DEIM Forum 2010 F10-1

データストリームに対する ハイブリッド型連続的外れ値検出手法の提案

石田　梢† 北川　博之†,††

† 筑波大学大学院システム情報工学研究科　〒 305–8573 茨城県つくば市天王台 1–1–1
†† 筑波大学大学院計算科学研究センター　〒 305–8573 茨城県つくば市天王台 1–1–1
E-mail: †kozue-i@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

あらまし　外れ値検出は，データ集合中で他の大多数のデータから大きく異なる値や特徴を持つデータを抽出する技術で，ネットワークへの不正侵入の検出や医療情報の異常値の検出など，様々な分野で応用されている．一方で，センサデバイスやネットワーク技術の進展により，データストリームとして配信されるデータが増加している．データストリームを用いた状況監視アプリケーションにおける迅速な異常の検出の場面等では，データストリームに対して外れ値検出を行うことが重要となる．データストリームに対して外れ値検出をするには，逐次到着する最新のデータに対して連続的に迅速な外れ値検出を行う必要がある．我々の先行研究では，データストリームとして表現される状態変化が大きくない場合に，効率的に外れ値検出を行う差分計算法を提案した．本研究では，データストリームとして表現される状態変化の大きさに応じて，差分計算法とナイーブな手法を切り替える効率的な手法を提案し，実験により有用性を評価する．

キーワード　外れ値検出，データストリーム，データマイニング，DB 外れ値

A Hybrid Method for Continuous Outlier Detection over Data Streams

Kozue ISHIDA† and Hiroyuki KITAGAWA†,††

† Graduate School of Systems and Information Engineering, University of Tsukuba　Tennodai1–1–1, Tsukuba-shi, Ibaraki, 305–8573 Japan
†† Center for Conputational Sciences, University of Tsukuba　Tennodai1–1–1, Tsukuba-shi, Ibaraki, 305–8573 Japan
E-mail: †kozue-i@kde.cs.tsukuba.ac.jp, ††kitagawa@cs.tsukuba.ac.jp

Abstract　Outlier detection is an important data mining technique, and it discovers outliers which have features that differ profoundly from other objects or values. The development of sensor devices and network techniques have increased data streams. This paper presents outlier detection over data streams by continuous monitoring of object statuses. In our previous works, we presented an effective outlier detection method with incremental processing, which is effective when object statuses do not change a lot over time. This paper introduces an improved method for outlier detection which dynamically switches between incremental processing. We evaluate effecriveness of the proposal by experiments.

Key words　outlier detection, data streams, data mining, DB-Oultier

1. ま え が き

データ集合中で他のデータから大きく異なる特徴や値を持つデータを検出する外れ値検出の技術は，ネットワークへの不正侵入の検出や医療データからの異常値の検出など様々な分野で応用され重要となっている．これまでにも，統計に基づく外れ値検出 [1], [2]，距離に基づく外れ値検出 [5]，クラスタリングによる外れ値の検出 [6], [7] 等，様々な手法が静的データ集合に対する外れ値検出手法として，提案されている．

一方で，近年のセンサデバイスやネットワーク技術の進展により，データストリームが増加している．データストリームに対する要求として，逐次配信されるデータをモニタリングし，異常を迅速に検出することがある．そのため，他とは大きく異なる値を取るデータストリームを検出するため外れ値検出が重

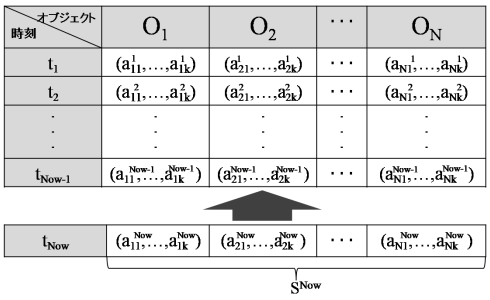

図 1　S^{Now} が到着する様子

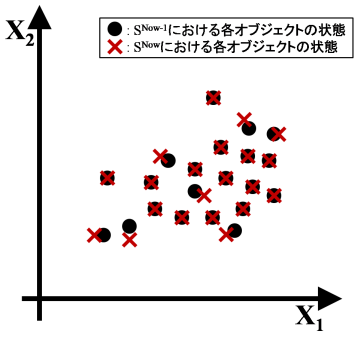

図 2　データ分布の変化

要となる．

例えば，同じ地域に置かれたセンサ群の測定値を監視して他のセンサとは大きく異なる値を示すセンサを検出，各社の株価変化を表すストリームを監視して他社と非常に異なる変化を示している会社を検出，移動するオブジェクト群の位置を監視して他のオブジェクトとはかけ離れた位置にあるオブジェクトを検出，等が考えられる．

一般に状態（センサ測定値，株価，位置等）が時間的に変化する N 個のオブジェクト $O_1, ..., O_N$ があった時，ある時刻において，もしオブジェクト O_i の状態が他のオブジェクトの状態と大きく異なる場合には，O_i はその時点において外れ値オブジェクトであるとみなすことができる．時間的に変化するオブジェクト群の連続的モニタリングにおいては，このような外れ値オブジェクトを連続的に検出するための技術が求められる．

本研究では，状態が時間的に変化する N 個のオブジェクト $O_1, ..., O_N$ の現在の時刻 t_{Now} における状態の集合に対して，その状態の集合の中で，他のオブジェクトの状態と大きく異なるオブジェクトを連続的に検出するための手法について考える．

より形式的に表すと次のようになる．状態が時間的に変化する N 個のオブジェクト $O_1, ..., O_N$ があり，時刻 $t_j (1 \leqq j \leqq Now)$ における各オブジェクトの状態が集合 $S^j = \{s_1^j, ..., s_N^j\}$ で表されるものとする．ただし，s_i^j は状態タプル $s_i^j = (a_{i1}^j, ..., a_{ik}^j)$ で，時刻 t_j におけるオブジェクト O_i の状態を表すとする．時刻 t_j におけるオブジェクト O_i の状態 s_i^j が時刻 t_j における他のオブジェクトの状態と大きく異なる場合に，O_i は時刻 t_j における外れ値オブジェクトであるとみなすことができる．

本稿では図 1 で表されるように，オブジェクトの状態集合 S^j がデータストリーム (S^1,..., S^{Now})（ただし，S^{Now} は現在の $O_1, ..., O_N$ の状態を表す状態集合とする）として連続的に与えられる際に，時刻 t_{Now} における状態集合 S^{Now} の外れ値オブジェクトを連続的に検出するための手法について考える．

我々の先行研究では，現在の状態集合は直前の時刻の状態集合が変化したものであり，図 2 で表すように，その変化があまり大きくないことも多いことを利用した差分計算法を提案している [10]．差分計算法では，直前の時刻の状態集合における外れ値検出結果を利用して差分計算を行い，現在の状態集合における外れ値オブジェクトを検出する．この手法は，直前時刻の状態集合からの変化が大きくない場合においては，毎回既存の静的なデータに対する外れ値検出手法を適用するナイーブな手法よりも効率的に処理を行うことが可能である．しかし，直前時刻の状態集合からの変化が大きい場合には効率的ではない．

本研究では，データストリームにおいて，直前時刻の状態集合からの変化の大きさに応じて，我々の先行研究で提案した差分計算法とナイーブな手法のうち，より処理速度が早い手法を選択し適用する手法を提案する．なお，本研究では，最も基本的な外れ値検出手法である距離に基づく外れ値検出手法 (Distance-Based Outlier，DB 外れ値) [3]〜[5] を基に外れ値を検出する．

本稿の構成は次の通りである．2. で関連研究について述べる．3. で準備として DB 外れ値の定義と DB 外れ値検出アルゴリズムについて述べる．4. で提案手法を示し，5. で提案手法の有用性を示した実験とその結果を示す．最後に 6. で本稿をまとめる．

2. 関 連 研 究

本節では既存の外れ値検出の研究について述べる．

静的データ集合に対する外れ値検出手法として様々な手法がこれまでに提案されている．代表的な手法としては，本研究で用いる DB 外れ値検出手法 [3]〜[5] の他に，統計に基づく手法 [1], [2]，クラスタリングによる手法 [6], [7]，密度を基にした手法 [8] 等がある．本研究でベースとする DB 外れ値検出は，他の手法に比べて，より単純で一般性のある手法である．[3]〜[5] ではその有用性や重要性が示されている．

データストリームに対する外れ値検出手法としては，次のような手法が提案されている．Yamanishi らの提案する手法 [9] では，データストリームに対して，データの確率モデルをオンライン忘却型学習アルゴリズムを用いて学習し，その確率モデルから逸脱したデータを外れ値として検出する．Angiulli ら [11] は，単位時間に 1 つの値が到着するデータストリームにおいて，過去の一定時間内に到着した値の集合中で逸脱した値を DB 外れ値検出手法を用いて検出する手法を提案している．データストリームに対して外れ値検出を行う点では上記の研究は本研究と共通である．しかし，上記の研究がいずれも過去に到着した値と比較して最新の値もしくは過去の一定時間内に到着した値が外れ値となっているかを検出するのに対し，本研究では最新の状態集合における外れ値を検出する点で対象とする問題が異なる．

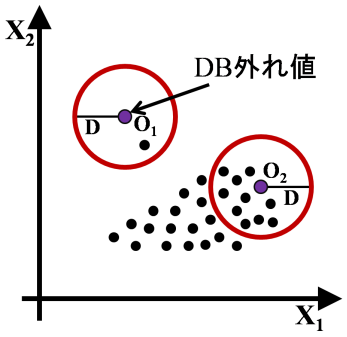

図 3　DB 外れ値の例: $p = 0.9$, $N = 30$

3. 準　　備

本節では，本研究で用いる外れ値の定義と，既存の静的データに対する DB 外れ値検出アルゴリズムを説明する．

3.1　DB 外れ値

本研究では，Knorr らにより定義された DB 外れ値を外れ値の定義として用いる [5].

[定　義　1]

N 個の k 次元オブジェクト $O_1, ..., O_N$ からなるオブジェクト集合 S において，オブジェクト O_i が **DB**(p, D) **外れ値**であるとは，O_i からの距離が D より大きい範囲に，S 中の $\lceil pN \rceil$ 個以上のオブジェクトが存在するということである．□

以降では，オブジェクト O_i から距離が D 以下の範囲を O_i の **D 近傍**と呼ぶ．$M = \lfloor N(1-p) \rfloor$ を用いると，O_i が DB(p, D) 外れ値であるとは，オブジェクト O_i の D 近傍内のオブジェクト数 (O_i を含む) が M 個以下ということと等価である．

$k = 2$ の場合の DB 外れ値の例を図 3 に示す．各点はオブジェクトを表す．$p = 0.9$, $N = 30$ とする．従って，もしあるオブジェクトの D 近傍内のオブジェクト数が $M = 3[M = 30 * (1-0.9)]$ 以下であれば，そのオブジェクトは DB 外れ値である．図 3 の左上の円，右下の円はそれぞれオブジェクト O_1, O_2 の D 近傍を表す．この時，O_1 の D 近傍内のオブジェクト数は $2(\leqq M)$ であるので，O_1 は DB 外れ値である．一方，O_2 の D 近傍内のオブジェクト数は $11(> M)$ であるので O_2 は DB 外れ値ではない．

3.2　セルを用いたアルゴリズム

N 個の k 次元オブジェクト $O_1, ..., O_N$ からなるオブジェクト集合 S 中の DB 外れ値を検出する最も単純な方法は，各オブジェクト $O_i (1 \leqq i \leqq N)$ に対して O_i と $O_p (1 \leqq p \leqq N, i \neq p)$ の距離 $d(O_i, O_p)$ を順次計算し，O_i の D 近傍のオブジェクト数を数えて判断する方法である．しかし，このような単純な方法で外れ値検出を行うと，$O(N^2)$ 回の距離計算が必要となり，全体の計算量は膨大になる．

距離計算の回数を減らし，全体の計算量を削減するために，セルを用いたアルゴリズム (Cell-Based Algorithm) が Knorr らにより提案された [5]．セルを用いたアルゴリズムでは，距離計算の回数を減らすために，オブジェクトが存在する空間をセルで分割して計算を行う．

3.2.1　セ ル 構 造

N 個の k 次元オブジェクト $O_1, ..., O_N$ は，X_1，...,X_k 軸からなる k 次元空間中の点として表現できる．この k 次元空間中を一辺の長さが $l = \frac{D}{2\sqrt{k}}$(対角線の長さが $\frac{D}{2}$) のセルに分割する．X_1 軸について x_1 番目，...，X_k 軸について x_k 番目のセルを $C_{x_1,...,x_k}$ と表す．なお以降は，特にセルの各軸に対する添字を明記する必要がない場合は，セルを単に C_x と略記する．

この時，次の性質 1 が成り立つ．

[性　質　1]

同一セル中の 2 オブジェクト間の距離は $\frac{D}{2}$ 以下である．

セル C_x に含まれる 2 つのオブジェクト O_i，O_p 間の距離が最も長くなるのは，O_i，O_p 間の距離 $d(O_i, O_p)$ がセルの対角線の長さ $\frac{D}{2}$ と等しくなる場合である．従って性質 1 が成り立つ．

次に，$C_{x_1,...,x_k}$ の L_1 近傍範囲を定める．

[定　義　2]

$C_{x_1,...,x_k}$ の L_1 近傍 $L_1(C_{x_1,...,x_k})$ を，次のように定義する．$L_1(C_{x_1,..,x_k}) = \{C_{u_1,...,u_k} \mid |u_i - x_i| \leqq 1 (1 \leqq i \leqq k) \wedge C_{u_1,...,u_k} \neq C_{x_1,...,x_k}\}$　□

定義 2 から，次の性質 2 が成り立つ．

[性　質　2]

$C_{u_1,...,u_k} \in L_1(C_{x_1,...,x_k})$，$O_i \in C_{x_1,...,x_k}$，$O_p \in C_{u_1,...,u_k}$ の時 $d(O_i, O_p) \leqq D$.

セル C_x に含まれるオブジェクト O_i，C_x の L_1 近傍のセル C_u に含まれるオブジェクト O_p 間の距離が最も長くなるのは，O_i，O_p 間の距離 $d(O_i, O_p)$ がセルの対角線の長さの 2 倍と等しくなる場合である．従って性質 2 は成り立つ．

[定　義　3]

$C_{x_1,...,x_k}$ の L_2 近傍 $L_2(C_{x_1,...,x_k})$ を，次のように定義する．$L_2(C_{x_1,...,x_k}) = \{C_{u_1,...,u_k} \mid |u_i - x_i| \leqq \lceil 2\sqrt{k} \rceil (1 \leqq i \leqq k) \wedge C_{u_1,...,u_k} \notin L_1(C_{x_1,...,x_k}) \wedge C_{u_1,...,u_k} \neq C_{x_1,...,x_k}\}$ □

定義 3 から，次の性質 3 が成り立つ．

[性　質　3]

$C_{u_1,...,u_k} \notin L_1(C_{x_1,...,x_k})$，$C_{u_1,...,u_k} \notin L_2(C_{x_1,...,x_k})$，$C_{u_1,...,u_k} \neq C_{x_1,...,x_k}$，$O_i \in C_{x_1,...,x_k}$，$O_p \in C_{u_1,...,u_k}$ の時，$d(O_i, O_p) > D$.

C_x 内オブジェクト O_i と，C_x，C_x の L_1 近傍のセル，L_2 近傍のセルに含まれないオブジェクト O_p の距離が最小になる時は，定義 3 から必ず $d(O_i, O_p) > \lceil 2\sqrt{k} \rceil l$ が成り立つ．従って，$\lceil 2\sqrt{k} \rceil l \geqq 2\sqrt{k} l = 2\sqrt{k} \frac{D}{2\sqrt{k}} = D$ であることにより，性質 3 は成り立つ．

$C_{x_1,...,x_k}$ 内のオブジェクト数を n_0，$L_1(C_{x_1,...,x_k})$ 内のオブジェクト数を n_1，$L_2(C_{x_1,...,x_k})$ 内のオブジェクト数を n_2 とする．性質 1〜3 から次の性質 4 が成り立つ．

[性　質　4]

（1）　$n_0 > M$ ならば，$C_{x_1,...,x_k}$ 内の全オブジェクトは DB 外れ値ではない．

（2）　$n_0 + n_1 > M$ ならば，$C_{x_1,...,x_k}$ 内の全オブジェクトは DB 外れ値ではない．

（3）　$n_0 + n_1 + n_2 \leqq M$ ならば，$C_{x_1,...,x_k}$ 内の全オブジェクトは DB 外れ値である．

ここで，(1) の場合は *red*，(2) の場合は *pink*，(3) の場合は *yellow* とセル $C_{x_1,...,x_k}$ を色付けする．各セルに対してこのような方法で色付けする処理を行った後に，未だ色付けされていないオブジェクトが存在するセルがある場合，そのセルを *white* に色付けする．以下では，このセルに色付けする処理を **CCD** (Cell Color Decision) と呼ぶ．セルを用いたアルゴリズムでは，このように性質 4 に従いセルを分類し，セル毎に 3.2.2 で示すアルゴリズムに従い，DB 外れ値を検出する．

3.2.2　アルゴリズム

アルゴリズム 1 はセルを用いたアルゴリズムの処理の流れである．セル C_x の n_0，n_1，n_2 の値をそれぞれ $n_0(x)$，$n_1(x)$，$n_2(x)$ と表す．3-5 行目では $n_0 > M$ であるセルを *red* とする (性質 4(1))．6-8 行目では L_1 近傍に *red* セルが存在するセルを *pink* とする．これは，任意のセル C_x の L_1 近傍に *red* セルが存在する場合，C_x は必ず性質 4(2) を満たすからである．10-16 行目で性質 4(2) を満たすセルを *pink*，性質 4(3) を満たすセルを *yellow* とし，*yellow* とされたセル内の全オブジェクトを外れ値とする (性質 4(3))．最後に 17-24 行目で，16 行目までの処理で *red*，*pink*，もしくは *yellow* と判定されなかったセル C_w を *white* とし，*white* と判定されたセルに対して 19-24 行目にてセルに含まれる各オブジェクト O_i に DB 外れ値判定を行う．

本稿では特に *white* セルに対する処理 (19-24 行目) を **WCP** (White Cell Process)，*white* セル内の個々のオブジェクトに対する処理 (18-24 行目) を **WOP** (White Object Process) と呼ぶ．また，セルに対して CCD を行う処理 (3-24 行目) を **CCD 処理**と呼ぶ．

アルゴリズム 1: セルを用いたアルゴリズム

```
1.  for each オブジェクト O_i do
2.      O_i を含むセル C_x を特定し，n_0(x) を 1 増やす．
3.  for each セル C_x do
4.      if n_0(x) > M then
5.          C_x を red とする．
6.  for each red セル C_r do
7.          for each C_r の L_1 近傍のセルで red でないセル C_u
8.              C_u を pink とする．
9.  for each セル内にオブジェクトが存在し，色がついていないセル C_w do
10.     n_1(w) ← Σ_{C_i ∈ L_1(C_w)} n_0(i)
11.     if n_0(w) + n_1(w) > M then
12.         C_w を pink にする．
13.     else
14.         n_2(w) ← Σ_{C_i ∈ L_2(C_w)} n_0(i)
15.         if n_0(w) + n_1(w) + n_2(w) ≦ M then
16.             C_w を yellow とし，C_w 内の全オブジェクトを外
                れ値とする．
17.         else
18.             C_w を white とする．
19.             for each オブジェクト O_i ∈ C_w do
20.                 Count ← n_0(w) + n_1(w)
21.                 for each オブジェクト O_p ∈ C_u, C_u ∈ L_2(C_w)
                    do
22.                     if dist(O_i, O_p) ≦ D then
23.                         Count を 1 増やし，Count > M に
                            なったら O_i は外れ値ではないと判断し，
                            次のオブジェクトの処理 (20 行目) を開始．
24.                 O_i を外れ値とする．
```

4.　提 案 手 法

本節では，提案手法で検出するデータストリームにおける DB 外れ値の定義を示した後に，提案手法のアプローチを述べ，提案アルゴリズムを説明する．

4.1　CDB 外れ値

1 節で述べたように，本研究では，状態が変化する N 個のオブジェクト $O_1, ..., O_N$ が存在し，時刻 t_1 から t_{Now} までの状態集合 $S^j (= \{s_1^j, ..., s_N^j\}, 1 \leqq j \leqq Now)$ が与えられるものとする．ただし，$s_i^j = (a_{i1}^j, ..., a_{ik}^j)$ はオブジェクト O_i の時刻 t_j での状態タプルとする．ここで，時刻 t_j における DB 外れ値，および，CDB 外れ値を以下のように定義する．

[定　義　4]

オブジェクト O_i の状態タプル s_i^j が状態集合 S^j に対して DB 外れ値である時，s_i^j は時刻 t_j における DB 外れ値であると言う．時刻 t_{Now} における DB 外れ値を **CDB 外れ値** (Current-DB 外れ値) と言う．□

以降では，O_i の現在時刻 t_{Now} における状態タプル s_i^{Now} が CDB 外れ値である時，O_i を **CDB 外れ値オブジェクト**であると言う．また，オブジェクト O_i が時刻 t_j において DB 外れ値オブジェクトであるか否かを，O_i の t_j における外れ値の判定と言う．

4.2　提案手法のアプローチ

本研究では，データストリームに対して，効率的に DB 外れ値検出を行う手法を提案する．図 1 で示すように，現在時刻 t_{Now} における状態集合 S^{Now} が到着した時点で t_{Now} における DB 外れ値，すなわち CDB 外れ値を検出する．

我々はこれまでに，差分計算法を提案した [10]．差分計算法は，データストリームとして表現されるオブジェクトの状態の直前時刻からの変化が大きくない場合に効率的に処理を行うことができる．一方で，直前時刻からの状態変化が大きい場合は，セルを用いたアルゴリズムを毎回適用するナイーブな手法のほうが差分計算法に比べて効率的に処理を行うことができる．なお，本稿ではセルを用いたアルゴリズムを毎回適用するナイーブな手法を**再計算法**と言う．

提案手法では，S^{Now} が到着するたびに，S^{Now-1} からのオブジェクト状態変化の大きさに応じて，差分計算法と再計算法

のうち，より処理コストが小さい処理方法を選択して処理を行う．

本稿では，提案アルゴリズムにおいて処理を選択した後の，差分手法に基づいて処理を行う部分を**差分計算処理**, 再計算法に基づいて処理を行う部分を**再計算処理**と言う．

4.2.1 前　　提

以降では，直前時刻 t_{Now-1} から時刻 t_{Now} の間に状態が変化したオブジェクトを，t_{Now} における **SC オブジェクト** (State-Change Object) と言う．

提案手法では，次の 3 点を前提とする．

- 入力は，時刻 t_{Now} における SC オブジェクトの状態集合 $\Delta^{Now}(=\{s_q^{Now} \mid 1 \leqq q \leqq N \wedge s_q^{Now} \neq s_q^{Now-1}\})$ とする．
- 出力は，CDB 外れ値オブジェクトの集合とする．
- 連続的 CDB 外れ値検出アルゴリズムが対象とする状態集合は最初の状態集合 S^1 を除く 2 回目以降の状態集合 $S^{Now}(2 \leqq Now)$ とし，S^1 に対する外れ値オブジェクト検出は 3.2 節で述べたセルを用いたアルゴリズムを用いる．

4.2.2 基本的なアイデア

提案手法では，差分計算処理と再計算処理を切り替えて処理を行うために，次の二点を考慮する．

a） **データ構造の統一**

差分計算法と再計算法では，各セルが保持するデータ構造が異なる．そのため，2 つの処理方法をデータの欠落なく，かつ，効率的に切り替えるためには，再計算処理と差分計算処理はデータ構造を統一する必要がある．

提案手法では，再計算処理におけるセルが持つデータ構造を，差分計算法で用いているセルが持つデータ構造と同じとする．

b） **2 つの計算法の切り替えのための指標**

再計算法と差分計算法は，セルをベースとして処理を行うため，処理対象となるセルの数が処理時間に影響を与える．そのため提案手法では，再計算処理と差分計算処理を選択する時，定義 5 に示す指標でオブジェクトの状態集合の直前時刻からの変化の大きさを評価する．

[定　義　5]

オブジェクトの状態集合の直前時刻からの変化の大きさを測る指標として **CDS 変化率** (Cell-based Data Stream 変化率) を次のように定義する．

$$\text{CDS 変化率} = \frac{\text{SC オブジェクトの状態を含むセルの数}}{\text{全オブジェクトの状態を含むセルの数}}$$

提案手法では，閾値 θ を設定し，CDS 変化率が θ よりも大きい場合は再計算処理を行い, θ 以下である場合は差分計算処理を行う．

4.2.3 差分計算処理

差分計算法のベースはこれまでに我々の先行研究で提案している [10]．差分計算処理は，以降で説明する差分計算法に従い処理を行う．

データストリームにおいて，時刻 t_{Now} の状態集合 S^{Now} は直前時刻 t_{Now-1} の状態集合 S^{Now-1} の変化が小さい場合がある．例えば，直前時刻の状態と現在の状態が同じ，もしくは，直前時刻の状態と現在時刻での状態の差が少ないオブジェクト多く存在する等である．このような場合に，再計算法で処理を行うのでは無駄な処理が多く，適していない．

差分計算法は，セルを用いたアルゴリズムのアイデアをベースとし，外れ値の判定が変化する可能性のあるオブジェクトを含むセルに対してのみ処理を行う．差分計算法では，t_{Now} における SC オブジェクトの状態集合 Δ^{Now} を用いて差分計算を行う．

差分計算法では，次に示す差分計算のための [アイデア 1], [アイデア 2] に基づいて処理を行う．

[アイデア 1] 処理を行うセルの限定

t_{Now} における SC オブジェクトの状態変化により，そのセル内に存在するオブジェクトの外れ値の判定が変わる可能性のあるセルを**再判定セル**と言う．

アイデア 1 では，再判定セルに対象を限定して処理を行う．

a) SC オブジェクトが 1 つ存在する場合の再判定セル

t_{Now} における SC オブジェクト O_q が 1 つのみ存在する場合の再判定セルについて説明する．

セルで分割された状態空間において，O_q の状態変化は，次の（1）, (2）のいずれかに分類できる．

(1)：s_q^{Now} が含まれるセル C_x と s_q^{Now-1} が含まれるセル $C_{x'}$ が異なる．$s_q^{Now} \in C_x$, $s_q^{Now-1} \in C_{x'}$, $C_x \neq C_{x'}$.

(2)：s_q^{Now} が含まれるセルと s_q^{Now-1} が含まれるセルが同一．$s_q^{Now} \in C_x$, $s_q^{Now-1} \in C_x$.

（1）が成り立つ時，C_x と $C_{x'}$ の n_0，すなわち $n_0(x)$ と $n_0(x')$ が変化する．この時，C_x と C_x の L_1 近傍と L_2 近傍のオブジェクトを含むセルは再判定セルである．

（2）が成り立つ場合は C_x の $n_0(x)$ が変化しない．ただし，（2）が成り立つ場合においては，次のいずれかの条件が成り立つ時，再判定セルが存在する．

(2a)：s_q^{Now} を含むセル C_x の L_2 近傍に $white$ セル C_w が存在する場合

(2b)：s_q^{Now} を含むセル C_x が $white$ である場合

（2）が成り立ち，かつ，C_x の L_2 近傍に $white$ セル C_w が存在した場合，C_w に含まれるオブジェクトの状態と O_q の状態の距離は変化する．3.2 節で述べたように，$white$ セルに含まれるオブジェクトは，L_2 近傍のオブジェクトとの距離を計算して外れ値の判定を行うため，C_w は再判定セルである．

（2）が成り立ち，かつ，C_x が $white$ である場合，O_q の D 近傍の範囲が変化するため，O_q の外れ値の判定が変わる可能性がある．従ってこの時，C_x は再判定セルである．

b) SC オブジェクトが複数存在する場合の再判定セルの分類

一般には，t_{Now} における SC オブジェクトは複数存在する可能性がある．さらに，a) で述べた再判定セルではそれぞれ行う処理の内容が異なる．そのため，a) で述べた分類に基づき，SC オブジェクトが複数存在する場合の再判定セルを次のように定義する．

[定　義　6]

再判定セルとして，次の 4 種類のセルを定義する．

TypeA セル:（1）に該当する SC オブジェクト O_q の状態を t_{Now} または t_{Now-1} において含むセル．

TypeB セル: TypeA セルの L_1，L_2 近傍内に含まれるセル．ただし，自身が TypeA セルである場合を除く．

TypeC セル: その L_2 近傍内に（2）に該当する SC オブジェクト O_q の状態を t_{Now} および t_{Now-1} において含み，t_{Now} において $white$ であるセル．ただし，自身が TypeA，B セルである場合を除く．

TypeD セル: （2）に該当する SC オブジェクト O_q の状態を t_{Now} および t_{Now-1} において含み，t_{Now} において $white$ であるセル．ただし，自身が TypeA，B，C セルである場合を除く．

CDB 外れ値オブジェクトの検出において，t_{Now-1} における判定との差異の検出の対象を，以上の 4 種類の再判定セルに限定するのがアイデア 1 である．

[アイデア 2] n_1，n_2 値の保持

セルを用いたアルゴリズムにおけるセルの色付けをする際必要な値として，L_1 近傍，L_2 近傍のセルに含まれるオブジェクト数である n_1，n_2 値がある．セル C_x の n_1，n_2 値を計算する単純な方法は，C_x の L_1 近傍，L_2 近傍のセルの n_0 の値を順次読み出し，それらを合計するものである．

アイデア 2 では，t_{Now} における C_x の n_1，n_2 値を毎回このような方法で計算するのではなく，直前時刻 t_{Now-1} でのセル C_x の n_1，n_2 値を保持し，最新の状態集合 S^{Now} が到着した時に，n_1，n_2 値が変化するセルに対してのみ，n_1，n_2 の値更新して再計算する．

以降では，差分計算法において，n_1，n_2 値を保持している場合があることを前提に，CCD を行う処理を **CCD2 処理**と呼ぶ．CCD2 処理では，セル C_x に対して CCD を行う場合は，保持している n_1，n_2 値を用いて判定を行う．

4.2.4 再計算処理

再判定法は，最新の状態集合 S^{Now} が到着するたびに S^{Now} に対してセルを用いたアルゴリズム (3.2 節) を適用する方法である．再判定処理は，再判定法に基づき処理を行う．ただし，4.2.3[アイデア 2] で述べた通り，差分計算処理では，各セルはそのセルの n_1，n_2 値を保持する．そのため，再計算処理においても n_1，n_2 値を計算した場合は，その値を保持する．このようにして，再計算処理は差分計算処理と同じデータ構造を保持したまま処理を行う．

4.3 提案アルゴリズム

提案アルゴリズムを説明する．提案アルゴリズムは直前時刻からのオブジェクトの状態変化の大きさに応じて，再計算処理と差分計算処理のうちでより効率的な処理方法を選択するハイブリッド型のアルゴリズムである．

まず，提案アルゴリズムの流れを説明し，その後差分計算処理，再判定処理のそれぞれの処理の流れを説明する．

提案アルゴリズムをアルゴリズム 2 に示す．1-4 行目において，入力された SC オブジェクトの現在時刻での状態が含まれるセルを特定する．5 行目では，差分計算処理における再判定セル (TypeA，TypeC，TypeD セル) を分類するためのラベルを付ける．

6-7 行目において，CDS 変化率を計算し，θ 値を基に手法を選択する．CDS 変化率が θ よりも大きい場合は 8 行目で再計算処理を行い，θ 以下の場合は，10 行目で差分計算処理を行う．

アルゴリズム 2: 提案アルゴリズム

入力: Δ^{Now}
出力: CDB 外れ値オブジェクト

```
1.  for each s_q^Now ∈ Δ^Now do
2.    s_q^Now が含まれるセル C_x を特定する．
3.    if s_q^{Now-1} が含まれるセル C_{x'} と C_x が異なる then
4.      n_0(x) を 1 増やし，n_0(x') を 1 減らす．
5.    C_{x'} または C_x が TypeA，TypeC，TypeD に該当する場合は，それぞれラベル A，C，D を付ける．
6.  CDS 変化率を計算する．
7.  if CDS 変化率 > θ then
8.    再計算処理．(アルゴリズム 4 参照)
9.  else
10.   差分計算処理．(アルゴリズム 3 参照)
```

アルゴリズム 3: 差分計算処理

```
1.  for each ラベル A セル C_a do
2.    for each セル C_u ∈ L_1(C_a) do
3.      最新の n_1(u) を保持していれば n_1(u) を更新．
4.      C_u にラベル B を付ける．
5.    for each セル C_u ∈ L_2(C_a) do
6.      最新の n_2(u) を保持していれば n_2(u) を更新．
7.      C_u にラベル B を付ける．
8.  for each TypeA または B のセル C_b do
9.    CCD2 処理．(4.2.3[アイデア 2] 参照)
10. for each TypeC セル C_c do
11.   WCP．(3.2.2 参照)
12. for each TypeD セル C_d do
13.   for each SC オブジェクト O_q ∈ C_d do
14.     WOP．(3.2.2 参照)
```

4.3.1 差分計算処理のアルゴリズム

差分計算処理をアルゴリズム 3 に示す．なお，再判定セルの分類で用いるラベル A，C，D は既に付与されていることを前提とする (アルゴリズム 2，5 行目参照)．1-7 行目で TypeB のセルを特定し，8-14 行目でそれぞれのラベルを基に Type を分類し，Type 毎に処理を行う．8-9 行目では TypeA，B のセルに対して CCD2 処理を行う．10-11 行目では TypeC のセルに対して WCP を行い，12-14 行目では TypeD のセルに含まれる SC オブジェクトに対して WOP を行う．

4.3.2 再計算処理のアルゴリズム

再計算処理の流れをアルゴリズム 4 に示す．処理の流れは，3.2.2 で述べたセルを用いたアルゴリズムに従う．

ただし，4.2.4 で述べた通り，各セルが n_1，n_2 値を保持する点が異なる．13 行目において，セル C_w の $n_1(w)$ の値を計算した場合は，14 行目において，C_w はその値を保持しておく．同様に，18 行目において，セル C_w の $n_2(w)$ の値を計算した場合は，19 行目において，C_w はその値を保持しておく．

5. 実　　験

本実験では，2 次元の人工データを用いて閾値 θ の値の評価

アルゴリズム 4:　再計算処理

```
1.  for each セル C_x do
2.    if n_0(x) > M then
3.      C_x を red とする.
4.  for each red セル C_r do
5.      for each C_r の L_1 近傍のセル C_u
6.        if C_u が red でない場合 then
7.          C_u を pink とする.
8.  for each セル内にオブジェクトが存在し，色がついていないセル C_w do
9.    n_1(w) ← Σ_{C_i∈L_1(C_w)} n_0(i)
10.   n_1(w) の値を保持する.
11.   if n_0(w) + n_1(w) > M then
12.     C_w を pink にする.
13.   else
14.     n_2(w) ← Σ_{C_i∈L_2(C_w)} n_0(i)
15.     n_2(w) の値を保持する.
16.     if n_0(w) + n_1(w) + n_2(w) ≦ M then
17.       C_w を yellow とし，C_w 内の全オブジェクトを外れ値とする.
18.     else
19.       C_w を white とする.
20.       WCP.（3.2.2 参照）
```

と，提案アルゴリズムと比較手法の処理時間を評価を行った．比較手法としては再計算法のみを用いた手法 (NCB と略記) と，差分計算法のみを用いた手法 (ICB と略記) を用いた．

実験に使用したのは，AMD Athlon(tm) 64 × 2 Dual Coure Processor 3800+ 2GHz の CPU と 1982MB のメインメモリを持つ Microsoft Windows Vista マシンであり，全てのアルゴリズムの実装は Java 1.6.0 02 で行った．

5.1　実験データ

処理時間や振る舞いの特徴を定量的に測るための人工データとして，移動体オブジェクトの動きをシミュレートした人工データを使用した．使用した人工データは，各移動体オブジェクトの現在位置情報 (x 座標と y 座標の値) をオブジェクトの状態として持つデータストリームとした．なお，移動体の位置情報は 1 秒毎に更新される．

各パラメタ (全オブジェクト数，SC オブジェクトの割合，SC オブジェクトの移動量) とオブジェクトの状態の分布は次のように設定し，それぞれのパラメタを組み合わせて実験を行い，処理時間を測った．全オブジェクト数は 10000〜50000，全オブジェクトに対する SC オブジェクトの割合は 10〜50% ，SC オブジェクトの移動量は 0.5〜20.5 をそれぞれ用いた．オブジェクトの状態の分布としては，移動体オブジェクトが存在する位置を上部には疎に，下部には密に設定した分布と，移動体オブジェクトの x 座標，y 座標がそれぞれ正規分布に従うように設定し，中央部分が密に，周辺部分が疎になるように設定した分布の 2 種類を用いた．

5.2　θ の値の評価

5.2.1　実 験 概 要

提案アルゴリズムにおける閾値 θ 値を決めるために，ICB と NCB の処理時間が等しくなる CDS 変化率を求める式を重回帰分析により求めた．なお，ICB と NCB の処理時間が等しくなる時の CDS 変化率を θ 値とした．

5.2.2　θ 値の概算式

ICB と NCB の処理時間に影響を与えるパラメタ (全オブジェクト数，M，D，SC オブジェクトの割合，SC オブジェクトの移動量) を多様に変化させ，ICB と NCB の処理時間を計測した．M は 1〜5，D は 6〜10 に設定し，5.1 で述べた実験データを用いた．

計測した処理時間から，NCB と ICB の処理時間と CDS 変化率，処理の前に与えられ，かつ処理時間に影響を与えるパラメタ (全オブジェクト数，M) の関係を表す式 (1) を重回帰分析により求めた．なお，ICB の処理時間を T_{ICB}，NCB の処理時間を T_{NCB} と表し，全オブジェクト数を N で表す．

$$\frac{T_{NCB}}{T_{ICB}} = CDS\ 変化率^{\alpha} \times M^{\beta} \times N^{\gamma} \times 2^{\delta} \qquad (1)$$

解析を行った結果 $\alpha = -0.5133$，$\beta = -0.2414$，$\gamma = 0.2206$，$\delta = 1.3526$ が得られた．なお，重相関係数は 0.7664 であった．上記の式において，$\frac{T_{NCB}}{T_{ICB}} = 1$ となる場合の CDS 変化率を θ とし，θ に関する式に変換した式を次に示す．

$$\theta = (M^{\beta} \times N^{\gamma} \times 2^{\delta})^{-\frac{1}{\alpha}} \qquad (2)$$

5.3　処理時間の評価

5.3.1　実 験 概 要

データストリームにおけるオブジェクトの状態の直前時刻からの変化の大きさが変動した場合の，提案アルゴリズム (HCB と略記) と ICB，NCB の処理時間の評価を行った．

用いた実験データの基本的な値は，全オブジェクト数が 30000，SC オブジェクトの平均速度が $v = 1$[/ 秒] である．

5.3.2　実験データの種類

オブジェクトの状態変化の大きさを変化した時の処理時間を評価するために，オブジェクトの状態変化の大きさを人工的に変化させた 4 種類の状態変化列 (データセット 1，2，3，4) を用いた．図 4〜7 は各状態変化列の状態変化の大きさ (SC オブジェクトの割合または SC オブジェクトの平均速度) を表す．横軸は状態集合の到着時刻である．各データセットは 50 秒間の状態変化列であり，横軸は状態集合の到着時刻を表す．

図 4，6 はそれぞれ，データセット 1，3 の全オブジェクト数に対する SC オブジェクトの割合を表す．縦軸は SC オブジェクトの割合である．データセット 1 では，SC オブジェクトの割合を図 4 のようになだらかに変化させた．データセット 3 では，SC オブジェクトの割合を図 6 のように 10 秒毎に急激に変化させた．なお，SC オブジェクトの割合以外の値は基本的な値とした．

図 5，7 はそれぞれ，データセット 2，4 の SC オブジェクトの平均速度表す．縦軸は SC オブジェクトの平均速度である．データセット 2 では，SC オブジェクトの平均速度を図 5 のようになだらかに変化させた．データセット 4 では，SC オブジェクトの平均速度を図 7 のように 10 秒毎に急激に変化させた．なお，SC オブジェクトの平均速度以外の値は基本的な値とした．

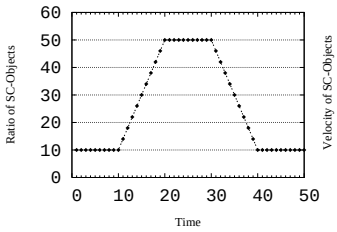

図 4　データセット 1

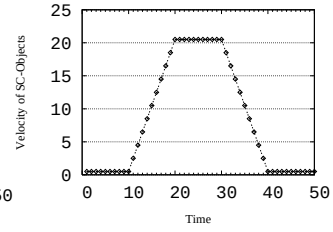

図 5　データセット 2

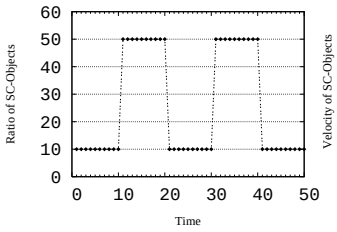

図 6　データセット 3

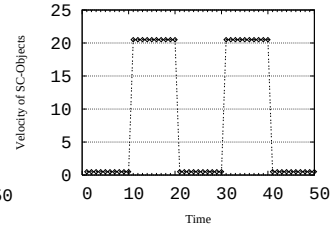

図 7　データセット 4

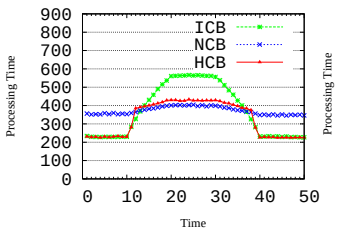

図 8　N=30000，データセット 1

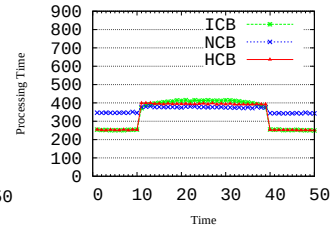

図 9　N=30000，データセット 2

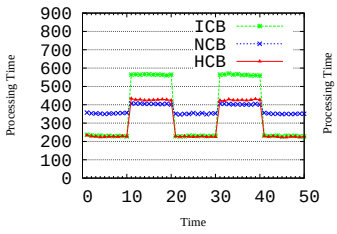

図 10　N=30000，データセット 3

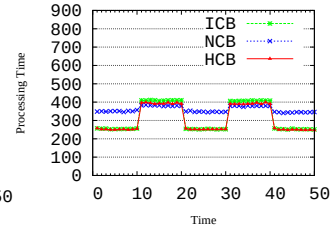

図 11　N=30000，データセット 4

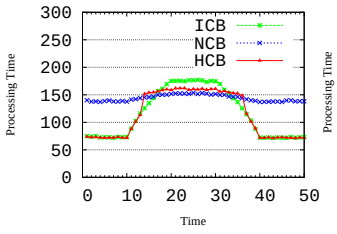

図 12　N=10000，データセット 1

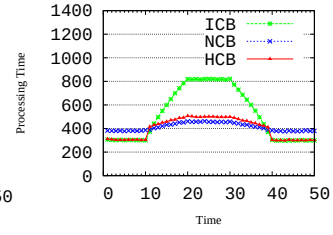

図 13　N=50000，データセット 1

5.4　実験結果と考察

実験では，データセット 1〜4 に対して，新たな状態の到着に伴う外れ値オブジェクトの検出に必要な毎回の処理時間の測定を行った．HCB の θ 値は式 (2) を用いて決定した．全オブジェクト数が 30000 の時のデータセット 1〜4 の実験結果を図 8〜11 に，全オブジェクト数が 10000，50000 の時のデータセット 1 の実験結果を図 12，13 にそれぞれ示す．なお，横軸はデータセット 1〜4 で示した，状態集合の到着時刻を示す．

図 8〜13 の結果から，HCB は ICB と NCB を比較した時に処理時間の短い手法を選択して処理を行えていることが分かる．すなわち，今回の実験においては，θ が適切に設定されており，θ 値を設定するための式として式 (2) が有用であったことが分かる．ただし，今回は対象としていないデータ分布や次元数などの処理時間に影響を与える要因が変化した場合等に関しては，今後検討する必要がある．また，HCB において再計算処理を用いた場合は，元々の NCB の処理時間よりも若干長くはなっているが，大幅な違いは見られないと言える．さらに，オブジェクトの状態変化の大きさがなだらかに変わるデータセット (データセット 1，2) と，急激に変わるデータセット (データセット 3，4) を用いた場合の両者においても，HCB では明らかなオーバヘッドが生じることなく処理の切り替えが行えていることが分かる．実験結果からオブジェクトの状態変化の大きさが変動した場合でも，HCB が NCB，ICB と比べて最も効率的に処理が行える手法であると言える．

6.　ま　と　め

本研究では，データストリームに対するハイブリッド型の連続的 CDB 外れ値検出手法を提案した．セルを用いたアルゴリズムをベースとした再計算処理と差分計算処理を，データストリームとして表現される状態の変化の大きさより，処理時間が短くなる手法を選択して処理を行う手法を提案した．さらに実験により，提案手法が最も効率的に処理を行うことができることを示した．今後の課題としては，提案手法における処理の切り替え方法の改良が挙げられる．

謝辞　本研究の一部は科学研究費補助金特定領域研究 (＃21013004) による．

文　　献

[1] V.Barret and T. Lewis, Outliers in statistical data, Wiley, Chichester, 2001.
[2] E.Eskin, "Anomaly detection over noisy data using learned probability distributions," ICML, 1994.
[3] E.M.Knorr and R.T.Ng, "Algorithms for mining distance-based outliers in large datasets," VLDB, 1998.
[4] E.M.Knorr and R.T.Ng, "Finding intensional knowledge of distance-based outliers," VLDB, 1999.
[5] E.M.Knorr, R.T.Ng and V.Tucakov, "Distance-based outliers: algorithms and applications," VLDB, 2000.
[6] R.T.Ng and J.Han, "Efficient and effective clustering methods for spasial data mining," VLDB, 1994.
[7] M.Ester, H.P.Kriegel and X.Xu, "A datavase interface for clustering in large spatial databases," KDD, 1995.
[8] M.M.Breunig, H.P.Kriegel, R.T.Ng and J.Sander, "Lof: identifying density-based local outliers," SIGMOD, 2000.
[9] K.Yamanishi and J.Takeuchi, "Discovering outlier filtering rules from unlabeled Data: Combining a Supervised Learner with an Unsupervised Learner," KDD 2001.
[10] K.Ishida and H.Kitagawa, "Detecting Current Outliers: Continuous Outlier Detection over Time-Series Data Streams," DEXA 2008.
[11] F.Angiulli and F.Fassetti, "Detecting distance-based outliers in streams of data," CIKM 2007.