Vol.2016-DBS-163 No.13
Vol.2016-IFAT-123 No.13
2016/9/14

データの動的な変化に対応可能な対話的外れ値分析

坂詰 知完[1,a)]　北川 博之[2,b)]　天笠 俊之[2,c)]

概要：データマイニングにおいてデータセットの中で他と大きく異なる値をもつ外れ値を検出することは重要なタスクとなっている．代表的な外れ値検出手法の一つとして距離に基づく手法があるが，ユーザが求める外れ値を検出するための適切なパラメタの選択が容易でないことが知られている．この問題に対し，ユーザが求める外れ値検出に適切なパラメタ選択を支援する対話的な外れ値分析手法 ONION が提案されている．ONION では対象データ集合を事前に分析し索引構造を構築することで，種々の対話的分析を支援する．しかし，対象データ集合に更新がある場合の索引構造の更新については考慮されていない．本研究では，ONION の索引構造に加えて，カウンタ付きグリッド索引を組み合わせて用いることで，データの動的な変化にも対応可能な対話的外れ値分析手法を提案し，実験によりその有用性を評価する．

Interactive Outlier Analysis on Dynamic Datasets

Sakazume Chihiro[1,a)]　Kitagawa Hiroyuki[2,b)]　Amagasa Toshiyuki[2,c)]

1. はじめに

近年，データマイニングにおいてデータセットの中で他と大きく異なる値を持つ外れ値を検出することは重要なタスクとなっている．外れ値は，「異なるメカニズムで生成された疑いを喚起するようなほかの観測値から大きく外れた観測値」と Hawkins によって定義されている [1]．外れ値検出はクレジットカードの不正使用検出や監視ビデオでの異常な動きの検出など多くのアプリケーションで用いられており，現在，外れ値検出手法は様々なものが提案されている．代表的な検出法として距離に基づく手法 [2] や最近傍に基づく手法 [3]，密度に基づく手法 [4]，クラスタリングに基づく手法 [5][6]，角度に基づく手法 [7][8] などがあげられる．

現在，広く一般的に用いられている外れ値検出は距離を用いた手法であるが，パラメタの設定によっては外れ値を適切に検出できないという問題がある．そこで，ユーザが求める外れ値を検出するためのパラメタ設定を適切し，対話的な外れ値分析を行う ONION[9] が提案された．しかし，ONION は静的なデータを想定しており，データの動的な変化には対応していない．一方で近年，センサー等から得られる動的に変化するデータの増加により，外れ値分析にリアルタイム性が求められており，[7] をデータの変動に対応させる手法 [10] などが提案されている．

そこで，本研究では，ONION を動的に変化するデータに対応させ，データが変化した場合でも効率的に外れ値分析を行う手法を提案する．実験評価では，提案法を用いずに ONION を動的環境に対応させた場合との速度比較を行う．

2. 距離に基づく外れ値検出

2.1 基本的な考え方

距離に基づく外れ値検出は，対象オブジェクトとその他すべてのオブジェクト間の距離を計算し，距離 r 内にあるオブジェクト数が k 個より少ないとき外れ値とする手法である [2]．図 1 に距離に基づく外れ値検出の例を示す．左右の図は同一のオブジェクト集合を示しており，左側の図ではオブジェクト A から r 内に 4 つのオブジェクトが存在し，右側の図ではオブジェクト B から r 内に 2 つのオブ

[1] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba
[2] 筑波大学計算科学研究センター
Center for Computational Sciences, University of Tsukuba
[a)] sakazume@kde.cs.tsukuba.ac.jp
[b)] kitagawa@cs.tsukuba.ac.jp
[c)] amagasa@cs.tsukuba.ac.jp

ジェクトが存在することがわかる．k=3 としたとき，r 内に 3 個未満の場合は外れ値となるため，オブジェクト A は正常値となる一方で，オブジェクト B は外れ値となる．

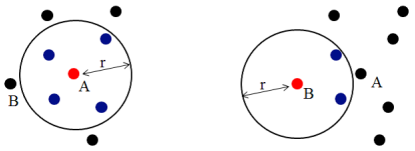

図 1　距離に基づく外れ値検出

2.2　問題点

各オブジェクトが外れ値になるか正常値になるかの基準はパラメタ k と r によって決定される．そのため，パラメタの設定を適切にしなければ，本来正常値であるデータを外れ値，外れ値であるべきデータが正常値に分類されてしまう．図 2-5 では k の値を変えることによって，データの分類が変わる様子を示している．図 2 ではオブジェクトの分布を示しており，オブジェクト A と B をユーザが外れ値として検出したいオブジェクトとする．図 3 では k=4 とした場合に外れ値に分類されるオブジェクトを示している．このとき，オブジェクト A と B を外れ値として検出することができているが，その他のオブジェクト C，D，E，F も外れ値として検出され，本来正常値として分類されるべきオブジェクトも含めて外れ値に分類されてしまう．一方で，図 4 では k=2 とした場合に外れ値に分類されるオブジェクトを示しているが，外れ値として検出したいオブジェクト A と B の内，オブジェクト B を外れ値として検出することができていない．今回の場合は k=3 とすることで図 5 に示されるようにオブジェクト A，B に加えてそれらと類似した特徴をもつオブジェクト E のみを外れ値として検出することができる．距離に基づく外れ値検出を行う場合は，パラメタの設定を適切にすることが重要と言える．

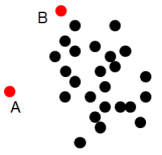

図 2　オブジェクトの分布

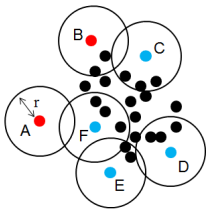

図 3　k=4 の場合

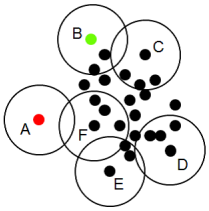

図 4　k=2 の場合

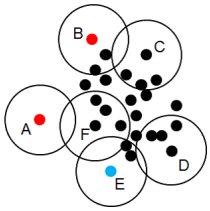

図 5　k=3 の場合

3.　ONION

ONION は k の取り得る範囲 $[k_{min},k_{max}]$ と r の取り得る範囲 $[r_{min},r_{max}]$ を与え，その範囲内でユーザが求める外れ値を検出するための適切なパラメタの組合せ (k,r) を示すことを可能にする．組合せによって外れ値か外れ値ではないかが変わってくるが，ONION では外れ値と正常値，外れ値候補の 3 種類に分類する．対象とするオブジェクトを O，O から k 番目のオブジェクトまでの距離を D_O^k としたとき，$D_O^{k_{max}} \leq r_{min}$ ならば O は上記の k と r の範囲内で常に正常値となる．一方で，$D_O^{k_{min}} > r_{max}$ ならば O は上記の k と r の範囲内で常に外れ値となる．どちらにも満たさないオブジェクトは k と r の値によって外れ値に成り得るオブジェクトなので外れ値候補に分類される．

ONION は外れ値候補に分類されたデータのみから構成される 3 種類の構造をもち，それぞれ ONION 空間 (O-Space)，パラメタ空間 (P-Space)，データ空間 (D-Space) と呼ばれる．各空間で外れ値分析を行うことで，ユーザが求める外れ値を検出することができる．外れ値分析は，Comparative Outlier Analytics(CO) と Outlier-Centric Parameter Space Exploration(PES)，Outlier Detection(OD) の 3 種類がある．

3.1　P-Space

本研究では，P-Space を用いているため P-Space の詳細を説明する．図 6 のように各デオブジェクトに ID が割り振られ，図 7 で示されるように P-Space は k ごとに各外れ値候補 oc から k 番目に近いオブジェクトまでの距離を昇順で保持し，図 6 に対応した ID をもつ．図 7 では k_{min} において ID が 2,90,...,57,40 のデータが外れ値候補であり，各外れ値候補から k_{min} 番目との距離をもつことを示している．

ID	座標
1	(x_1, y_1)
2	(x_2, y_2)
⋮	⋮
N	(x_N, y_N)

図 6　オブジェクト表

ID	距離
2	$D_{oc_1}^{k_{min}}$
90	$D_{oc_2}^{k_{min}}$
⋮	⋮
57	$D_{oc_{n-1}}^{k_{min}}$
40	$D_{oc_n}^{k_{min}}$

k_{min}

ID	距離
2	$D_{oc_1}^{k_{min}+1}$
90	$D_{oc_2}^{k_{min}+1}$
⋮	⋮
57	$D_{oc_{n-1}}^{k_{min}+1}$
40	$D_{oc_n}^{k_{min}+1}$

$k_{min}+1$

…

ID	距離
2	$D_{oc_1}^{k_{max}}$
90	$D_{oc_2}^{k_{max}}$
⋮	⋮
40	$D_{oc_n}^{k_{min}}$
57	$D_{oc_{n-1}}^{k_{max}}$

k_{max}

図 7　P-Space

Vol.2016-DBS-163 No.13
Vol.2016-IFAT-123 No.13
2016/9/14

3.2 P-Space を用いた効率的な外れ値分析

P-Space に対し，例として PSE によって外れ値分析を行う場合を説明する．PSE では外れ値集合 O_{in} を入力とする．O_{in} に近いオブジェクト集合を外れ値とするパラメタの範囲を調べ，$\delta(-1 < \delta < 1)$ でその近さを調整する．もし，得られた外れ値の数が多すぎる場合は，$\delta \leq 0$ を与えて，$|O_j| = (1+\delta)|O_{in}|$ となる r と k の組合せを得ることで外れ値の数を減らす．このとき，$O_j \subseteq O_{in}$ となる．得られた外れ値の数が少なすぎる場合は，$\delta \geq 0$ を与えて，$|O_j| = (1+\delta)|O_{in}|$ となる r と k の組合せを得ることで外れ値の数を増やす．このとき，$O_j \supseteq O_{in}$ となる．図 8 に示す P-Space をもつ場合の動作を説明する．{57,90} を O_{in} とした場合，P-Space は各 k で距離の昇順で並んでいるため，{90} が外れ値ならば，{90} 以降の {80,57,40} も外れ値になる．よって，{57,90} を外れ値にするには {90} が外れ値になるパラメタ範囲を調べればよい．図 8 では，{90} から 3 番目に近いオブジェクトとの距離は 10 であることがわかるので，k=3 で r=10 未満ならば {90} は外れ値になる．{90} の前にある {2} を外れ値にする必要はないので r=3 以上となる．また，{90} から 4 番目に近いオブジェクトとの距離は 17 なので，k=4 で r=17 未満ならば {90} は外れ値になり，r=5 以上となる．同様に {90} から 5 番目に近いオブジェクトとの距離は 22 であり，k=5 で r=22 未満ならば {90} は外れ値になり，r=9 以上となる．よって，O_{in} を外れ値とするパラメタ範囲は (k,r)=(3,[3,10))，(4,[5,17))，(5,[9,22)) となる．このパラメタのとき，外れ値は {90,80,57,40} である．得られた外れ値の数が多すぎる場合は O_{in} を減らす．よって，{57} が外れ値となる r と k の組合せを求めることになる．図 8 では {57} の 3 番目に近いオブジェクトとの距離は 30 であり，k=3 で r=30 未満ならば {57} は外れ値になる．{80} は外れ値にする必要はないので r=18 以上となる．また，{57} の 4 番目に近いオブジェクトとの距離は 40 であり，k=4 で r=40 未満ならば {57} は外れ値になり，r=20 以上となる．同様に {57} の 5 番目に近いオブジェクトとの距離は 60 であり，k=5 で r=60 未満ならば {57} は外れ値になり，r=53 以上となる．この結果，{57} を外れ値とするパラメタ範囲は (k,r)=(3,[18,30))，(4,[20,40))，(5,[53,60)) となり，外れ値は k=3 と k=4 で {57,40}，k=5 で {57} となる．

ID	距離
2	3
90	10
80	18
57	30
40	35

k=3

ID	距離
2	5
90	17
80	20
57	40
40	53

k=4

ID	距離
2	9
90	22
80	31
40	55
57	60

k=5

図 8 P-Space による外れ値分析の例

4. 動的環境における対話的外れ値分析

4.1 データの動的変化

動的環境ではオブジェクトの追加や削除，移動が考えられるため，オブジェクトの状態が外れ値から正常値，正常値から外れ値のように変化していく．動的環境において k=3 とした場合のオブジェクトの状態変化の例を図 9-11 を用いて説明する．図 9 ではオブジェクト H が追加された場合を示す．オブジェクト B は距離 r 内のオブジェクト数が 2 個だったため，オブジェクト H が追加される前は外れ値に分類されていたが，追加によって r 内にあるオブジェクト数が 3 個になり，正常値に分類される．追加されたオブジェクト H は r 内にあるオブジェクト数が 1 個であるため外れ値になる．図 10 ではオブジェクトが削除された場合を示す．オブジェクト C は r 内にあるオブジェクト数が 3 個だったため，オブジェクト G が削除される前は正常値に分類されていたが，削除によって r 内にあるオブジェクト数が 2 個になり，外れ値になる．図 11 ではオブジェクトが移動した場合を示す．オブジェクト G が H の位置に移動したとき，オブジェクト B は外れ値から正常値，オブジェクト C は正常値から外れ値，オブジェクト H は外れ値となる．

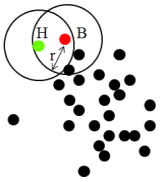

図 9 オブジェクトの追加

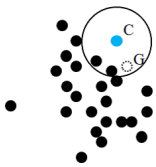

図 10 オブジェクトの削除

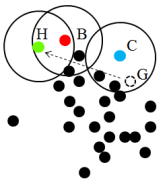

図 11 オブジェクトの移動

4.2 動的環境における ONION の問題点

ONION を動的環境に対応させる場合，問題点は 2 つある．1 つ目は各オブジェクトの状態の更新である．外れ値が外れ値候補になる場合や外れ値候補が外れ値，外れ値候補が正常値，正常値が外れ値候補など状態の変化を更新する必要がある．2 つ目がオブジェクトの状態変更による各空間がもつ値の更新である．各空間は外れ値候補のみで構成されるため，新たに外れ値候補になったオブジェクトに関する値の追加や外れ値候補ではなくなったオブジェクトに関する値は削除しなければならない．

Vol.2016-DBS-163 No.13
Vol.2016-IFAT-123 No.13
2016/9/14

5. 提案手法

本研究では，P-Space を動的環境に対応させ，各オブジェクトの状態変更を効率的に検出し，P-Space を更新する手法を提案する．提案手法では，ONION に用いている P-Space に加えて，カウンタ付きグリッド索引を用いる．また，オブジェクト表にカウンタと種別を追加する．本研究ではオブジェクトの追加と削除を対象とする．P-Space に影響のあるデータの状態変更は，追加の場合は，外れ値→外れ値候補，外れ値候補→正常値であり，削除の場合は，正常値→外れ値候補，外れ値候補→外れ値である．オブジェクトの追加や削除に応じて，各オブジェクトがもつカウンタを更新し，状態変更を踏まえて P-Space の更新を行う．

5.1 カウンタ付きグリッド索引

カウンタ付きグリッド索引はオブジェクトの位置を表す空間をグリッドで分割し，各グリッド内にあるオブジェクト数と各オブジェクトの ID を保持する．例として図 12 のようにオブジェクトが位置しているとき，各グリッドごとにオブジェクト数とオブジェクトの ID を保持する様子を図 13 に示す．例えば一番右上のグリッドにはオブジェクトが 1 つあり，そのオブジェクトは {12} であることを示す．また，図 14 に示すように各オブジェクトごとに r_{min} 以下と r_{max} 以下の範囲内のオブジェクト数を保持することで，追加や削除するオブジェクトから r_{min} 以下と r_{max} 以下の範囲内に含まれるオブジェクトのみを対象とし，カウンタを更新すればよいので状態変更を効率的に行うことができる．オブジェクトが追加されたとき，追加されたオブジェクトから r_{min} や r_{max} 以下の範囲内にあるオブジェクトはカウンタを 1 増やし，削除の場合は 1 減らす．カウンタをもつことによって毎回各グリッド内にあるオブジェクト数を計算しなくて済む．また図 14 の種別では正常値を 0，外れ値候補を 1，外れ値を 2 とする．

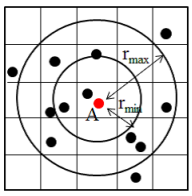

図 12 オブジェクト位置

				(1,{12})
(1,{11})	(1,{2})	(1,{10})		
	(2,{4,5})	(2,{1,6})		(1,{9})
	(1,{7})		(2,{3,8})	
			(1,{13})	

図 13 カウンタ付きグリッド索引

ID	座標	r_{min} カウンタ	r_{max} カウンタ	種別
1	(x_1, y_1)	2	5	1
2	(x_2, y_2)	10	35	0
⋮	⋮	⋮	⋮	⋮
N	(x_N, y_N)	0	1	2

図 14 オブジェクト表

5.2 状態変更アルゴリズム

オブジェクトを追加した場合のアルゴリズムを Algorithm 1 に示す．このアルゴリズムでは追加オブジェクトから r_{min}，r_{max} の範囲内に含まれるオブジェクトに関して図 14 のオブジェクト表で保持するカウンタを更新し，状態変更の有無を調べる．1-4 行目では追加オブジェクトから半径 r_{min} の円内に完全に含まれるグリッドにあるオブジェクトの r_{min} と r_{max} のカウンタをともに 1 増やす．半径 r_{min} の円上のグリッドにあるオブジェクトは，5-16 行目で r_{min} の範囲内にあるか調べるために距離計算を行う必要がある．距離が r_{min} 以下であれば両方のカウンタを増やし，r_{min} より大きいならば r_{max} のカウンタのみ増やす．17 行目以降も同様に r_{max} に完全に含まれるグリッドにあるオブジェクトは r_{max} のカウンタを増やし，半径 r_{max} の円上のグリッドにあるオブジェクトは距離計算を行う．カウンタの更新により，r_{max} カウンタが k_{min} になれば外れ値→外れ値候補，r_{min} カウンタが k_{max} になれば外れ値候補→正常値になる．オブジェクトを削除した場合は Algorithm 1 で 1 増やしていた箇所を 1 減らす処理を行う．削除の場合は r_{min} カウンタが $k_{max} - 1$ になれば正常値→外れ値候補，r_{max} カウンタが $k_{min} - 1$ になれば外れ値候補→外れ値になる．

5.3 P-Space 更新アルゴリズム

P-Space の更新アルゴリズムを Algorithm 2 に示す．1-5 行目では外れ値候補→正常値または外れ値候補→外れ値になった場合にそのオブジェクト ID に対応した箇所を P-Space から削除する処理を行う．外れ値候補のままだったオブジェクト O_{oc} に関しては 6-19 行目の処理を行う．追加や削除されたオブジェクトとの距離によっては O_{oc} の k_{min} から k_{max} 番目までの距離が変わる可能性がある．追加の場合，追加オブジェクトとの距離が k_{min} 番目に近いオブジェクトとの距離以上かつ k_{max} 番目に近いオブジェクトとの距離以下ならば，追加オブジェクトとの距離が何番目の距離に該当するのかを調べ，該当する k 番目の距離として保持する．追加以前に k 番目として保持していた距離は k+1 番目の距離，k+1 番目として保持していた距離は k+2 番目の距離となり，k_{max} 番目の距離まで更新を行う．追加オブジェクトとの距離が k_{min} 番目に近いオブジェクトとの距離より小さい場合，$k_{min} - 1$ 番目に近いオブジェクトとの距離が新しく k_{min} 番目の距離となるため

$k_{min}-1$ 番目の距離を求める必要がある．この場合にはカウンタ付きグリッド索引を用い，O_{oc} があるグリッドから近傍のオブジェクト数が $k_{min}-1$ 個になるまでグリッド範囲を広げていき，その範囲にあるオブジェクトとの距離計算により $k_{min}-1$ 番目の距離を求める．その後 k_{min}-1 番目の距離は k_{min} 番目の距離として保持し，k_{min} 番目だった距離は $k_{min}+1$ 番目の距離となり，k_{max} まで同様に更新する．削除の場合，削除オブジェクトとの距離が k_{max} 番目に近いオブジェクトとの距離以下ならば $k_{max}+1$ 番目に近いオブジェクトとの距離を求める必要がある．$k_{max}+1$ 番目の距離を k_{max} 番目の距離，k_{max} 番目だった距離が $k_{max}-1$ 番目の距離となり，削除されたオブジェクトとの距離を保持していた箇所まで更新を行う．

Algorithm 1 状態変更アルゴリズム

for each 半径 r_{min} の円に完全に含まれるグリッド **do**
  グリッド内にあるオブジェクトの $counter_{min}$, $counter_{max}$ を+1.
  カウンタに応じて状態変更.
end for
for each 半径 r_{min} の円上にあるグリッド **do**
  for each グリッド内にあるオブジェクト **do**
    追加オブジェクトとの距離 dist を算出.
    if $dist \leq r_{min}$ **then**
      $counter_{min}$ と $counter_{max}$ を+1.
      カウンタに応じて状態変更.
    else
      $counter_{max}$ を+1.
      カウンタに応じて状態変更.
    end if
  end for
end for
for each 半径 r_{max} の円に完全に含まれ，半径 r_{min} の円には含まれず，半径 r_{min} の円上ではないグリッド **do**
  $counter_{max}$ を+1.
  カウンタに応じて状態変更.
end for
for each 半径 r_{max} の円上にあるグリッド **do**
  for each グリッド内にあるオブジェクト **do**
    追加オブジェクトとの距離 dist を算出.
    if $dist \leq r_{max}$ **then**
      $counter_{max}$ を+1.
      カウンタに応じて状態変更.
    end if
  end for
end for

Algorithm 2 P-Space 更新アルゴリズム

for each 外れ値候補→正常値と外れ値候補→外れ値になったオブジェクト O_n **do**
  for $i=1$ to $k_{max}-k_{min}+1$ **do**
    P-Space から O_n に対応する箇所を削除
  end for
end for
for each オブジェクトの追加で外れ値候補のままだったオブジェクト O_{oc} **do**
  O_{oc} と追加されたオブジェクト O_{new} の距離 dist を算出
  if O_{oc} から k_{min} 番目の距離 $\leq$ dist $\leq$ O_{oc} から k_{max} 番目の距離 **then**
    dist を O_{oc} の該当する k 番目の距離とし，k 番目の距離を k+1，k+1 の距離を k+2,... とする.
  else if $dist < O_{oc}$ から k_{min} 番目の距離 **then**
    $k_{min}-1$ 番目の距離を求め，$k_{min}-1$ 番目の距離を k_{min}，k_{min} 番目の距離を $k_{min}+1,\ldots$ とする
  end if
end for
for each オブジェクトの削除で外れ値候補のままだったオブジェクト O_{oc} **do**
  O_{oc} と削除されるオブジェクト O_{delete} の距離 dist を算出
  if dist $\leq$ O_{oc} から k_{max} 番目の距離 **then**
    O_{oc} から $k_{max}+1$ 番目の距離を求め，$k_{max}+1$ の距離を k_{max}，k_{max} の距離を $k_{max}-1,\ldots$ とする.
  end if
end for
for each 外れ値候補になった O_{new} と外れ値→外れ値候補，正常値→外れ値候補になったオブジェクト **do**
  k_{min} から k_{max} までの距離を求めて P-Sapce に追加する.
end for

6. 実験評価

本実験ではオブジェクトの追加と削除のそれぞれで P-Space の更新が完了するまでの時間を計測する．パラメータは $k_{min}=3$，$k_{max}=8$，$r_{min}=50$，$r_{max}=150$ を与え，グリッドの対角線の長さを $r_{min}/2$ としている．使用したマシンは Intel Core i7 3.70GHz の CPU と 64GB のメモリをもつ Windows8 のマシンである．実装は MATLAB を用いた．

6.1 比較手法

提案手法との比較のためにカウンタなしグリッド索引を用いる手法とデータの更新があった場合に毎回 P-Space を再計算する手法の 2 種類を用いる．カウンタなしグリッド索引では，追加や削除されるオブジェクトから r_{min} と

r_{max} 内にあるオブジェクトの状態更新の際に，そのオブジェクトから r_{min} と r_{max} 内にあるグリッドがもつオブジェクト数をカウントし直す必要がある．毎回 P-Space を再計算する手法では，オブジェクト追加の場合に外れ値と外れ値候補のオブジェクトで全オブジェクト間の距離を計算し，オブジェクト削除の場合には正常値と外れ値候補のオブジェクトで全オブジェクト間の距離を計算する必要がある．

6.2 実験データ

実験は平均 0，標準偏差 250 のガウス分布によって生成した人工データを使用した．次元数は 2 次元でデータ数は 10000 である．追加と削除を行う前の状態は外れ値が 13 個，外れ値候補が 164 個となっている．追加や削除するオブジェクトは正常値になるようにした．

6.3 実験結果

オブジェクトを追加したときの実験結果を図 15 に示す．図 15 からわかるように提案手法であるカウンタ付きグリッド索引が最も更新時間が短い結果になった．一方で，毎回 P-Space を再計算する手法は距離計算に時間がかかるため，ほかの手法と比べて更新時間が大きくかかることが確認できる．オブジェクトを削除したときの実験結果を図 16 に示す．図 16 から明らかなように提案手法であるカウンタ付きグリッド索引が最も更新時間が短い結果になった．オブジェクトの状態変更をする際，削除オブジェクトから半径 r_{min} と r_{max} の円内にあるオブジェクトを対象に確認しなければならないが，正常値のオブジェクトを削除する場合，円内には多くの正常値オブジェクトが存在すると考えられる．よって正常値→外れ値候補への状態変更を確認しなければならないため，カウンタをもたないグリッド索引では大きく時間がかかった．一方でオブジェクトを追加する場合は正常値のオブジェクトの状態変更はないため，状態変更の確認をすべきオブジェクトが少なくなり，カウンタ付きグリッド索引との更新時間差も少なかったと考えられる．

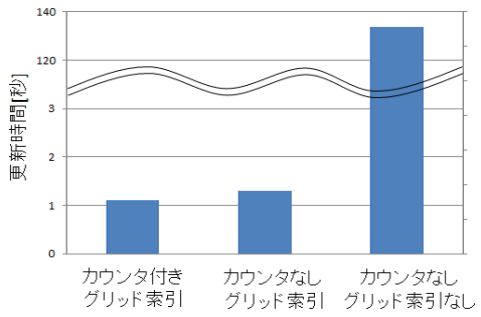

図 15 オブジェクト追加による更新時間の比較

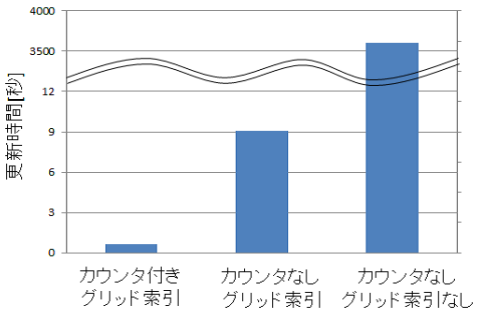

図 16 オブジェクト削除による更新時間の比較

7. まとめ

本研究では，ONION を動的環境に対応させ，動的に変化するデータに対応可能な対話的外れ値分析手法の提案と，その有効性を示すために 2 種類の手法と比較して実験を行った．結果として，提案手法が最も効率的に P-Space を更新することが確認できた．

今後の課題として，より多様なデータセットを用いた比較評価を行うことが考えられる．

謝辞 本研究の一部は，科研費・基盤研究 (B)(26280037) の助成による．

参考文献

[1] D. Hawkins. Identification of Outliers. Chapman and Hall,London, 1980.
[2] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In Proc. VLDB, 1998.
[3] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In Proc. SIGMOD, 2000.
[4] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. LOF: Identifying density-based local outliers. In Proc. SIGMOD, 2000.
[5] Jiang,M,F.,Tseng,S.S.,Su,C.M.:Two-phase Clustering Process for Outlier Detection. Pattern Recognition Letters,2001,22(6-7):691-700
[6] Guha, S., Rastogi, R. and Shim, K.,“ROCK: A Robust Clustering Algorithm for Categorical Attributes,”ICDE 15, 1999.
[7] H.-P. Kriegel, M. Schubert, and A. Zimek. Angle-based Outlier Detection in High-dimensional Data. In Proceedings KDD’08, pages 444-452, 2008.
[8] R. P. N. PhamA. A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data. Proc. ACM SIGKDD, pages 877-885, 2012.
[9] Lei Cao, MingruiWei, Di Yang, Elke A. Rundensteiner. Online Outlier Exploration Over Large Datasets. KDD’15, 2015.
[10] Hao Ye, Hiroyuki Kitagawa, Jun Xiao. Continuous Angle-based Outlier Detection on High-dimensional Data Streams. IDEAS’15, 2015.