

# 時系列データのための ストリームマイニング技術

櫻井保志

日本電信電話(株) NTT サイバースペース研究所  
sakurai.yasushi@lab.ntt.co.jp

ユビキタスコンピューティングという言葉があるように、今後センサの小型化や低価格化によって、大規模なセンサネットワークがさまざまなところで構築されるようになると思われる。その際に、大量のセンサから送られてくる時系列データ、すなわちデータストリームを高速に分析するストリームマイニング技術は非常に重要になってくる。ストリームマイニングの研究には、探索、トレンド検出、予測などさまざまな取り組みがあるが、本稿では、各々の取り組みから代表的な技術を紹介する。さらに、最新の取り組みとして、ストリームからの遅延相関検出技術を紹介する。この技術によって、たとえば類似したトレンドのストリームを出力するセンサ群をリアルタイムに検出したり、あるいは故障によって異常値を出力するセンサを即座に発見することが可能になる。

## データストリームとストリームマイニング

### ●●データストリームとは

データストリームとは、簡単に言うとネットワークから高速に流れてくる大量のデータのことである。例として、株価などの金融データ、商品販売のような流通データなどが挙げられる。また近年では、センサの小型化や低価格化によって、大規模なセンサネットワークがさまざまなところで構築されてきている。従来、センサは主として工場やプラント設備など特殊な場所で利用されてきたが、今後はビルに温度や照度のセンサを設置して空調や照明を自動的に制御したり、自動車に数多くのセンサを設置して遠隔地から自動車の各装置に故障がないかどうかを監視したりするなど、生活のさまざまな場面でセンサが使われるようになるだろう。そうすると、大量のセンサから送られてくる計測値の時系列、これもデータストリームである。

### ●●データストリーム処理

データストリーム処理に関する取り組みは大きく2つに分けることができる。

#### (A)データストリーム管理システム

1つはシステム化技術に関する取り組みである。これは、高いビットレートで配信されるデータストリームを高速に処理するための技術である。このような技術によって開発されたシステムはデータストリーム管理システム、もしくはDSMS (Data Stream Management

System)と呼ばれている。典型的な例では、既存のデータベース管理システムによるSQLに基づいた問合せ処理と同様に、データストリームからSQLなどを使って問合せを行うことができる。DSMSについては、本誌の4月号の記事<sup>1)</sup>にて解説されているので参照されたい。

#### (B)ストリームマイニング

もう一方は、時系列から有用な規則やパターンを見つけるための取り組みである。時系列データについては、時系列解析、機械学習、知識発見、データベースなどの分野で取り組まれてきたが、従来の取り組みの多くは、蓄積された有限長の時系列データの処理に注目してきた。近年、これまでの時系列処理とは異なり、データストリームを指向した時系列処理の研究が盛んになっており、ストリームマイニングと呼ばれている。これは、単にデータベースに蓄えられた大規模データを分析するのではなく、どんどん増え続けるデータの流れをリアルタイムに分析し、監視するための技術である。センサデータの分析処理、ネットワークの分析や監視、人や車のような移動物体の監視などへの応用が期待されている。増え続ける大規模なデータを分析するため、また利用者に情報をリアルタイムに提供するため、ストリームマイニングの技術は高速化と省メモリ化を図っている。

ここで、従来の時系列処理とストリームマイニングとの違いを説明する。たとえば、データベース分野における時系列検索の取り組みでは、図-1(a)のように大量のデータを時系列データベースとしてディスクに格納する。

検索効率を向上させるためにあらかじめインデックスを構築しておく場合もある。そして、問合せを受け付けたとき、インデックスを用いて高速に時系列データの検索を行う。これに対してストリームマイニングでは図-1 (b)のように、ストリーム処理エンジンがスケッチと呼ばれるデータストリームの要約情報を持っている。スケッチはデータを受信するごとに更新される。スケッチがあるため、オリジナルのデータを破棄することができ、そして時系列処理を省メモリかつ高速に実行することができる。

ストリームマイニングの研究には、さまざまな取り組みがあり、以下は現在行われている代表的な取り組みである。

- **統計情報の取得**：頻出値や部分シーケンスの和など、ストリームの統計情報を計算する技術
- **分類**：複数のストリームを受信しているときに、継続的に分類(クラスタリング)する技術
- **類似探索**：複数のストリームを受信しているときに、どのストリームとどのストリームが似ているのかを継続的に調べる技術
- **トレンド検出**：データストリームがどのような周期性や周波数を持っているのかを継続的に調べる技術
- **予測**：ストリームの将来の値を予測し続ける技術  
どの取り組みも、無限に伸びていくデータストリームを限られた時間とメモリ空間で解こうとするものである。このため、厳密に正解を求めるのではなく、近似解を求めている。これらの取り組みの中で、統計情報の取得と分類については、本誌の2005年1月号の記事<sup>2)</sup>に述べられているので参照されたい。本稿では、類似探索、トレンド検出、予測について解説する。さらに、最新の取り組みとして、ストリームからの遅延相関検出技術を紹介する。

## ●●データストリーム処理の応用例

ストリームマイニングについて、機能別の応用例をここで紹介したい。まず、類似探索とトレンド検出に関する応用例については次のようなものが挙げられる。

### ● **故障検出**

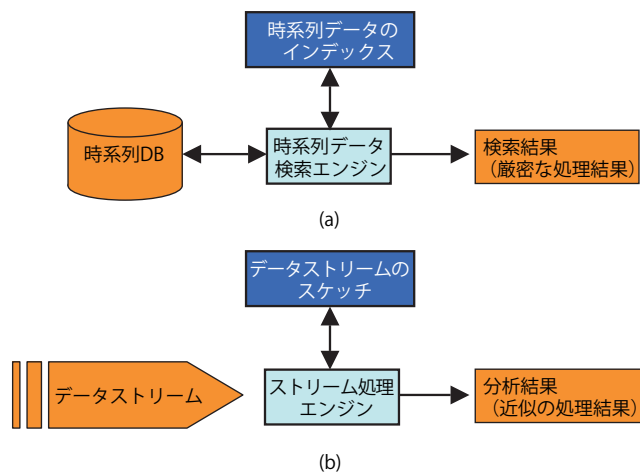
現在の時間帯は毎週、ある決まったパターンの時系列データをセンサ(もしくは設備の機器)から受信していたが、今受信した時系列データはまったく異なるパターンである。この場合、センサの故障が考えられる。

### ● **映像検出**

あらかじめ用意した映像シーンと類似したシーンがストリームで送られてきた場合、それを検出して録画する。

### ● **ネットワーク監視**

あるサーバに関して、SYN-ACKパケットの量が、ある特定のパターンで推移している場合、そのサーバがDoS攻撃を受けている可能性が考えられる。



● 図-1 従来の時系列データ検索とデータストリーム処理

次は、情報予測に関する応用例である。従来のように天気や株価の予測が応用例として考えられるが、ストリームマイニングの技術はそのような場合のみならず、以下のような高速性が要求される場面でも用いることができる。

### ● **負荷分散**

あるサーバの負荷の予測値が一定レベル以上を超える場合、一部の処理を他のサーバに振り分ける。実際の負荷が基準を超える前に対処することが可能となる。

### ● **設備監視**

設備内のセンサの予測値が基準を超える場合、管理者に通知する。

### ● **自動車のセンサによる部品監視**

自動車の各部品にセンサを取り付けておき、ネットワーク経由で監視する。部品の破損がセンサデータから予測される場合、運転手に通知する。

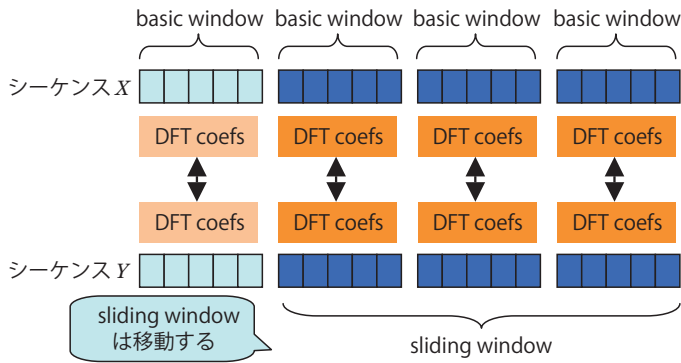
ストリームマイニングでは、実際にどのようにデータストリームを処理するのだろうか。以下では、類似探索、トレンド検出、予測、各々の取り組みから代表的な技術を紹介する。

## データストリームからの類似シーケンスの探索

### ●● **相関値とは**

時系列解析では、2つのシーケンスが似ているかどうかの尺度として、相関値が使われる。相関値とは、平均を0、分散を1に正規化した2つのシーケンスの内積であり、-1から+1までの値をとる。シーケンス  $X = (x_1, \dots, x_n)$  と  $Y = (y_1, \dots, y_n)$  が与えられているとき、相関値  $\rho$  は次のように求める。

$$\rho = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sigma_x \cdot \sigma_y} \quad (1)$$



●図-2 StatStreamによるシーケンスマッチング処理

ここで、 $\bar{x}$ と $\sigma_x$ はXの平均と標準偏差である。相関値は、2つのシーケンスが連動していれば+1に近い値をとり、逆に正反対の動きをしていると-1に近い値をとり、そして、まったく連動していないときに0の値をとり、無相関と呼ばれている。

## ●部分シーケンスの相関検出

部分シーケンスの相関を検出するための技術として、文献3)において提案されているStatStreamを紹介する。これは、複数のデータストリームのシーケンスが与えられたとき、相関のある部分シーケンスのペアを高速に検出するための手法である。この手法では、2種類のウィンドウサイズを用いている。

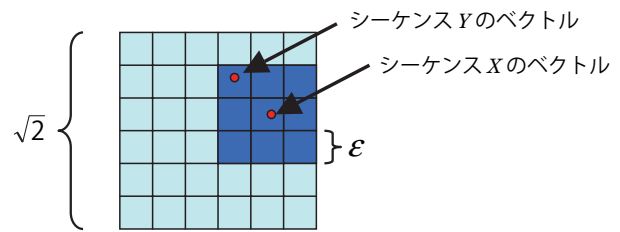
### ● Sliding window

sliding windowのモデルに基づいて、部分シーケンスの相関を調べる。このモデルでは、一定幅のウィンドウ内で2つの部分シーケンスのマッチングを行い、新たなデータを受信するごとにウィンドウを移動させていく。

### ● Basic window

オリジナルのシーケンスを要約するためにスケッチを計算する。シーケンスはbasic windowごとに区切られ、そのbasic windowごとにスケッチが計算される。オリジナルのシーケンスを用いて相関値を計算する代わりに、スケッチを用いて計算することによって、処理の高速化を図っている。

図-2は、2つのウィンドウサイズを用いた相関値計算の様子を示している。XとYの2つのシーケンスが与えられたとき、basic windowごとに部分シーケンスを正規化し、離散フーリエ変換(DFT)の係数値を計算する。StatStreamのスケッチは、このDFT係数である。そして、sliding window内のすべてのDFT係数のユークリッド距離を計算する。部分シーケンスの相関値 $\rho$ は、以下のようにユークリッド距離 $d$ から簡単に求めることができる。



●図-3 Grid structureによる類似部分シーケンスの高速探索

$$\rho = 1 - \frac{1}{2}d^2 \quad (2)$$

## ●データ構造による探索の高速化

StatStreamによる探索処理にはもう1つの特徴がある。それは、彼らがGrid structureと呼んでいるデータ構造を用いる点である。図-3はGrid structureの例である。この構造はシンプルなベクトルの配列であり、すべてのシーケンスのDFT係数をベクトルとして多次元空間にマッピングすることによって作ることができる。上述のように、距離が小さいシーケンスのペアは相関値が高いという関係がある。そこで、データ構造を用いることによって、多次元空間内で近接するベクトルのペアを効率的に見つける。このことによって、シーケンスのすべてのペアのチェックを回避し、相関のあるシーケンスのペアを高速に見つけることができる。

## ウェーブレット変換によるデータストリームからのトレンド検出

### ●ウェーブレット変換とは

ウェーブレット変換は、数値関数のトレンドを捉えることができる数学的な変換として、天文、気象、マルチメディアなどさまざまな分野で用いられている。変換はhigh pass filterとlow pass filterによって得ることができる。たとえば、最も簡単なHaar基底の場合では、高周波と低周波はそれぞれ以下のように、差と和の操作によって得られる。

$$\begin{aligned} W_{h,t} &= (V_{h-1,2t} - V_{h-1,2t-1})/\sqrt{2} \\ V_{h,t} &= (V_{h-1,2t} + V_{h-1,2t-1})/\sqrt{2} \end{aligned} \quad (3)$$

ここで、 $h$ はウェーブレットのレベル、 $t$ は時間を表す。ウェーブレット変換はシーケンスを階層的に表現したものであり、小さい係数値を無視することによって、効率



的なデータ圧縮が可能になる。また、シーケンスの長さ  $n$  に対して  $O(n)$  の時間で計算することができる。

ウェーブレット変換の係数は、シーケンスと基底の内積に等しい。すなわち、 $i$  番目のウェーブレット係数  $W_i$  は、 $X$  と  $i$  番目の基底  $B_i$  との内積と言える。

## ●●ウェーブレット変換のデータストリームへの適用

データストリームのトレンド検出とは、データストリームがどのような周期性や周波数を持っているのかを継続的に調べる技術である。トレンド検出の要素技術の1つとして、ここではデータストリームからウェーブレット変換を計算する Gilbert らの手法<sup>4)</sup>を紹介する。データストリームのウェーブレット変換を求めたい——これが彼らのモチベーションである。しかし、データストリームはネットワークから高速に流れてくるデータである。データを受信するごとに、ウェーブレット変換を最初から計算していたのでは、その計算はシーケンスが長くなるにつれて増大する。彼らはこの計算時間を大幅に低減化している。

ストリームマイニングでは、インクリメンタル (incremental) なアルゴリズムが好まれる。これは、データを受信するたびに毎回最初から計算をするのではなく、前回の結果を用いて計算するようなアルゴリズムを指す。たとえば、シーケンスの和や分散はインクリメンタルに求めることが可能である。なぜなら、データを受信するたびにシーケンスの和や分散を更新する場合、前回までの累積値に今回の値を加えるだけで済むためである。

Gilbert らは、ウェーブレット変換をインクリメンタルに計算するためのアルゴリズムを提案している。そして、オリジナルのシーケンスを保存することなく、省メモリでの計算が可能である。

## ●●ランダム射影

Gilbert らは、インクリメンタルなウェーブレット変換を実現するためにランダム射影を用いている。ランダム射影とは、ランダム関数によって基底を作り、多次元空間へのマッピングを行う手法である。たとえば、Achlioptas の手法<sup>5)</sup>では、ランダム関数によって作成された行列  $M$  によって、シーケンス  $X$  の射影を以下のように求めることができる。

$$P_X = \frac{1}{\sqrt{k}} XM$$
$$M(i,j) = m_{ij} = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases} \quad (4)$$

ここで、 $P_X$  は  $X$  の  $k$  次元空間内のベクトルである。この方法によって、低い計算コストでの次元圧縮が可能と

なる。計算量に関しては、データ受信ごとに  $O(1)$  の時間で  $P_X$  を更新することができる。また、その精度については確率的に保証されている。

## ●●インクリメンタルなウェーブレット変換

Gilbert らの手法では、ランダム射影をデータストリームのスケッチとして用い、そのスケッチによってウェーブレット係数の推定を行う。ウェーブレット係数  $W$  がシーケンス  $X$  と基底  $B$  の内積に等しいことは以前に述べた。内積はユークリッド距離から計算できるため、 $X$  と  $B$  のユークリッド距離から  $W$  を求めることが可能である。

ここで、 $X$  と  $B$  のランダム射影  $V_X$  と  $V_B$  を考える。ランダム射影はオリジナルのシーケンスの距離関係を保存する。よって、 $X$  と  $B$  の距離を、 $V_X$  と  $V_B$  を用いることによって近似することができる。そこで、Gilbert らは、 $V_X$  と  $V_B$  からウェーブレット係数  $W$  を求めることを提案している。 $V_X$  と  $V_B$  を用いるため、 $W$  をインクリメンタルに求めることが可能である。その計算量は、データ受信ごとに  $O(1)$  の時間、つまりシーケンス長に依存せず一定である。

## パターンの発見による データストリームの情報予測

### ●●自己回帰モデルとは

時系列解析の分野において、予測のためのアプローチとして自己回帰モデル (AutoRegressive model) が知られている。たとえば、最も簡単な AR(2) では、シーケンスの値  $x_t$  をその前の2つの値  $x_{t-1}$  と  $x_{t-2}$  から計算する。

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} \quad (5)$$

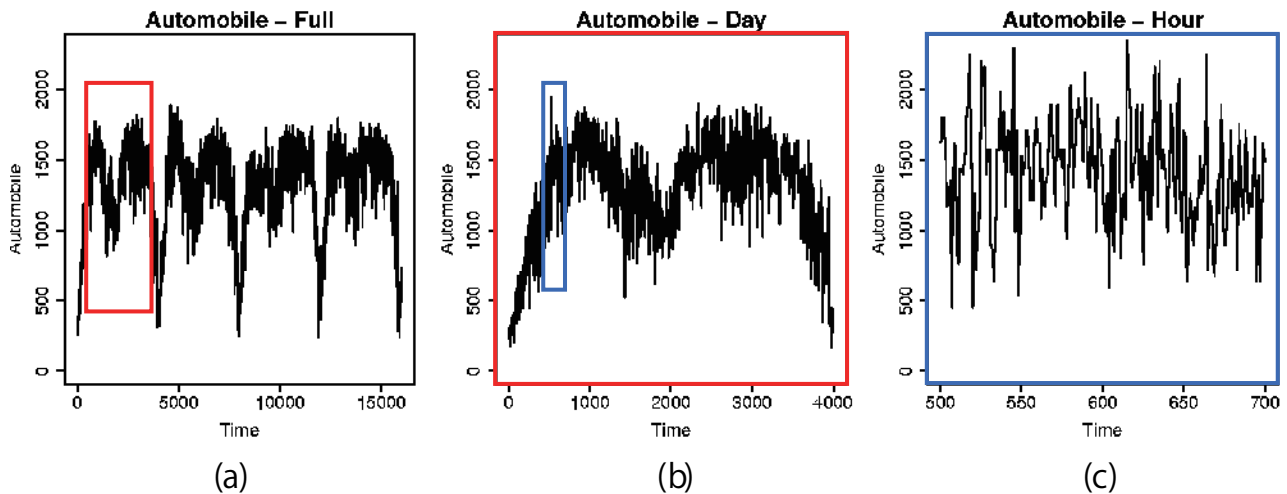
具体的には、長さ  $n$  のシーケンス  $X=(x_1, \dots, x_n)$  が与えられているとき、 $x_t$  を最小の誤差で表現する  $\alpha_1$  と  $\alpha_2$  を最小2乗法によって求める。 $\alpha_1$  と  $\alpha_2$  から将来の値  $x_{n+1}$  の予測は

$$x_{n+1} = \alpha_1 x_n + \alpha_2 x_{n-1} \quad (6)$$

のように求めることができる。文献(6)に示されているように最小2乗法もインクリメンタルに計算することができる。

### ●●パターンの発見と情報予測

データストリームの予測技術として、Papadimitriou らの手法<sup>7)</sup>を紹介する。彼らが AWSOM と呼ぶ提案手法は、主としてパターンの発見、特に階層的なパター



● 図-4 階層的なパターンの例

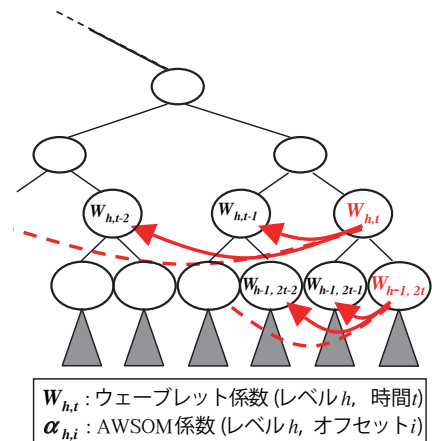
ンを高速に見つけることを目的としている。たとえば、**図-4**は自動車の交通量を示したものである。図-4(a)を見ると、交通量が1日周期で変動していることが分かる、その1日に着目すると図-4(b)に示すように半日の周期があることが分かる。これは朝夕のラッシュアワーである。図-4(c)は1時間未満の部分シーケンスに着目しており、これは周期性のないノイズと考えることができる。AWSOMは、このような階層的パターンが存在するシーケンスから周期性を発見し、予測するための手法である。AWSOMは高速性が要求されるような場面でも用いることができるため、さまざまな応用例が考えられる。たとえば、システムの負荷やネットワークトラフィックの監視である。1日や1週間の周期、月や年の周期でシステムの負荷やネットワークトラフィックが変化するような場合、膨大なオリジナルのシーケンスを保存することなく、異なる周期を考慮した予測が可能となる。

## ●●ウェーブレット変換を用いた情報予測

周期的なシーケンスにウェーブレット変換を行うと、その周期に対応するレベルにおいて、ウェーブレット係数は高いエネルギーを持つ。そこで、Papadimitriouらは階層的パターンを持つシーケンスを予測するために、ウェーブレット変換とARのような線形モデルを組み合わせたモデルを提案している。

$$\begin{aligned}
 W_{h,t} &= \alpha_{h,1} W_{h,t-1} + \alpha_{h,2} W_{h,t-2} + \dots \\
 W_{h-1,2t} &= \alpha_{h-1,1} W_{h-1,2t-1} + \alpha_{h-1,2} W_{h-1,2t-2} + \dots
 \end{aligned}
 \tag{7}$$

ここで、 $W_{h,t}$ はレベル $h$ 、時間 $t$ のウェーブレット係数を示している。式(7)においてレベル $h$ では、 $W_{h,t}$ を最小の誤差で表現する $\alpha_{h,1}$ と $\alpha_{h,2}$ を最小2乗法によって求める。**図-5**は、その様子を示したものである。



● 図-5 AWSOM

彼らの手法 AWSOM は、ウェーブレット変換を用いることにより、階層的パターンを検出できるだけでなく、以下のような特長がある。

### ● ウェーブレット係数の選択

エネルギーの低いウェーブレット係数を破棄することによって、ノイズを除去することができる。

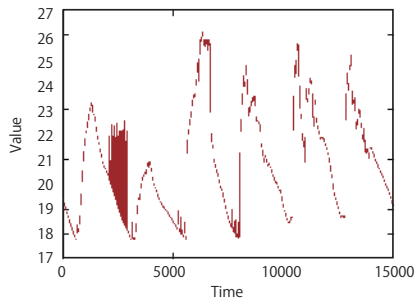
### ● インクリメンタルな処理

ウェーブレット係数は、Gilbertらの手法<sup>4)</sup>によってインクリメンタルに求めることができる。さらに線形モデルも文献6)に示されているようにインクリメンタルに計算することができる。よって、予測処理はインクリメンタルに実行することができる。インクリメンタルな処理によって、高いビットレートのデータストリームであっても、将来の値を予測し続けることが可能となる。

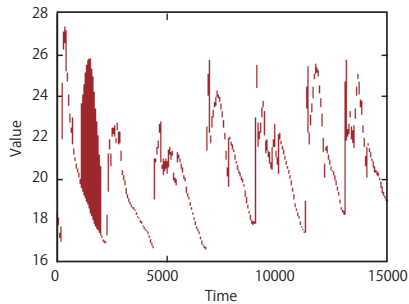
## データストリームにおける遅延相関の検出

### ●●遅延相関とは

筆者らは複数ストリーム間の遅延相関を検出する技術について取り組んでいる。これは、ストリームマイニングにおいて、類似探索の取り組みの1つである。相関の

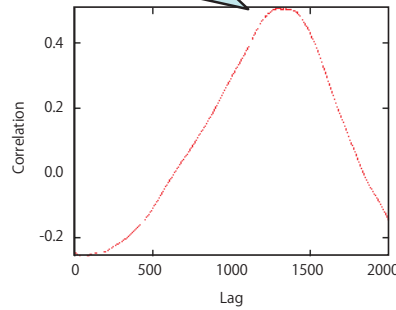


温度センサ X



温度センサ Y

温度センサ Xは、遅延1,300で  
温度センサ Yと相関している



相互相関関数

● 図-6 センサデータの遅延相関

ある時系列には大抵いくらかの遅延がある。

- 金利が下がると数カ月後に住宅着工数が増加する。
- 水道水にフッ素を入れると数年後に虫歯の患者数が減少する。

などは典型的な遅延相関の例である。「風が吹けば桶屋が儲かる」という言葉があるが、このようなケースをデータストリームから見つけようとする取り組みである。

筆者らは、複数のストリームがネットワーク上を流れているときに、どのストリームとどのストリームが相関しているのかを検出し、そしてストリームの各ペアにどれくらいの遅延があるのかを利用者に通知するような技術を開発した。たとえば、次のような応用が考えられる。

- 分散システムにおいて、あるサーバの CPU 利用率が上昇すると、それに連動してどのサーバに負荷がかかるのかが分かる。この知識によって、システムの負荷分散を動的に行うことが可能になる。
- 本来は連動しているセンサ群の中で、一部のセンサがまったく異なる動きをしている場合、センサの故障が考えられる。このような状況を即座に検出することが可能になる。

### ●●遅延相関と相互相関関数

遅延相関について、もう少し詳しく説明する。遅延相関は、相互相関関数から求めることができる。シーケンス X と Y の相互相関関数は以下の通りである。

$$R(l) = \frac{\sum_{t=l+1}^n (x_t - \bar{x})(y_{t-l} - \bar{y})}{\sqrt{\sum_{t=l+1}^n (x_t - \bar{x})^2} \sqrt{\sum_{t=1}^{n-l} (y_t - \bar{y})^2}} \quad (8)$$

$$\bar{x} = \frac{1}{n-l} \sum_{t=l+1}^n x_t, \quad \bar{y} = \frac{1}{n-l} \sum_{t=1}^{n-l} y_t$$

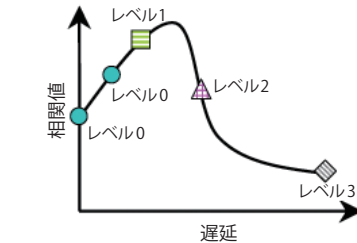
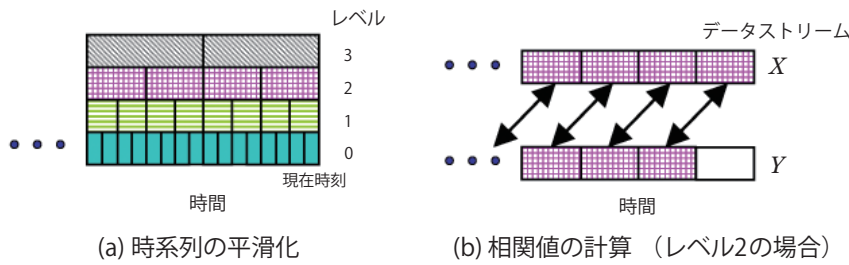
ここで、 $l$  は遅延であり、 $R(l)$  は X が  $l$  遅れたときの相関値である。 $l=0$  のとき、式 (8) と式 (1) は同じになり、 $R(0) = \rho$ 。相互相関関数は X をずらしたときに両者にどれほどの相関値があるのかを示したものである。そして、遅延  $l$  において高い相関値を持つとき、X と Y は  $l$  の遅延相関を持つという。

ここで例を用いて遅延相関の求め方を説明する。

図-6 左側の 2 つの時系列は温度センサデータであり、約 1 分ごとに計測した温度を示している。ここで、たとえばプラント設備の中で、2 つの装置に設置した温度センサが連動していることを監視している状況を想定する。2 つの温度センサはよく似た傾向を示しているが、若干のずれがある。具体的には、温度センサ X の方が少々遅れている。たとえば温度センサ X の最初のピークは、温度センサ Y の最初のピークより 1,000 くらい遅い時刻に位置している。そして、右側の図は相互相関関数である。縦軸は相関値、横軸は遅延の大きさである。図-6 では、2 つの温度センサに遅延がないものとして相関を調べると、あまり相関がないという結果になるが、温度センサ Y を少し遅らせて両者の相関を調べると高い相関値が得られる。そして、遅延が 1,300 のところで相関値のピークがある。したがって、この場合は「温度センサ X は 1,300 の遅延で温度センサ Y と相関している」といえる。この結果から、何の故障もなく温度センサが連動している様子が分かる。

### ●●相互相関関数を近似する

データストリームは刻々と流れてくる。時が経つにしたがって、そのデータストリームはどんどん長くなり、トレンドが変わることもある。ある時刻では相関がなかったストリームのペアも、時が経つと相関のあるペアになるかも



(c) 相互相関関数のサンプリングと補間

● 図-7 相互相関関数の近似

れない。そこで、筆者らは遅延相関を効率的にモニタリングするための技術である BRAID を開発した<sup>8)</sup>。

式 (8) に示した相互相関関数を求めるには、平均、分散、内積が必要である。遅延相関を求めるため、BRAID はデータストリームの統計情報 (平均、分散、内積) を保有している。平均、分散、内積はインクリメンタルに求めることができるため、BRAID はデータを受信するごとに、それらの統計情報を更新する。利用者もしくはアプリケーションからの要求があると、統計情報から遅延相関を計算し、相関のあるストリームのペアとその遅延の値を出力する。たとえば、「2 番目のストリームは 55 番目のストリームと相関があり、その遅延は 1,300」のように出力する。

遅延相関を計算するためには、データ受信ごとに  $O(n)$  の計算量とメモリ量が必要である。これは、システムを運用していくと、計算時間とメモリ量はどんどん時間に比例して増えていくことを意味している。そこで、高速化と省メモリ化を達成するために、BRAID では図-7 のように平滑化とサンプリングを用いて相互相関関数を近似する。

### ● 時系列の平滑化

オリジナルのデータストリームをレベル 0 の時系列として、レベル  $h$  の時系列の値を 2 個ずつとって平均値を計算し、レベル  $h+1$  の時系列を作成する。

### ● 相関値の計算

データストリーム  $Y$  を 1 つ遅らせて、データストリーム  $X$  との相関値を計算する。

### ● 相互相関関数のサンプリングと補間

レベルごとに相関値を 1 つずつ計算する。これは結果的に相互相関関数をサンプリングすることになる。相関値を補間して相互相関関数を近似する。

この近似によって遅延相関の計算時間を低減化させている。具体的には、データ受信ごとに  $O(\log n)$  のメモリ量と  $O(1)$  の平均時間で遅延相関を検出することができる。

もちろん、計算結果は近似解になるが、十分な精度があることが評価結果から分かっている。

## ユビキタス社会に向けて

将来のユビキタス社会ではネットワークにつながったさまざまなセンサが使われるだろう。そのようなセンサデータから意味のある情報を抽出し、提供するストリームマイニングは重要なコア技術になると考えられている。現在はまだまだ発展途上であるが、洗練された技術が確立されたときには、わたしたちの暮らしはより便利に、安全に、そして快適になるだろう。

本稿ではストリームマイニングの中でも、トレンド検出、類似探索、予測について、各々の取り組みから代表的な技術を紹介した。本稿で紹介した技術以外にもさまざまな研究成果がある。より詳しい内容に興味のある読者は、データベース (SIGMOD, VLDB, PODS)、データマイニング (KDD, PKDD)、計算理論 (STOC, FOCS) などの国際会議の予稿集を参照されたい。

### 参考文献

- 1) 白石 陽: センサネットワークのためのデータベース技術, 情報処理, Vol.47, No.4, pp.387-393 (Apr. 2006).
- 2) 有村博紀, 喜田拓也: データストリームのためのマイニング技術, 情報処理, Vol.46, No.1, pp.4-11 (Jan. 2005).
- 3) Zhu, Y. and Shasha, D.: StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time, In Proc. of VLDB, pp.358-369 (2002).
- 4) Gilbert, A. C., Kotidis, Y., Muthukrishnan, S. and Strauss, M.: Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries, In Proc. of VLDB, pp.79-88 (2001).
- 5) Achlioptas, D.: Database-Friendly Random Projections, In Proc. of PODS, pp.274-281 (2001).
- 6) Yi, B.-K., Sidiropoulos, N., Johnson, T., Jagadish, H. and Faloutsos, C.: Online Data Mining for Co-Evolving Time Sequences, In Proc. of ICDE, pp.13-22 (2000).
- 7) Papadimitriou, S., Brockwell, A. and Faloutsos, C.: Adaptive, Hands-Off Stream Mining, In Proc. of VLDB, pp.560-571 (2003).
- 8) Sakurai, Y., Papadimitriou, S. and Faloutsos, C.: BRAID: Stream Mining through Group Lag Correlations, In Proc. of ACM SIGMOD Conference, pp.599-610 (2005). (平成 18 年 5 月 25 日受付)