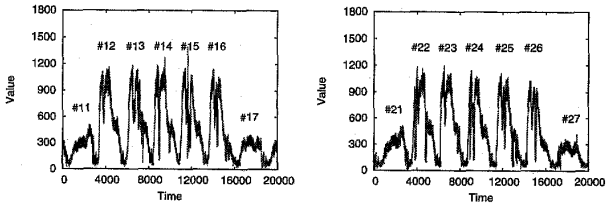
図 1 を用いてこの問題を説明する．問題定義の詳細については，**2.** で述べる．図 1 は，ウェブサイトに

† 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，京都府
NTT Communication Science Laboratories, NTT Corporation, 2–4 Hikaridai, Seika-cho, Soraku-gun, Kyoto-fu, 619–0237 Japan
†† 名古屋大学大学院情報科学研究科，名古屋市
Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
††† 名古屋大学情報基盤センター，名古屋市
Information Technology Center, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464–8601 Japan
* 本論文はデータ工学研究専門委員会推薦論文である．

 © (社) 電子情報通信学会 2011

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers



(a) シーケンス#1　(b) シーケンス#2

図 1　部分シーケンスマッチングの例
Fig. 1　Example of subsequence matching.

おけるアクセス数を 30 秒ごとに記録したものである．これらのサイトのアクセス数のシーケンスには，平日 5 日間の連続するアクセスパターン（# 12 から# 16 までと# 22 から# 26 まで）と週末のアクセスパターン（# 11 と# 17，# 21 と# 27）の二つの共通するパターンが含まれている．平日パターンのアクセス数は週末パターンのアクセス数より多く，夕方にかけて次第に減少する．一方週末パターンは平日パターンほど大きな変化は見られない．これらのパターンの周期はパターンごとに異なっており，小さなスケールで見ると非常にバースト的なデータである．本論文で解決したい問題は，二つのシーケンス間の部分的な類似を見つけることである．具体的には平日パターン# 12 から# 16 と# 22 から# 26，週末パターン# 11 と# 21，# 17 と# 27 などの部分シーケンスペアを意味し，これらはシーケンス# 1 と# 2 における類似する部分シーケンスペアである．

データストリームに対する部分シーケンスマッチングは，様々なアプリケーションに適用可能である．以下に具体的な例を示す．

- Web のトレンド分析：Web サイトにおけるアクセスパターンは，Web コンテンツやサイト構造の変化だけでなく，ユーザの興味の推移にも影響され，動的に変化する．リアルタイムなアクセスパターンの追跡により，ホットなトレンドを逃すことなく，サイトの改善やユーザへのパーソナライズによる情報提供を行うことが可能となる [9]．例えば，アクセスパターンの共通性に基づいたグループごとにユーザをクラスタリングすることで，Web サイトを利用するユーザの振舞いを分析することが可能となる．これらのグループは，類似するシーケンスペアを検出することで発見される．
- 動作認識：モーションキャプチャデータを用いた人の動作認識は，コンピュータアニメーションやスポーツ，医療において重要とされており，多くの研究で取り組まれている [10]～[12]．このデータは 1 秒間に何度も計測され，複数のマーカから取得される高次元のデータストリームとなる．1 人のユーザが同じ動作を行ったとしても，測定されるデータシーケンスが完全に一致することはまれである．これは，歩く早さの違いや腕の振幅などといった動作の微妙な違いにより，時間軸に対して伸縮した異なるパターンとして計測されるためである．このようなデータに対するシーケンスマッチングの例としては，高い頻度で生じる動作パターンを検出することによるトレンド分析が挙げられる．これは，モーションシーケンスの高速なタグ付けなどに非常に有効となる．

従来，DTW は蓄積されたデータに対して利用されてきた．シーケンスの長さが固定された従来の設定と異なり，データストリームは頻繁にデータが生成されるため，シーケンス長は時間の経過とともに発達する．DTW を用いて部分的な類似を見つけるためには，データストリームをすべての考えられる部分シーケンスに分割し，それらすべての類似度を計算しなければならない．そのため，類似度を求めるための計算時間やメモリ使用量は大幅に増加する．通常，データストリーム処理ではこのような厳しい条件下での高速処理を達成するために，近似解を求めることが多い [13]．しかしながら，アルゴリズムは高い性能を維持しつつも，すべての正しい解を出力することが望ましい．

本論文では，データストリームにおける部分シーケンスマッチングのための手法である CrossMatch を提案する．CrossMatch は厳密に DTW に基づいた手法であり，類似する部分シーケンスペアをワンパスで特定することができる．理論的な分析を行い，CrossMatch が精度を犠牲にすることなく，条件を満たす解を検出することを証明する．更に，実データと人工データを用いた実験により，CrossMatch が類似する部分シーケンスペアを正しく特定し，計算時間とメモリ使用量を期待したとおりに改善することを示す．

本論文は次のように構成される．まず，**2.** で問題定義を行う．**3.** で提案手法である CrossMatch について述べ，**4.** において，CrossMatch の有効性を検証するための実験を行い，その結果を考察する．**5.** においてストリームマイニングの関連研究を示し，最後に **6.** でまとめを述べる．

2. 問題定義

本章では，シーケンス間の類似度を測定するダイ

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

ナミックタイムワーピング (DTW：Dynamic Time Warping) [3], [4] について説明し，本論文で解決したい問題について述べる．

2.1 ダイナミックタイムワーピング (DTW)

DTW は時間スケールを考慮した有用な距離尺度の一つである．シーケンス間の距離を最小化するように時間軸方向にシーケンス長を調整する性質をもっているため，シーケンス長やサンプリングレートが異なるシーケンス間の類似度を計算することができる．

二つのシーケンス間の DTW 距離は動的計画法によって計算され，二つのシーケンスを最適に調節した後の距離の合計となる．図 2 は DTW の計算方法を例示したものである．左図は，距離計算のための DTW による最適なシーケンスの対応付けを表している．距離計算には右図で示されるようなタイムワーピング行列が用いられる．距離値を計算するために対応づけられた要素同士の並びはワーピングパスと呼ばれ，右図では濃く色付けされたセルの集合として示されている．

長さ n のシーケンス $X=(x_1,x_2,...,x_n)$ と長さ m のシーケンス $Y=(y_1,y_2,...,y_m)$ を考える．これらの DTW 距離 $D(X,Y)$ は次のように定義される．

$$D(X,Y)=d(n,m)$$

$$d(i,j)=||x_i-y_j||+min\begin{cases}d(i,j-1)\\d(i-1,j)\\d(i-1,j-1)\end{cases}$$

$$d(0,0)=0,\quad d(i,0)=d(0,j)=\infty$$

$$(i=1,...,n;\ j=1,...,m) \qquad (1)$$

ここで，$||x_i-y_j||=(x_i-y_j)^2$ は二つの数値の距離を表す．L1 距離 ($||x_i-y_j||=|x_i-y_j|$) など，他の選択でもかまわない．本論文で提案するアルゴリズムは，これらの選択とは完全に独立したものである．

一方のシーケンスの小さな範囲と，もう一方のシーケンスの大きな範囲が対応することによるマッチングの精度低下を避けるため，ワーピング制約が導入される．これは，不要なワーピングパスを取り除き，ワーピングの範囲を制限するものであり，Sakoe-Chiba band [14] が代表的なワーピング制約の一つである．図 2 のグレーに色づけられたセルは，Sakoe-Chiba band におけるワーピングの範囲を表しており，具体的には $|i-j|\leq w$ という制限のもとで計算される．

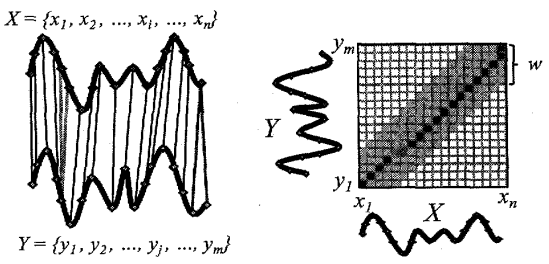


図 2 DTW による全体シーケンスマッチング
Fig. 2 Illustration of the sequence matching using DTW.

タイムワーピング行列は nm 個のセルから構成されるため，DTW の計算時間は $O(nm)$ となる．メモリ使用量については，タイムワーピング行列の 2 列（現在の列と直前の列）だけ必要となるため，各時刻当り $O(n)$ となる．ワーピング制約を導入した場合，タイムワーピング行列のセルが nw または mw 個に制限されるため，計算時間は $O(nw)$ または $O(mw)$ に削減される．また，このときのメモリ使用量は $2w$ 個のセルのみ保持するため，$O(w)$ となる．

2.2 Cross-similarity

データーストリーム X は，$x_1,x_2,...,x_n,...$ の値からなる半無限長のシーケンスである．x_n は最新の値であり，時間が進むごとに n は増加する．$X[i_s:i_e]$ は時刻 i_s から始まり i_e で終わる X の部分シーケンスであり，$X[i_s:i_e]$ の長さは $l_x=i_e-i_s+1$ となる．同様に，$Y[j_s:j_e]$ は時刻 j_s から始まり j_e で終わる Y の部分シーケンスであり，その長さは $l_y=j_e-j_s+1$ となる．本論文の目的は，DTW に基づき，データストリーム処理でシーケンス間の部分的な類似を見つけることである．より具体的には，次の条件を満たす部分シーケンスペアを検出する．

$$D(X[i_s:i_e],Y[j_s:j_e])\leq\varepsilon L(l_x,l_y) \qquad (2)$$

ここで，$D(X[i_s:i_e],Y[j_s:j_e])$ は部分シーケンスペア $X[i_s:i_e]$ と $Y[j_s:j_e]$ 間の DTW 距離であり，L は部分シーケンスペアの長さを表す関数である．本論文では，二つの部分シーケンスの平均長である $L(l_x,l_y)=(l_x+l_y)/2$ を用いるが，$L(l_x,l_y)=max(l_x,l_y)$ や $L(l_x,l_y)=min(l_x,l_y)$ を用いてもかまわない．DTW 距離は，対応づけられた要素の距離の合計で表されるため，部分シーケンスの長さが長くなるにつれてその値は増加する．そのため，類似判定のしきい値も部分シーケンス長に比例することが望ましいといえ，$\varepsilon L(l_x,l_y)$ を用いて判定を行う．

ここで，X と Y 間の部分的な類似について定義する．

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

[定義 1]（Cross-similarity） 二つのシーケンス X と Y，距離のしきい値 ε，適合する部分シーケンスの長さのしきい値 l_{min} が与えられたとき，次の条件を満たす部分シーケンスペア $X[i_s:i_e]$ と $Y[j_s:j_e]$ は cross-similarity の性質をもつ．

$$D(X[i_s:i_e],Y[j_s:j_e]) \leq \varepsilon(L(l_x,l_y)-l_{min}) \quad (3)$$

適合する部分シーケンスの長さのしきい値 l_{min} は，ユーザによって与えられる値である．定義 1 は，オリジナルの DTW [3], [4] に，適合するシーケンスの長さという概念を加えたものである．すなわち，部分シーケンスペアの長さ L が l_{min} 以上となる部分シーケンスペアを検出することを意味する．オリジナルの DTW は，ノイズなどによってシーケンス長の短い，意味のないシーケンスペアを適合ペアとして検出する可能性がある．そのため，これらを排除することにより，ユーザの真の要求を満たすシーケンスペアを検出する．

部分シーケンスペア $X[i_s:i_e]$ と $Y[j_s:j_e]$ が適合するとき，距離値が極小値をとる部分シーケンスペアに重複 (overlap) する多くの部分シーケンスペアも適合してしまう．重複とは部分シーケンスペア間のワーピングパスが交わることであり，次のように定義される．

[定義 2]（Overlap） X と Y の部分シーケンスペアのワーピングパス二つが与えられたとき，重複とはそれらのワーピングパスの少なくとも一つの要素が共有されることである．

重複する部分シーケンスペアは冗長な情報であり，不必要な結果についても報告することでアルゴリズムの処理速度が低下する可能性もある．本論文では，重複する部分シーケンスペアの中から距離値が最小値をとる部分シーケンスペアを検出する．すなわち，本論文において解決したい問題は，cross-similarity の最適解を見つけることである．

[問題 1] 二つのシーケンス X と Y，しきい値 ε，適合する部分シーケンスの長さのしきい値 l_{min} が与えられたとき，次の条件を満たす部分シーケンスペア $X[i_s:i_e]$ と $Y[j_s:j_e]$ を検出する．

（1） $X[i_s:i_e]$ と $Y[j_s:j_e]$ は cross-similarity の性質をもつ．

（2） 重複する部分シーケンスペアのグループの中で，$D(X[i_s:i_e],Y[j_s:j_e])-\varepsilon(L(l_x,l_y)-l_{min})$ が最小値をとる．

これ以降，問題 1 の条件 (1) を満たすものを適合する部分シーケンスペア，二つの条件を満たすものを最適な部分シーケンスペアと呼ぶ．

通常，データストリームの最新の要素は過去に到着した要素より重要であるとされている [15]．そこで，DTW にワーピング制約である Sakoe-Chiba band [14] を導入し，新たに到着した要素に注目する．すなわち，このワーピング制約を利用してタイムワーピング行列の中のセルを制限することにより，新しい要素に存在する cross-similarity を検出する．

本論文で示した問題は従来の時系列分析の問題とは異なり，シーケンス間の部分的な類似を DTW に基づいて検出することに焦点を当てている．理想的には，アルゴリズムはサンプリングレートが異なるシーケンスや時間周期が変化するシーケンスについてもロバストに対応し，データストリーム処理の中でも正しい解を出力することを保証するべきである．データストリーム処理は低メモリと高スピード処理が要求されるため，解決したい問題はより困難なものとなる．

3. 提案手法

本章では，データストリームにおける cross-similarity の最適解を見つけるための最も素直な解法を示し，提案手法である CrossMatch について述べる．

3.1 ナイーブな手法

cross-similarity を見つけるための最も素直な方法は，データストリーム X と Y の考えられるすべての部分シーケンスの組合せについて，それらの DTW 距離を計算することである．ここで，考えられるすべての部分シーケンスとは，$X[i_s:i_e]$ については $1 \leq i_s \leq n-1 \wedge i_s < i_e \leq n$，$Y[j_s:j_e]$ については $1 \leq j_s \leq m-1 \wedge j_s < j_e \leq m$ の範囲の部分シーケンスのことであり，ワーピング制約の範囲内にあるすべての要素を開始点とするタイムワーピング行列を作成し，その距離を計算することを意味する．本論文ではこの手法をナイーブな手法と呼ぶ．

二つのシーケンス $X=(x_1,...,x_i,...,x_n)$ と $Y=(y_1,...,y_j,...,y_m)$ が与えられたとき，X の i 番目の要素と Y の j 番目の要素から始まるタイムワーピング行列におけるセル (p,q) が示す距離を $d_{i,j}(p,q)$ とし，ワーピングの範囲を w とする．X と Y 間の部分シーケンスマッチングの距離は次のように計算される．

$$D(X[i_s:i_e],Y[j_s:j_e]) = d_{i_s,j_s}(l_x,l_y)$$

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

$$d_{i,j}(p,q) = ||x_{i+p-1} - y_{j+q-1}|| + d_{best}$$

$$d_{best} = min \begin{cases} d_{i,j}(p,q-1) \\ d_{i,j}(p-1,q) \\ d_{i,j}(p-1,q-1) \end{cases}$$

$$d_{i,j}(0,0) = 0, \quad d_{i,j}(p,0) = d_{i,j}(0,q) = \infty$$

$$(i = 1,...,n;\ p = 1,...,n-i+1;$$
$$j = 1,...,m;\ q = 1,...,m-j+1;$$
$$|i+p-(j+q)| \leq w) \qquad (4)$$

ナイーブな手法では，毎時刻，すなわち，1 データが到着するたびに新たなタイムワーピング行列を作成する必要がある．例えば，時刻 i で到着する x_i については，x_i と既に到着している Y の要素 y_{i-w} から y_i を開始点とするタイムワーピング行列が新たに作成される．そして，既に作成されているタイムワーピング行列を含めたすべての行列について，x_i と y_{i-w} から y_i に対応するセルの距離値が更新される．定義 1 を満たす部分シーケンスペアについては，重複する部分シーケンスペアのグループの中で $D(X[i_s:i_e],Y[j_s:j_e])-\varepsilon(L(l_x,l_y)-l_{min})$ が最小値となる部分シーケンスペア一つが選択される．これらのペアは，重複する部分シーケンスペアの存在が確認されなくなったときに報告される．

図 3 はナイーブな手法における cross-similarity の検出を示している．ナイーブな手法は，毎時刻，新たな行列を作成するため，$O(nw+mw)$ 個の行列を扱うことになる．更に，その一つひとつの行列において，毎時刻 $O(w)$ の距離値を更新する．そのため合計すると，$O(nw^2+mw^2)$ 個もの値を更新しなければならない．本論文では，この膨大なコストを精度を犠牲にすることなく劇的に削減できることを示す．

3.2 CrossMatch

本節では cross-similarity の最適解を見つける CrossMatch について述べる．CrossMatch は，cross-similarity を求めるための新しいスコアリング関数と位置行列，そしてそれらを用いたストリーム処理アルゴリズムの三つのアイデアに基づいている．

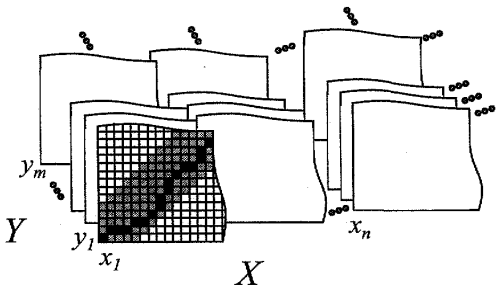


図 3 ナイーブな手法における cross-similarity の検出
Fig. 3 Detection of cross-similarity in naive solution.

3.2.1 スコアリング関数

前節で示したように，ナイーブな手法は，時間が経つにつれ，距離計算する部分シーケンスペアが増加する．定義 1 より，距離判定のしきい値は部分シーケンス長に比例する．これは，現時刻で不適合な部分シーケンスペアであっても今後検出される可能性があることを意味しており，適合する可能性のある部分シーケンスペアとそうでない部分シーケンスペアを切り分けることは困難である．しかしながら，もしこれらを切り分けることができれば，必要な部分シーケンスペアのみを追い続ければよく，距離計算は大幅に効率化される．

この目的を達成するため，DTW 距離を直接計算するのでなく間接的に計算する，スコアリング関数を提案する．スコアリング関数は，DTW と同様に動的計画法に基づいて類似度の計算を行うが，結果をスコアとして出力する点が異なる．スコアとは類似度を計算する行列（スコア行列と呼ぶ）の各要素の累積値として表され，DTW では距離値が小さいほど類似度が高いのに対し，スコアリング関数ではスコア値が大きいほど類似度が高いという逆の性質をもっている．スコアリング関数は，以下のように定義される．

[定義 3]（スコアリング関数） 二つのシーケンス $X=(x_1,\ldots,x_i,\ldots,x_n)$ と $Y=(y_1,\ldots,y_j,\ldots,y_m)$ が与えられたとき，$X[i_s:i_e]$ と $Y[j_s:j_e]$ のスコア値 $V(X[i_s:i_e],Y[j_s:j_e])$ は次のように計算される．

$$V(X[i_s:i_e],Y[j_s:j_e]) = v(i_e,j_e)$$

$$v(i,j) = max \begin{cases} 0 \\ \varepsilon b_v - ||x_i - y_j|| + v(i,j-1) \\ \varepsilon b_h - ||x_i - y_j|| + v(i-1,j) \\ \varepsilon b_d - ||x_i - y_j|| + v(i-1,j-1) \end{cases}$$

$$v(0,0) = v(i,0) = v(0,j) = 0$$
$$(i = 1,...,n; j = 1,...,m; |i-j| \leq w) \qquad (5)$$

スコア値を計算するために対応づけられた要素同士の並びが，スコア行列におけるワーピングパスである．スコア値は距離のしきい値 ε と各要素の距離値の差を累積することにより決定されるため，部分シーケンスペアが条件を満たさない場合には，スコア値は負の値

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

を示す．そのため，不適合な部分シーケンスペアをスコアの値から判断することが可能となる．また，負の値となった要素のスコアをゼロでリセットすることで，その要素から新たにスコア計算を開始する．この性質を利用して，不適合な部分シーケンスペアを枝刈りし，適合する可能性の高い部分シーケンスペアのみを一つの行列で効率的に計算することが可能となる．

更に重要なことに，スコアリング関数により出力されるスコア値は，DTW 距離に変換可能であることが保証されている（これ以降，この性質を可逆と表記する）．式 (5) において，b_v，b_h，b_d は L によって決定される重みである．例えば，$L(l_x, l_y) = (l_x + l_y)/2$ において，$b_v = b_h = 1/2$，$b_d = 1$ となる(注1)．これは，垂直または水平方向の要素が引き継がれた場合，シーケンス長は 1/2 増加し，対角方向の要素が引き継がれた場合は 1 増加するためである．スコア行列の中での各ワーピングパス上の重み (b_v，b_h，b_d) の合計は，$L(l_x, l_y)$ と等しくなるように設計されており，DTW とスコアリング関数の可逆の関係が保証される（詳細は **3.3.1** に示す）．部分シーケンスペアの DTW 距離は，次の式により計算される．

$$D(X[i_s:i_e], Y[j_s:j_e]) = \varepsilon L(l_x, l_y) - V(X[i_s:i_e], Y[j_s:j_e]) \tag{6}$$

3.2.2 位置行列

スコアリング関数により，部分シーケンスペア $X[i_s : i_e]$ と $Y[j_s : j_e]$ 間の cross-similarity の存在を知ることが可能となった．しかし，このアイデアのみでは問題を解決することはできない．単にスコアリング関数のみを用いた場合，部分シーケンスペアの開始点に関する情報が失われてしまう．すなわち，一つの行列のみで適合する部分シーケンスペアのスコアを得ることができるが，どの部分シーケンスペアが最大スコア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 のアイデアとして，部分シーケンスペアの開始点を保持するための位置行列を導入する．

［定義 4］（位置行列） スコア $v(i,j)$ に対応する位置行列の開始点 $s(i,j)$ は以下のように求められる．

$$s(i,j) = \begin{cases} s(i,j-1) & (v(i,j-1) > 0 \wedge v(i,j) \\ & = b_v\varepsilon + v(i,j-1) - \|x_i - y_j\|) \\ s(i-1,j) & (v(i-1,j) > 0 \wedge v(i,j) \\ & = b_h\varepsilon + v(i-1,j) - \|x_i - y_j\|) \\ s(i-1,j-1) & (v(i-1,j-1) > 0 \wedge v(i,j) \\ & = b_d\varepsilon + v(i-1,j-1) - \|x_i - y_j\|) \\ (i,j) & (\text{otherwise}) \end{cases} \tag{7}$$

開始点は座標値として表され，$s(i_e, j_e)$ は，$X[i_s:i_e]$ と $Y[j_s:j_e]$ のマッチングの開始点 (i_s, j_s) を指し示す．スコア計算と開始点の更新は，各々の行列において完全に同じワーピングパス上を伝搬するため，データストリーム処理においても最大スコアを示す最適な部分シーケンスペアの開始点を得ることができる．

また，位置行列により，重複する部分シーケンスペアの判断が容易になる．二つの部分シーケンスペアのワーピングパスが同じ要素を共有するとき，これらの部分シーケンスペアは同じ開始点を保持する．これは，位置行列において，同じ要素を共有した時点でその要素の開始点が引き継がれていくためである．そのため，開始点の一致を見ることで重複する部分シーケンスの存在を確認することが可能となる．

3.2.3 ストリーム処理アルゴリズム

最後に，上記で述べたスコアリング関数とスコア行列，位置行列をデータストリーム処理に適用する．図 4 はストリーム処理アルゴリズムである CrossMatch を示している．最新のデータ x_n が到着するたびに，ス

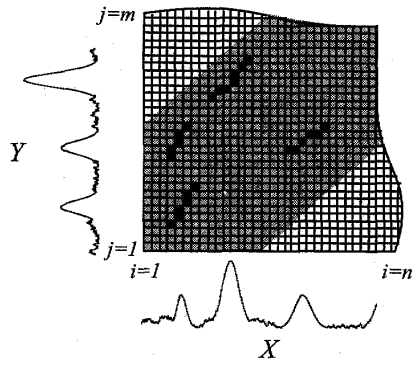


図 4 CrossMatch による cross-simirality の検出
Fig. 4 Detection of cross-similarity in CrossMatch.

(注1)：$L(l_x, l_y) = max(l_x, l_y)$ において，$l_x > l_y$ の場合，$b_d = b_h = 1$，$b_v = 0$，$l_x < l_y$ の場合，$b_d = b_v = 1$，$b_h = 0$，$l_x = l_y$ の場合，$b_d = 1$，$b_v = b_h = 0$ となる．$L(l_x, l_y) = min(l_x, l_y)$ についても同様に決定される．

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

コア $v(n,j)$ と開始点 $s(n,j)$ を式 (5) と (7) を用いてインクリメンタルに計算する．そして cross-similarity の条件（定義 1）を満たす部分シーケンスペアを検出すると，そのスコア $C'_v := v(n,j)$，開始点 $C'_s := s(n,j)$，終了点 $C'_e := (n,j)$ を候補集合の配列 $\mathcal{S}$ に格納する．重複する部分シーケンスペアが存在する場合，候補配列には重複する候補ペアのグループの中で $v(n,j) - \varepsilon l_{min}$ が最大値となるペアのみが格納される．部分シーケンスペアの重複は，配列 $\mathcal{S}$ に含まれるすべての開始点（この集合を S_{c_s} とする）と現在の開始点 C'_s が一致するかどうかによって判断される．

候補配列には，開始点が異なる複数の最適な部分シーケンスペア（この時点では候補部分シーケンスペア）の情報 C が保持されている（すなわち，スコア C_v，開始点 C_s，終了点 C_e）．CrossMatch は以下の二つの条件を満たすとき，候補部分シーケンスペアを cross-similarity の最適解として報告する．

$$(\forall_i, s(i,m) \neq C_s) \ \wedge \ (\forall_j, s(n,j) \neq C_s)$$

これは，候補部分シーケンスペアが今後出現する部分シーケンスペアによって置き換わることがないことを意味する．最終的な部分シーケンスペアの類似度は式 (6) により DTW 距離 d_{min} として報告される．

アルゴリズム 1 に詳細を示す．このアルゴリズムでは，時刻 n で X の要素 x_n を受け取った場合の処理に焦点を当てているが，時刻 m で受け取った Y の要素 y_m の処理についても，同様に計算することができる．

図 5 は，$X = (5,12,6,10,6,4,20)$，$Y = (11,6,9,4,2,9,12)$ の二つのシーケンスについて，距離のしきい値 $\varepsilon = 14$，適合する部分シーケンスの長さ $l_{min} = 3$，ワーピングの範囲 $w = 3$ を設定した場合に計算されるスコア行列と位置行列の例である．アルゴリズムはセル (i,j) にスコア $v(i,j)$ と開始点 $s(i,j)$ を保持する．濃く色付けされたセルは最適な類似部分シーケンスペアを表しており，薄く色付けされたセルは適合する部分シーケンスペアを表している．

ここで，図 5 を用いてアルゴリズムの動作を説明する．説明を単純にするために，x_i と y_j が交互に到着すると想定する．$j = 1$ で，アルゴリズムはスコアと開始点の計算を開始する．$i = 5$ において，$v(5,4) = 50 \geq \varepsilon l_{min}$ となる $(2,1)$ から始まる適合する部分シーケンスペア $X[2:5]$ と $Y[1:4]$ を検出する．$j = 5$ において，条件を満たす部分シーケンスペアは検出されないが，これから出現する部分シーケンスペアが最適な部分シーケンスペアになる可能性があるため，$X[2:5]$ と $Y[1:4]$ を報告しない．$i = 6$ において，これらのグループの最適な部分シーケンスペア $X[2:6]$ と $Y[1:5]$ を検出する．最終的に $j = 7$ において，今後出現する部分シーケンスペアが最適な部分シーケン

Algorithm 1 CrossMatch

Input: new value x_n
Output: optimal subsequence pairs and DTW distances

```
//Detect optimal subsequence pairs.
for j := m − w to m do
  C'_v := v(n, j);  //Score value by Equation (5)
  C'_s := s(n, j);  //Starting position by Equation (7)
  C'_e := (n, j);  //End position
  if C'_v ≥ εl_min then
    //Add the subsequence as a new candidate.
    if C'_s ∉ S_{c_s} then
      add C'_v, C'_s, and C'_e to S;
    else
      for each candidate C ∈ S do
        //Overwrite the maximum score.
        if C'_s = C_s ∧ C'_v ≥ C_v then
          C_v := C'_v;
          C_e := C'_e;
        end if
      end for
    end if
  end if
end for
//Report the optimal subsequences.
for each candidate C ∈ S do
  if (∀_i, s(i, m) ≠ C_s) ∧ (∀_j, s(n, j) ≠ C_s) then
    d_min := εL(l_x, l_y) − C_v;
    Report d_min, C_s and C_e;
    Remove C from S;
  end if
end for
```

$y_7 = 12$				35	10	0	0
$y_6 = 9$			17	32	39	42	0
$y_5 = 2$		0	19	0	41	[illegible]	0
$y_4 = 4$	13	0	28	11	50	57	0
$y_3 = 9$	4	18	25	40	38	20	
$y_2 = 6$	13	0	27	18	27		
$y_1 = 11$	0	13	0	13			
y_j \ x_i	$x_1 = 5$	$x_2 = 12$	$x_3 = 6$	$x_4 = 10$	$x_5 = 6$	$x_6 = 4$	$x_7 = 20$

(a) スコア行列

$y_7 = 12$				(1, 2)	(1, 2)	(6, 7)	(7, 7)
$y_6 = 9$			(1, 2)	(1, 2)	(2, 1)	(2, 1)	(7, 6)
$y_5 = 2$		(2, 5)	(1, 2)	(4, 5)	(2, 1)	[illegible]	(7, 5)
$y_4 = 4$	(1, 4)	(2, 4)	(1, 2)	(2, 1)	(2, 1)	(2, 1)	(7, 4)
$y_3 = 9$	(1, 2)	(1, 2)	(2, 1)	(2, 1)	(2, 1)	(2, 1)	
$y_2 = 6$	(1, 2)	(2, 2)	(2, 1)	(2, 1)	(4, 1)		
$y_1 = 11$	(1, 1)	(2, 1)	(3, 1)	(4, 1)			
y_j \ x_i	$x_1 = 5$	$x_2 = 12$	$x_3 = 6$	$x_4 = 10$	$x_5 = 6$	$x_6 = 4$	$x_7 = 20$

(b) 位置行列

図 5 cross-similarity の検出の例
Fig. 5 Example of detecting cross-similarity.

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

スペアとなることはないことが確認され，部分シーケンスペア $X[2:6]$ と $Y[1:5]$ を cross-similarity の最適解として報告する．

3.3 理論的な分析

3.3.1 精　　度

[補題 1]　二つのシーケンス X と Y が与えられたとき，問題 1 の二つの条件は次の条件と等価である．

（1）　$V(X[i_s:i_e],Y[j_s:j_e]) \geq \varepsilon l_{min}$

（2）　ワーピングパスが交差する部分シーケンスペアのグループの中で，$V(X[i_s:i_e],Y[j_s:j_e])-\varepsilon l_{min}$ が最大値をとる．

(証明 1)　タイムワーピング行列とスコア行列の双方で，開始点 (i_s,j_s)，終了点 (i_e,j_e) であるワーピングパスが (i_e,j_e-1) を通るとするならば，式 (4) より

$$||x_{i_e}-y_{j_e}|| = d_{i_s,j_s}(l_x,l_y)-d_{i_s,j_s}(l_x,l_y-1).$$

また，$b_v=L(l_x,l_y)-L(l_x,l_y-1)$ から，式 (5) より，

$$\begin{aligned}||x_{i_e}-y_{j_e}|| &= \varepsilon L(l_x,l_y)-v(i_e,j_e)\\ &\quad -\varepsilon L(l_x,l_y-1)+v(i_e,j_e-1)\end{aligned}$$

が成り立つ．同様に，(i_e-1,j_e) と (i_e-1,j_e-1) を通るときは各々

$$\begin{aligned}||x_{i_e}-y_{j_e}|| &= d_{i_s,j_s}(l_x,l_y)-d_{i_s,j_s}(l_x-1,l_y)\\ &= \varepsilon L(l_x,l_y)-v(i_e,j_e)\\ &\quad -\varepsilon L(l_x-1,l_y)+v(i_e-1,j_e)\\ ||x_{i_e}-y_{j_e}|| &= d_{i_s,j_s}(l_x,l_y)-d_{i_s,j_s}(l_x-1,l_y-1)\\ &= \varepsilon L(l_x,l_y)-v(i_e,j_e)\\ &\quad -\varepsilon L(l_x-1,l_y-1)+v(i_e-1,j_e-1)\end{aligned}$$

が成り立つ．

$$||x_{i_s}-y_{j_s}|| = d_{i_s,j_s}(1,1) = \varepsilon - v(i_s,j_s)$$

であるため，タイムワーピング行列とスコア行列は同じワーピングパスを共有する．また，ワーピングパス上の重みの合計は $L(l_x,l_y)$ であるため，式 (5) より

$$v(i_e,j_e) = \varepsilon L(l_x,l_y)-d_{i_s,j_s}(i_e,j_e)$$

が成り立ち，DTW 距離とスコアの関係式

$$d_{i_s,j_s}(l_x,l_y) = \varepsilon L(l_x,l_y)-v(i_e,j_e) \qquad (8)$$

が導かれる．更に，式 (3) と (8) より，

$$v(i_e,j_e) \geq \varepsilon l_{min}$$

が得られる．

タイムワーピング行列では，問題 1 の条件 (2) より，(i_s,j_s) から (i_e,j_e) までの最適ワーピングパスはワーピングパスが交差する部分シーケンスペアのグループの中で最小距離を示すことが明らかである．また式 (5) と式 (8) より，スコア行列でも同じワーピングパスが選択され，最大スコアを出力することが分かる．よって，問題 1 の二つの条件と等価となる．□

[補題 2]　CrossMatch は，cross-similarity を満たす最適な部分シーケンスペアを検出する．

(証明 2)　最適な部分シーケンスペア $X[i_s:i_e]$ と $Y[j_s:j_e]$ のワーピングパスの開始点を $C_s=(i_s,j_s)$ とする．$s(n,m)\neq C_s$ であるとき，部分シーケンスペア $X[i:n]$ と $Y[j:m]$ のワーピングパスは最適なワーピングパスと重複しない $(1\leq i\leq n, 1\leq j\leq m)$．同様に，以下の条件を満たすとき，今後出現する部分シーケンスペアは候補配列の中の部分シーケンスペアと重複することはない．

$$(\forall_i, s(i,m)\neq C_s) \ \wedge \ (\forall_j, s(n,j)\neq C_s).$$

CrossMatch は上記の条件を満たすときのみ $X[i_s:i_e]$ と $Y[j_s:j_e]$ を最適な部分シーケンスペアとして報告する．よって，CrossMatch は最適な部分シーケンスペアを見落とすことはない．□

3.3.2 計　算　量

二つの伸張しているシーケンスを X，Y，それぞれの長さを n，m，ワーピングの範囲を w とする．

[補題 3]　ナイーブな手法は，1 データ処理当り $O(nw^2+mw^2)$ のメモリ量と $O(nw^2+mw^2)$ の計算時間を要する．

(証明 3)　最適な部分シーケンスペアを特定するために，ナイーブな手法は $O(nw+mw)$ 個のタイムワーピング行列を用いる．時刻 i において x_i を受け取った場合，タイムワーピング行列の x_i と y_{i-w} から y_i 間の $O(w)$ 個の距離値が $O(nw)$ 個の行列に対して計算されるため，$O(nw^2)$ 個の値を更新する必要がある．同様に，時刻 j において y_j を受け取った場合には，$O(mw)$ 個の行列に対して $O(w)$ 個の距離値が計算されるため，$O(mw^2)$ 個の値を更新する必要がある．そのため，$O(nw^2+mw^2)$ の計算時間を要する．メモリ使用量については，x_i を受け取った場合の距離値の更新において，直前の列の距離値も必要となるため，$O(w)$ 個の配列 2 列（現在の列と直前の列）を保持する．y_j を受け取った場合にも同様の配列が必要となる

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

ため，$O(nw^2 + mw^2)$ となる． □

［補題 4］ 提案手法は，1 データ処理当り $O(w)$ のメモリ使用量と $O(w)$ の計算時間を要する．

（証明 4） 最適な部分シーケンスペアを特定するために，提案手法は 2 個の行列（スコア行列と位置行列）を用いる．時刻 i において x_i を受け取った場合には $O(w)$ 個の値を，時刻 j において y_j を受け取った場合には $O(w)$ 個の値を更新する．そのため，$O(w)$ の計算時間を要する．また，メモリ使用量については，$O(w)$ 個の配列 2 列が x_i，y_j それぞれを受け取った場合のスコア計算に必要となり，$O(w)$ となる． □

4. 評価実験

CrossMatch の有効性を検証するため，実データと人工データを用いた実験を行った．実験には，4 GByte のメモリと Intel Core2 Duo E6600 2.4 GHz の CPU を搭載した Linux マシンを用いた．本実験は，以下の問いに答えるように構成されている．

（1） 重複する部分シーケンスペアのフィルタリングはどのような影響をもたらすのか？

（2） CrossMatch は cross-similarity の検出にどう成功するのか？

（3） シーケンスの長さを変化させたとき，計算時間とメモリ使用量の観点で CrossMatch はどのようにスケールするのか？

4.1 冗長な情報のフィルタリング

3.2 で述べたように，CrossMatch は重複する部分シーケンスペアの中から最適な部分シーケンスペア（すなわち，cross-similarity の最適解）を検出する．問題 1 の条件 1 を満たす部分シーケンスペア（適合する部分シーケンスペア）には，冗長な情報が多く含まれており，多くのペアが重複している．冗長な情報のフィルタリングについて，CrossMatch の有効性を検証するために，二つの出力について比較を行った．

- Overlap: アルゴリズムは適合する部分シーケンスペアをすべて出力する．
- CrossMatch: アルゴリズムは，最適な部分シーケンスペアのみを出力する．

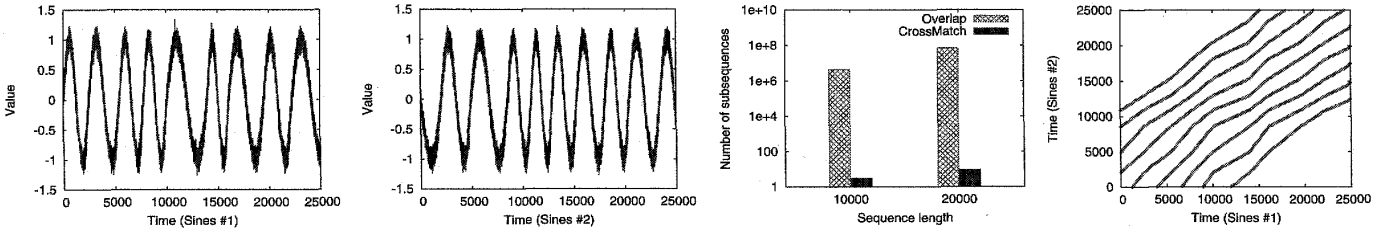
この実験のために，ホワイトノイズを含むサイン波から構成される人工データ *Sines* を用い（図 6 (a), (b)），適合する部分シーケンスの長さ l_{min} をシーケンス長の 15%，距離のしきい値 ε を $1.0e{-}2$ に，ワーピングの範囲 w をシーケンス長の 50%に設定した．

図 6 (c) は，検出された部分シーケンスペアの数を示している．シーケンス長が増加するに伴い，検出される部分シーケンスの数は増加する．特に Overlap は各シーケンス長において，100 万以上もの部分シーケンスペアが検出される．一方 CrossMatch は cross-similarity の最適解のみ出力するため，最適な部分シーケンスペアの数は適合する部分シーケンスペアの数より大幅に削減される．そのため，ユーザは真に必要な情報のみを獲得することが可能となる．

図 6 (d) は，CrossMatch によって検出された cross-similarity の最適解をワーピングパスとして示したものである．ワーピングパスは，部分シーケンスペアの終了点から選択されたセルを逆にたどることで求められる．例えば，部分シーケンスペア $X[i_s{:}i_e]$ と $Y[j_s{:}j_e]$ が適合したとき，(i_s, i_e) から (j_s, j_e) までのセルが表示される．この図から，CrossMatch が連続するサイン波を完全に特定していることが確認される．重複する部分シーケンスペアをフィルタリングすることにより，cross-similarity の関係が明らかになっている．

4.2 cross-similarity の検出

データストリーム間の最適な部分シーケンスペアの検出について，実データと人工データを用いたケーススタディを示す．各データセットの詳細と実験の設定は表 1 のとおりである．その他のパラメータとして，l_{min} は *RamdomSines* については 500，それ以外のデータ



(a) データシーケンス (*Sines #1*)　(b) データシーケンス (*Sines #2*)　(c) 検出された部分シーケンスペアの数　(d) CrossMatch による cross-similarity の発見

図 6 重複する部分シーケンスペアのフィルタリングの影響
Fig. 6 Effect of filtering overlapping subsequence pairs.

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

表 1　データセットの詳細と実験パラメータ設定
Table 1　Detail of datasets and experimental parameter settings.

Datasets	Sequence length		ε
	Seq. #1	Seq. #2	
RandomSines	25000	25000	1.0e-4
Automobile traffic	16000	16000	8.5e+4
Web	32000	32000	4.0e+4
Sunspots	18000	18000	3.0e+2
Temperature	30411	26151	2.0e-1

セットについてはシーケンス長の 15%に，w はすべてのデータセットについてシーケンス長の 50%に設定した．なお，実験結果を示す図 7 は，左と中央の図がデータセットを，右の図が検出された cross-similarity の最適なワーピングパスを示している．

4.2.1　*RandomSines*

RandomSines はホワイトノイズをもつ不連続のサイン波から構成された人工データである（図 7(a)）．各シーケンスにおけるサイン波の周期は一致しておらず，サイン波が現れる間隔も異なっている．図 7(a) の右図に示すとおり，CrossMatch はすべてのサイン波を完全に特定していることが分かる．各サイン波の周期の違いは，描画の傾きの違いとして現れている．

4.2.2　*Automobile traffic*

図 7(b) は自動車交通量の時系列データであり，1 日の周期と朝夕のラッシュアワーを示す半日周期が存在する．時間単位の交通量はバースト的であり，ホワイトノイズとみなすことができる．

CrossMatch は高周波である時間単位の交通量に惑わされることなく，1 日の周期を完璧に検出することに成功している．図 7(b) の右図において，描画線が連続していることと，それらが一定間隔で現れていることが，1 日の周期が繰り返されている様子を表している．また，描画線同士の間隔が 1 日の周期と対応しており，このデータの特徴がよく反映されている．

4.2.3　*Web*

Web は，Web サイトにおけるアクセス数を 10 秒ごとに記録したデータセットである．各サイトごとにアクセスパターンが異なる中，CrossMatch により cross-similarity が検出されたのが，メールサイトとブログサイトのアクセスパターンである（図 7(c)）．これらのデータはアクセス数のスケールが若干異なるが，朝からアクセスが上昇し，夜にかけてピークを迎えるという，よく似た 1 日の周期が存在する．

実験結果から，CrossMatch はこれらの検出にほぼ成功していることが確認される．*Automobile traffic* と異なり，cross-similarity の結果は曲がりくねった線を描いている．これは，CrossMatch がデータシーケンスの要素同士を時間軸方向に伸張させながら最適に対応づけたためであり，CrossMatch の時間スケールを考慮するという特徴によって cross-similarity が検出されていることが分かる．

4.2.4　*Sunspots*

図 7(d) のデータ集合は，1 日ごとの太陽の黒点数を記録したものである．太陽の黒点には周期性があることがよく知られており，太陽の活動とも密接に関連している．太陽活動が活発なときは黒点が多く出現し，逆に太陽活動が不活発なときは黒点が減少する．この変化は約 11 年の周期で増減する．CrossMatch は黒点数の各周期の増減を区別し，類似する変化を捉えている．

4.2.5　*Temperature*

Temperature は温度センサを使って 1 分間隔に約 16 日間取得されたデータである（図 7(e)）．これらのデータセットは多くの時刻で測定値が欠けており，それぞれの長さは異なっている．*Temperature # 1* と *# 2* は天候によって 18°C から 32°C まで大きく変動しているパターンが存在している．多くの欠損値を含むにもかかわらず，CrossMatch は温度が大きく変動している日を cross-similarity として捉えている．

4.3　性　　能

3.3.2 で議論した計算量を検証するために，シーケンス長を変化させた場合の計算時間，メモリ使用量を測定した．各実験では，CrossMatch，ナイーブな手法 (Naive)，SPRING の 3 手法についての比較を行っている．SPRING は文献 [1] で提案され，データストリームから固定長の問合せシーケンスに類似する部分シーケンスを検出するための手法である．SPRING は $O(n+m)$ 個の行列を用い，各時刻において $O(w)$ 個の値を更新する．そのため，cross-similarity を検出するために，1 データ処理当り $O(nw+mw)$ のメモリ使用量と $O(nw+mw)$ の時間が必要となる．実験には *Automobile traffic* データセットを用いた．

図 8 は計算時間に関する比較結果である．結果は，各シーケンス長において，1 データ処理当りに行列を更新する時間のみを測定し，その平均を計算時間として示している．実験結果から，CrossMatch の性能が，ナイーブな手法，SPRING と比べて非常に高いことが分かる．この傾向は **3.3.2** における理論的な議論と合致している．

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

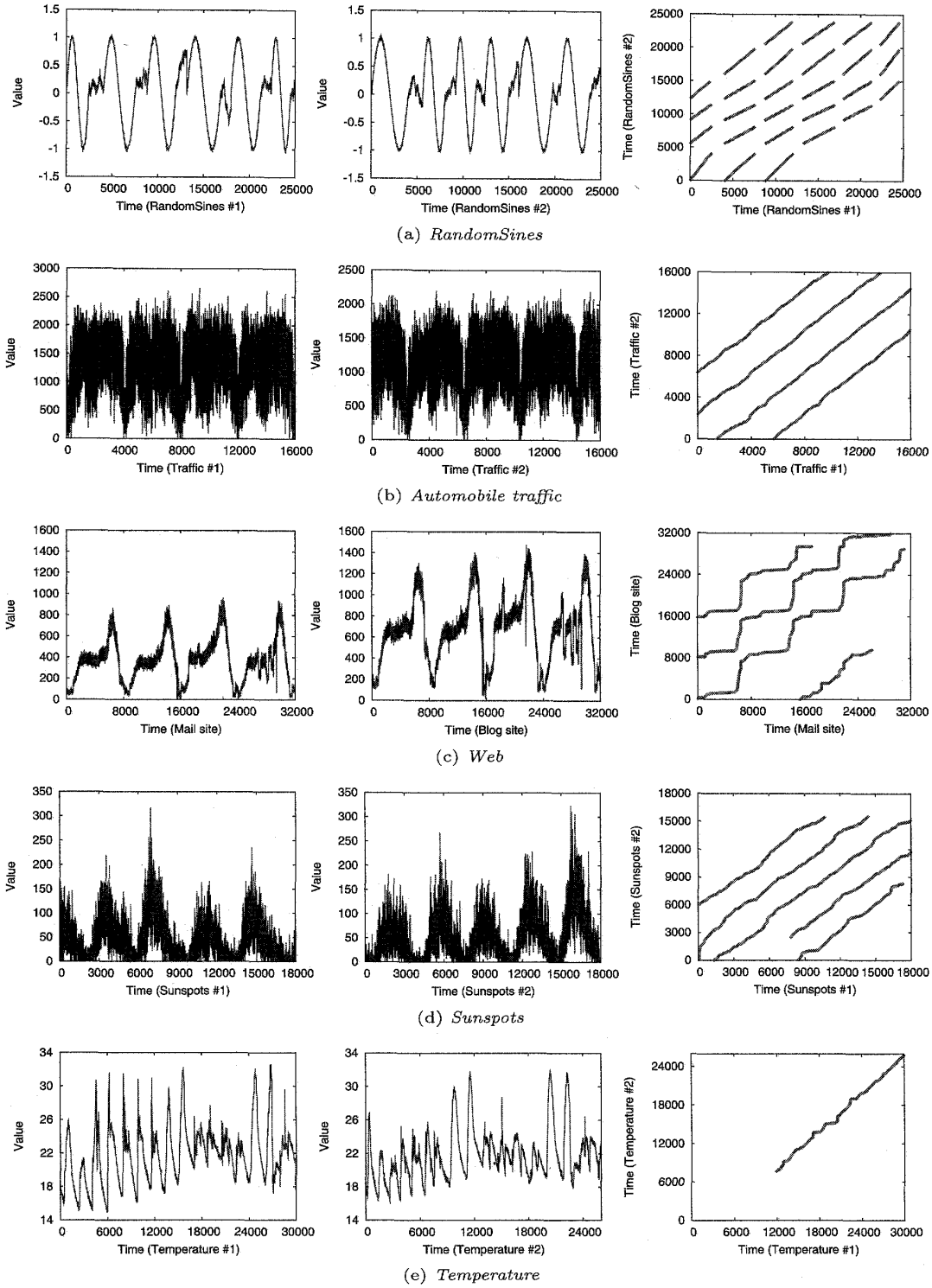


図 7 *RandomSines*, *Automobile traffic*, *Web*, *Sunspots*, *Temperature* を用いた cross-similarity の発見

Fig. 7 Discovery of cross-similarity using *RandomSines*, *Automobile traffic*, *Web*, *Sunspots*, and *Temperature*.

図 9 は，各行列のメモリ使用量を比較した結果である．図の横軸は，二つのデータストリームのシーケンス長を表している．CrossMatch のメモリ使用量はシーケンス長に対して一定であり，大幅な低減化を達成している．また，これらの結果も，**3.3.2** で議論したとおりのメモリ使用量であることが確認される．

5. 関連研究

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，ストリームのトレンド検出 [16] や相関検出 [17]〜

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

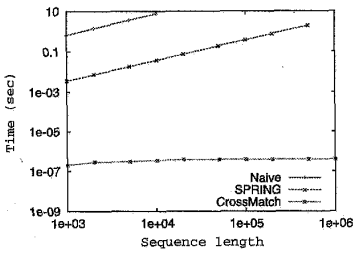


図 8　データシーケンス長に対する計算時間
Fig. 8　Wall clock time as a function of a sequence length.

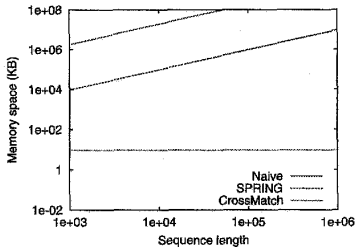


図 9　データシーケンス長に対するメモリ使用量
Fig. 9　Memory space consumption as a function of a sequence length.

[19]，予測 [20] など，様々な手法が提案されている．特に，データストリームにおけるシーケンスマッチングの研究としては，Chen らは時間周期のずれや振幅のスケールの違いを考慮する距離関数 SpADe (Spatial Assembling Distance) を提案し，SpADe を用いて問合せシーケンスに類似する部分シーケンスを検出する問題を扱った [21]．Zhou らは DTW 距離に基づいてオンラインで部分シーケンスを検出する手法を提案している [2]．SPRING 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であり，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 [1]．これらの手法は問合せシーケンスに類似する部分シーケンスを検出するためには有用であるが，本論文で扱う問題に適用した場合には計算量とメモリ使用量が増加する．

データストリームにおける cross-similarity 検出に関して，文献 [22] において類似する部分シーケンスペアを検出するアルゴリズムが提案されている．アルゴリズム的な観点から見て関連するのは文献 [23], [24] であり，DTW に基づく cross-similarity 検出のためのアルゴリズムが示されている．しかしながら，文献 [22] のアルゴリズムは正しい解の出力を保証しておらず，冗長な情報も含めた部分シーケンスペアを検出する．また，これらすべてのアルゴリズムは，シーケンス長に対して計算時間とメモリ使用量が線形に増加するという課題を残している．本論文で示した CrossMatch は，これらの課題を克服し，効果的かつ効率的に cross-similarity を特定する．

バイオインフォマティクス分野では，遺伝子や DNA 配列などのシーケンスのための探索アルゴリズムが研究され，Smith-Waterman アルゴリズムが最適なローカルアラインメントを見つけるために利用されている [25]．この分野の研究は，記号で表されるシーケンスに対するシーケンスマッチングに焦点を当てている．一方我々は，数値シーケンスに焦点を当てており，DTW 距離を正確に求め，距離値が最小となるペアの検出が保証されているという点で大きく異なっている．

本論文で提案した手法は，従来の手法と主に二つの点で異なっている．第 1 に，我々の手法は厳密に DTW に基づく手法であり，正しい解の検出が保証されていることである．第 2 に，我々の手法はデータストリーム処理における cross-similarity の検出を扱っていることである．一方，これまでの多くの研究は，問合せシーケンスに類似する部分シーケンスを検出することに焦点を当てている．本論文で明らかになった知見は，部分シーケンスマッチングを利用したアプリケーション開発の幅を広げるものであると信じている．

6. むすび

本論文では，データストリームにおける cross-similarity の問題を扱い，その問題を解決するための手法である CrossMatch を提案した．CrossMatch は DTW に基づく手法であり，cross-similarity を検出するための新しいスコアリング関数，位置行列，それらを用いたストリーム処理アルゴリズムの三つのアイデアから成り立っている．CrossMatch は探索漏れを発生させることなく類似する部分シーケンスペアを見つけることができるワンパスアルゴリズムであり，計算時間とメモリ使用量を劇的に低減化する．理論的な分析と，実データ，人工データを用いた実験により，CrossMatch の有効性が確認された．

文　献

[1] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," ICDE, pp.1046–1055, April 2007.
[2] M. Zhou and M.H. Wong, "Efficient online subsequence searching in data streams under dynamic time warping distance," ICDE, pp.686–695, April 2008.
[3] D.J. Berndt and J. Clifford, "Finding patterns in time series: A dynamic programming approach," Ad-

NII-Electronic Library Service

The Institute of Electronics, Information and Communication Engineers

vances in Knowledge Discovery and Data Mining, pp.229–248, 1996.

[4] L. Rabiner and B.-H. Juang, Fundamentals of speech recognition, Prentice Hall, Englewood Cliffs, NJ, 1993.

[5] J.-S.R. Jang and H.-R. Lee, "Hierarchical filtering method for content-based music retrieval via acoustic input," Multimedia, pp.401–410, Oct. 2001.

[6] H. Kawasaki, T. Yatabe, K. Ikeuchi, and M. Sakauchi, "Automatic modeling of a 3d city map from real-world video," Multimedia, vol.1, pp.11–18, Nov. 1999.

[7] D.W. Mount, Bioinformatics: Sequence and genome analysis, Cold Spring Harbor Laboratory, 2000.

[8] Y. Sakurai, M. Yoshikawa, and C. Faloutsos, "Ftw: Fast similarity search under the time warping distance," PODS, pp.326–337, June 2005.

[9] O. Nasraoui, C. Rojas, and C. Cardona, "A framework for mining evolving trends in web data streams using dynamic learning and retrospective validation," Comput. Netw., vol.50, no.10, pp.1488–1512, July 2006.

[10] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan, "Detecting time series motifs under uniform scaling," KDD, pp.844–853, Aug. 2007.

[11] E. Keogh, T. Palpanas, V. Zordan, D. Gunopulos, and M. Cardle, "Indexing large human-motion databases," VLDB, pp.780–791, Sept. 2004.

[12] G. Liu, J. Zhang, W. Wang, and L. McMillan, "A system for analyzing and indexing human-motion databases," SIGMOD, pp.924–926, June 2005.

[13] M.M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Mining data streams: A review," SIGMOD Rec., vol.34, pp.18–26, June 2005.

[14] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Trans. Acoust. Speech Signal Process., vol.26, no.1, pp.43–49, Feb. 1978.

[15] G. Cormode, F. Korn, and S. Tirthapura, "Time-decaying aggregates in out-of-order streams," PODS, pp.89–98, June 2008.

[16] S. Papadimitriou and P.S. Yu, "Optimal multi-scale patterns in time series streams," SIGMOD, pp.647–658, June 2006.

[17] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," VLDB, pp.697–708, Aug. 2005.

[18] Y. Zhu and D. Shasha, "Statstream: Statistical monitoring of thousands of data streams in real time," VLDB, pp.358–369, Aug. 2002.

[19] Y. Sakurai, S. Papadimitriou, and C. Faloutsos, "Braid: Stream mining through group lag correlations," SIGMOD, pp.599–610, June 2005.

[20] S. Papadimitriou, A. Brockwell, and C. Faloutsos, "Adaptive, hands-off stream mining," VLDB, pp.560–571, Sept. 2003.

[21] Y. Chen, M.A. Nascimento, B.C. Ooi, and A.K.H. Tung, "Spade: On shape-based pattern detection in streaming time series," ICDE, pp.786–795, April 2007.

[22] M. Toyoda, Y. Sakurai, and T. Ichikawa, "Identifying similar subsequences in data streams," DEXA, Lect. Notes Comput. Sci., vol.5181, pp.210–224, Sept. 2008.

[23] 豊田真智子，櫻井保志，市川俊一，"データストリームのための部分シーケンスマッチング，" DEIM フォーラム 2009 論文集，E8-7, March 2009.

[24] M. Toyoda and Y. Sakurai, "Discovery of cross-similarity in data streams," ICDE, pp.101–104, March 2010.

[25] T.F. Smith and M.S. Waterman, "Identification of common molecular subsequences," J. Molecular Biology, vol.147, pp.195–197, 1981.

（平成 23 年 1 月 20 日受付）

豊田真智子　（学生員）

2004 お茶の水女子大・理・情報科学卒．2006 同大大学院人間文化研究科修士課程了．同年日本電信電話（株）入社．データストリーム処理の研究開発に従事．現在名古屋大学大学院情報科学研究科博士後期課程在学中．情報処理学会，日本データベース学会各会員．

櫻井　保志　（正員）

1991 同志社大・工・電気卒．同年日本電信電話（株）入社．1999 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了．博士（工学）．2004～2005 カーネギーメロン大学客員研究員．本会平 19 年度論文賞，情報処理学会平 16 年度及び平 19 年度論文賞，平 18 年度長尾真記念特別賞，日本データベース学会上林奨励賞，ACM KDD Best Paper Award（2008 年及び 2010 年）など受賞．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，情報処理学会，日本データベース学会各会員．

石川　佳治　（正員）

1989 筑波大・第三学群・情報学類卒．1994 同大大学院博士課程工学研究科単位取得退学．同年奈良先端科学技術大学院大学助手．1999 筑波大学電子・情報工学系講師．2004 同助教授．2006 名古屋大学情報基盤センター教授．博士（工学）（筑波大学）．データベース，データ工学，情報検索等に興味をもつ．日本データベース学会，情報処理学会，人工知能学会，ACM，IEEE 各会員．

NII-Electronic Library Service