DEIM Forum 2017 I2-3

局所例外部分データの自動探索

小笠原麻斗† 水野 陽平† 佐々木勇和† 鬼塚 真†

† 大阪大学大学院情報科学研究科 〒 565–0871 大阪府吹田市山田丘 1-5

E-mail: †{ogasawara.asato,mizuno.yohei,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし ビジネスデータの主流な解析方法である OLAP（online analytical processing）を効率的に行うため，例外的な分析観点や部分データを自動探索し有用性の高い分析結果を分析者に推薦する技術が注目されている．既存技術では，販売データ等の時期性や地域性の影響が大きいデータにおいて，局所性を考慮して時期性や地域性が例外的な部分データを特定することができない．しかしこれは，部分データの時期性や地域性を局所的に見て分析結果の例外度を評価することで解決できる．本研究では，1) 局所例外部分データの自動探索手法と 2) 自動探索手法の高速化手法を提案する．提案手法の特徴として，1) 自動探索手法では部分データの例外度の定量化に局所外れ値検知手法 LOF を用いている点，2) 高速化手法では部分データ探索空間のセル分割による効率的な k 近傍探索とセルの重心点による LOF の上限・下限の推定を行っている点が挙げられる．高速化手法の実行時間を計測した結果，高速化なしの手法と比較して最大約 2 倍の速度上昇を確認でき，部分データ量が多くなるほど大幅に性能が改善する．

キーワード 探索的データ解析，OLAP，LOF

1. はじめに

ビッグデータ時代が到来し，企業が収集・蓄積するデータの大規模化・多様化が続いている．それに伴い，収集・蓄積したデータから有益な情報を抽出するため，OLAP（online analytical processing）や相関ルールマイニングなどの多くのデータマイニング技術が開発されてきた [1]．特に OLAP は，ビジネスデータの主流な解析方法として頻繁に用いられている．OLAP は，分析者が選択した分析観点やデータに基づき生成される分析結果から，データの傾向を把握する分析方法である．有用な分析結果は，往々にして全データの平均的な傾向から乖離した例外データから導き出すことができる．分析者はまず OLAP により例外データを発見し，その後それらの要因の調査により有効な知見を獲得することで企業の意思決定に役立てる．しかし OLAP では，分析観点やデータを選択して分析結果を確認するという一連の作業を有用な分析結果が得られるまで繰り返し行う必要があり，分析者にとって大きな負担となっている．

上記の問題の解決のため，探索的データ解析（Exploratory analysis）技術の研究が活発に行われている [2–6]．これらの研究では，例外的な分析結果を生み出す分析観点や部分データを自動で探索することによって分析者にとって有用性の高い分析結果を特定し，その分析結果を分析者に推薦する．部分データとは，分析対象のデータ全体から特定の条件（例えば，商品カテゴリー＝'T シャツ'）で選択した部分的なデータを指す．分析結果の例外度は，部分データの傾向（時期性や地域性）とデータ全体の傾向との乖離の大きさで算出されることが一般的である．すなわち，部分データの傾向が全体平均からどの程度離れているかという大域的な捉え方で例外度を定量化する．水野らの手法 [5, 6] は，ある分析観点において，全体データの分析結果との乖離が最も大きい分析結果を出す部分データ上位 n 件を特定する．この手法では例えば，販売データにおいて通常より販売数が鈍化している商品や店舗を，全商品や全店舗の平均販売数の傾向との比較により判断する．

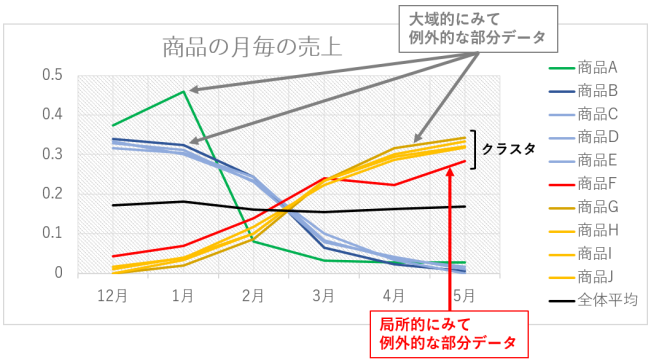

図 1 商品の月毎の売上（全月の売上を 1 として正規化）

部分データの傾向（地域性や時期性）の例外性は，通常の傾向から如何に離れているかで判断することが重要であるため，この手法のように，データ全体の平均傾向という大域的な傾向を部分データの通常の傾向とする方法では，局所的な傾向を共有する部分データの集合の通常の傾向を捉えられない．例えば実際の例として，同時期に購入される傾向が多いカフリンクスやネクタイピン，サスペンダーなどの紳士服関連アクセサリーの中で，仮にカフリンクスのみ売上傾向が異なっている場合にカフリンクスを特定することができない．特に，販売データ等の地域性（都市圏や地方などの地域依存する傾向）や時期性（夏や冬などの時期に依存する傾向）の偏りが大きいデータの分析においては，各部分データの傾向の偏りを考慮し，部分データを局所的に見てその中で例外的な傾向を示す部分データ（以後，局所例外部分データ）を特定することが重要である．

局所例外部分データを特定することの重要性を例を用いて説明する．例として，販売データの時期性の影響を見て販売戦略を決めるケースを考える．図 1 は，商品 A〜J の 10 個の部分

データの，月毎の売上を表している．グラフ上の線一つ一つが各部分データの月毎の売上を表し，中央の黒色の線は全体データの月毎の売上の平均を表す．商品 A,B,G は，全体平均からの乖離が大きいため，[5,6] の手法では大域的な例外度が高く算出される部分データであるが，商品 F は，全体平均からの乖離は小さいため例外度は最も低く算出される．しかし，商品 F,G,H,I,J の 5 つを春季に売上が大きい春物商品のクラスタとすると，商品 F はそのクラスタ内の商品の中では唯一 4 月の前月比売上が下がるという例外的な傾向を示すため時期の影響は大きいと判断できる（つまり，商品 F は局所例外部分データである）．よって，この原因を調査し，翌月の 5 月に商品 F の値引きセールを実施するなどの販売戦略を決定することで商品 F の売上を改善できる可能性があるため，商品 F は分析者にとって有用性が高い部分データといえる．このように，似た傾向を持つ部分データ（この例では，季節変動が似ている商品）を局所的にみて例外的な傾向を持つ部分データを特定することは重要である．

本稿では，局所例外部分データを自動探索する問題に取り組む．また，データの大規模化・多様化により OLAP で使用する分析対象の部分データの数や分析観点の数の増加が続き，分析作業の効率化・高速化が求められているという背景から，本稿では自動探索手法の高速化にも取り組む．1) 自動探索手法では，局所外れ値検知手法 LOF を用いることで部分データの局所的な例外度の定量化を行う．2) 高速化手法では，部分データの探索空間全体をセルに分割することにより LOF の計算に必要な部分データの k 近傍の探索範囲をいくつかのセルに絞る**効率的な k 近傍探索**技術と，セルの重心点を用いた LOF の上限・下限の推定による**例外度上位 n 件に入りえない部分データの足切り**技術の 2 つの技術により高速化を実現する．

評価実験では，高速化手法の実行時間の計測結果によると，高速化なしの手法と比較して最大約 2 倍の速度上昇が確認できる．また，効率的な k 近傍探索成功部分データ数と足切り成功部分データ数は，全部分データ数に比例し増加することも確認できる．さらに，実験結果から全部分データ数が多くなると大幅に高速化の性能改善が見込めることが分かる．

本稿の構成は，次の通りである．2 章で本稿の前提となる知識について概説する．3 章で提案する自動探索手法の詳細について説明し，4 章で提案する高速化手法の詳細について説明する．5 章で提案手法の評価を行い，6 章で提案手法により得られた分析結果について説明する．7 章で関連研究について述べ，8 章で本稿をまとめ，今後の課題について論ずる．

2. 前提知識

本研究では，部分データの局所的な例外度を定量化する技術として，LOF（local outlier factor）[7] を用いる．本章では，LOF について概説する．LOF は，密度に基づく代表的な外れ値検知手法である．LOF では，データセット内の要素（例えば，商品 1 つ 1 つ）を N 次元ユークリッド空間上の 1 つの点とみなし，点それぞれに対して例外度を表す LOF 値を算出する．ある点 A は，0 以上の実数 a_i（$1 \leqq i \leqq N$）の N 個の組で表される座標として，次式で定義される．

$$A := [a_1, a_2, \cdots, a_N] \tag{1}$$

A の LOF 値 $LOF_k(A)$ は，近傍の範囲を表すパラメータ k を用いて次式で定義される．

$$LOF_k(A) := \frac{\sum_{B \in N_k(A)} lrd_k(B)}{|N_k(A)|} / lrd_k(A) \tag{2}$$

但し，B は $N_k(A)$ の点を表し，$lrd_k(A)$ ($lrd_k(B)$) は $A(B)$ の局所到達可能密度（local reachability density）を表す．つまり，A の LOF 値は，$lrd_k(B)$ の平均と $lrd_k(A)$ の比で表される．k は分析者が事前に指定する．$lrd_k(A)$ は，次式で定義される．

$$lrd_k(A) := 1/(\frac{\sum_{B \in N_k(A)} reach\text{-}dist_k(A,B)}{|N_k(A)|}) \tag{3}$$

但し，$reach\text{-}dist_k(A,B)$ は，A から B への到達可能距離（reachability distance）を表す．つまり，$lrd_k(A)$ は，A から各 B への到達可能距離の平均の逆数であるため，この到達可能距離の平均が大きいほど，$lrd_k(A)$ は小さくなる．$reach\text{-}dist_k(A,B)$ は，次式で定義される．

$$reach\text{-}dist_k(A,B) := \max\{d(A,B), k\text{-}distance(B)\} \tag{4}$$

但し，$d(A,B)$ は A と B のユークリッド距離，$k\text{-}distance(B)$ は B から k 番目に近い点から B へのユークリッド距離を表す．$reach\text{-}dist_k(A,B)$ は，上記 2 つの距離の内，大きい方の距離の値をとる．

A の LOF 値は，A の局所到達可能密度と $N_k(A)$ の点の局所到達可能密度の平均によって決まる．また自身の局所到達可能密度と近傍の局所到達可能密度に差がないほど 1 に近づき，1 より大きいほど A の例外度が高いことを表す．LOF は，点の近傍との距離を用いて例外度を算出する点が局所たる所以である．本自動探索手法では，この LOF を用いて部分データ毎に例外度を算出する（3.1 節）．

3. 自動探索手法

本研究で解く問題は，分析者が事前に設定した OLAP クエリを探索候補となる全ての部分データへ適用し，それらのクエリ結果の中から例外度が高い上位 n 件の部分データを特定する問題である．各部分データの例外度は，LOF により算出された LOF 値である．

3.1 問題定義

本研究で解く問題は，リレーショナルデータベースを対象とする．以降，全体データを D，全体データ D の部分集合を部分データ S と呼ぶ．全体データ D はレコードの集合であり，各レコードはメジャー属性（売上金額，売上個数など）の集合とディメンション属性（商品カテゴリ，地域など）の集合から構成される．部分データ S は，全レコードの中で条件 C を満たすレコードの集合であり，次式で定義される．

$$S := \sigma_C(D) \tag{5}$$

但し，σ はリレーショナル代数における選択演算である．また，C は選択演算時のレコードの選択条件（例えば，”商品名”=商品 A）であり，論理積（AND）を伴う複数の条件の指定が可能である．本研究で解く問題では，単一のメジャー属性 m と単一のグループ化属性 g から成る OLAP クエリ q と，部分データの集合 $\mathbb{S}$ を事前に設定する．説明の簡略化のため，メジャー属性 m がとり得る値は全て，0 以上の実数と仮定する．部分データ集合 $\mathbb{S}$ は，選択条件を満たす部分データの集合である．OLAP クエリ q は次式で定義される．

$$q := gG_{f(m)} \tag{6}$$

但し，g は全体データ D のディメンション属性，f はメジャー属性 m に対する集約関数を表す．集約関数 f は件数計算（COUNT），平均値計算（AVG），総和計算（SUM）などがある．$gG_{f(m)}$ はレコード集合をディメンション属性 g でグループ化しグループ毎に集約関数 f をメジャー属性 m に適用する処理である．クエリ結果 $q(S)$ は，グループ化の値と集約値を組としたシーケンス型であり，次式で表現できる．

$$q(S) = [(V_1, W_1), (V_2, W_2), \cdots, (V_N, W_N)] \tag{7}$$

但し，V_i（$1 \leqq i \leqq N$）はグループ化対象のディメンション属性 g が持つ各属性値，W_i（$1 \leqq i \leqq N$）は集約値の各値である．N は取り得るグループ化属性値の数を表す．

本研究で解く問題では，部分データ S 毎に，クエリ結果 $q(S)$ に含まれる N 個の集約値（$W_1, W_2, \cdots, W_N$）を N 次元ユークリッド空間上の 1 つの点の座標に変換し，各点の LOF 値を算出する．そして，各点の LOF 値をその点の元となる部分データの例外度とする．クエリ結果 $q(S)$ を座標に変換するため，クエリ結果 $q(S)$ に含まれる N 個の集約値（$W_1, W_2, \cdots, W_N$）から成るシーケンス $q'(S)$ を次式で定義する．

$$q'(S) := [W_1, W_2, \cdots, W_N] \tag{8}$$

また，LOF 計算時，全部分データの $q'(S)$ を同じスケールで扱うため，各 $q'(S)$ を正規化する．$q'(S)$ の正規化後のシーケンス $P[q'(S)]$ を次式で定義する．

$$P[q'(S)] := [W_1/x, W_2/x, \cdots, W_N/x] \tag{9}$$

但し，$x = \sum_{l=1}^{N} W_l$ である．つまり，$P[q'(S)]$ は，集約値の合計 x に対する各集約値 W_i（$1 \leqq i \leqq N$）の割合のシーケンスであり，各値は 0 以上 1 以下の値をとる．この P が，部分データ S のクエリ結果 $q(S)$ から変換された座標を表し，式（1）の A に相当する．部分データ S のクエリ結果 $q(S)$ を元にした座標 $P[q'(S)]$ の LOF 値が，その部分データ S の例外度を定量化する関数 U となる．関数 U は次式で定義される．

$$U(S) := LOF_k(P[q'(S)]) \tag{10}$$

但し，k は分析者により事前に設定された，近傍の範囲を表す LOF 計算時のパラメータである．

以上を踏まえ，本研究で解く問題を以下のように定義する．

定義 1　部分データ集合 $\mathbb{S}$，$OLAP$ クエリ q，LOF 計算における近傍の範囲 k，結果として得たい局所例外部分データ件数 n を指定し，部分データ集合 $\mathbb{S}$ に属する全ての S の中で $U(S)$ が最も大きい上位 n 件の S を特定する．

3.2　自動探索手法の処理フロー

本節では提案する自動探索手法の処理フローについて説明する．本研究の自動探索手法は以下の 4 つのステップで構成されている．以降，部分データ集合 $\mathbb{S}$ に属する部分データを $S_1, S_2, \cdots, S_{|\mathbb{S}|}$（$|\mathbb{S}|$ は部分データ集合 $\mathbb{S}$ に属する部分データの数）とする．

（1）　**クエリ結果の取得**

分析者はまず，部分データ集合 $\mathbb{S}$ を決定するため，部分データとして扱うディメンション属性（例えば，商品カテゴリや商品ブランド）を選び，選択条件 C を決定する．また，OLAP クエリ q を指定するため，集約関数 f，メジャー属性 m，グループ化属性 g を選択する．そして，リレーショナルデータベース D に対して OLAP クエリ q を実行し，全ての部分データ S_i（$1 \leqq i \leqq |\mathbb{S}|$）のクエリ結果を取得する．

（2）　**クエリ結果の座標化**

取得した各クエリ結果 $q(S_1), q(S_2), \cdots, q(S_{|\mathbb{S}|})$ から導出された $q'(S_1), q'(S_2), \cdots, q'(S_{|\mathbb{S}|})$ を正規化し，集約値の割合のシーケンスを求める．

（3）　**LOF の計算**

LOF の計算は，N 次元ユークリッド空間上の点に対して行う．本手法における次元数は，OLAP クエリ q に用いたグループ化属性 g が取り得るユニークな属性値の数である．例えば，グループ化属性 g が「売上月」で g が取り得る属性値が「1 月」，「2 月」,$\cdots$,「12 月」の 12 個である場合，ユークリッド空間の次元数は 12 となる．このステップでは，前のステップで求めた $P[q'(S_1)], P[q'(S_2)], \cdots, P[q'(S_{|\mathbb{S}|})]$ 1 つ 1 つを N 次元ユークリッド空間上の 1 つの点として考え，全ての点に対して LOF 値を計算する．

（4）　**局所例外部分データの特定**

各部分データの例外度（$U(S_1), U(S_2), \cdots, U(S_{|\mathbb{S}|})$）に基づき，例外度が大きい上位 n 件の部分データを特定する．

図 2 に，グループ化属性 g が取り得るユニークな属性値の数が 3 である部分データにおいて，LOF 値が大きい上位 20 件の部分データの点を示す．この図から，LOF 値が大きい点は，点全体の平均的な座標位置から離れている点ではなく，自身の局所到達可能密度が近傍の局所到達可能密度と比べて低い点であることが分かる．

4.　高速化手法

本節では，3.2 節の自動探索手法の処理フローのステップ (3)「LOF の計算」において，LOF 計算全体を高速化する手法について説明する．LOF 計算では，距離空間において各点に対しその点の k 近傍を特定する必要があり，これが計算量 $O(M^2)$（M はデータサイズ）という高コスト化の原因になっている．つまり，探索対象の部分データの数が多くなるほど例外度の計

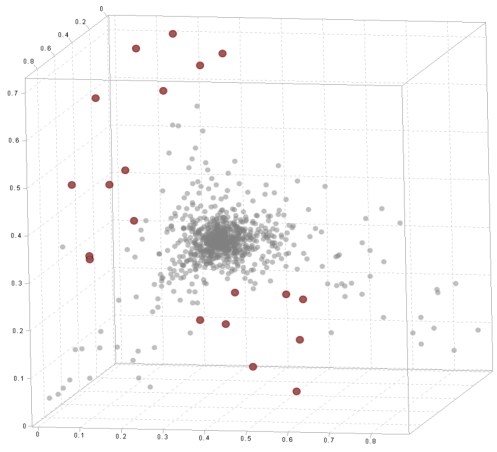

図 2　LOF 値が大きい上位 20 件の部分データ（3 次元ユークリッド空間）

算に長い時間を要するということであり，これは各点の k 近傍の探索を効率的に行うことで低コスト化が可能である．また，本研究では LOF 値上位 n 件の部分データの特定が目的であるため，上位 n 件の候補に入りえない部分データは自動探索処理の途中で足切りを行って正確な LOF 計算を省くことが望ましい．よって本高速化手法では，**効率的な k 近傍探索技術と部分データの足切り技術**の 2 つの技術により，全体の LOF 計算の高速化を行う．以降の節で，2 つの技術の詳細と高速化手法全体の処理フローについて説明する．

4.1　効率的な k 近傍探索

k 近傍の探索範囲の削減により，効率的な k 近傍探索を行う．本 k 近傍探索技術は，クラスタリングにおける計算量が $O(M)$（M はデータサイズ）であるグリッドベースクラスタリング [19] を用いる．まず，LOF 計算対象となる全ての点が分布する距離空間全体を格子状に区切ったより小さな空間（セル）へ分割する．そして各点の k 近傍探索時，k 近傍の探索範囲を全セルの中のいくつかのセルに絞ることで k 近傍の探索範囲を削減する．探索範囲のセルを決定するため，以下の定理を使う．

定理 1　ある点 A が属するセルを $Cell_A$ とし，$Cell_A$ の位置を次元毎の分割位置の組み合わせ $[x_1, x_2, \cdots, x_N]$（N は次元数）で表す．$Cell_A$ が次の 2 つの条件を満たす場合，A の k 近傍 $N_k(A)$ は，$Cell_A$ 又は $Cell_A$ の隣接セル内に必ず含まれる．

(i)　$Cell_A$ は $k+1$ 個以上の点を持つ

(ii)　$Cell_A$ 内の点における A 以外の全ての点に関して，A までの距離がセルの 1 辺の長さより小さい

証明 1　A のセルまたは隣接セル内に $N_k(A)$ が無いと仮定する．すると，$N_k(A)$ は A の隣接セルより A からみて更に遠い位置にあるセルに存在し，A からこの $N_k(A)$ への距離はセルの 1 辺の長さより長い．しかし，条件 (i)，(ii) が成り立つため，A からの距離がセルの 1 辺の長さより短い点は $Cell_A$ 内に k 個存在しており，仮定に反する．

本高速化手法では，定理 1 を満たす点の k 近傍探索範囲を，

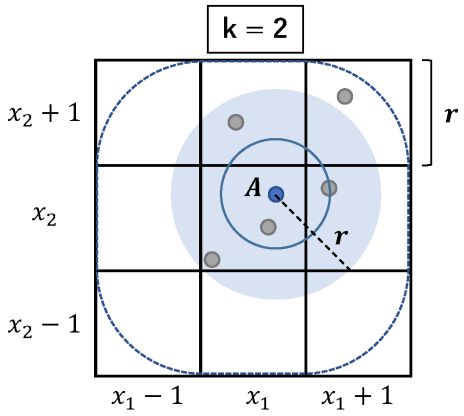

図 3　点 A の k 近傍探索範囲のセル

その点が「所属するセル」と「所属するセルの隣接セル」内の点に絞ることで効率的な k 近傍探索を実現する．図 3 に，ある点 A の 2 次元ユークリッド空間における k 近傍探索範囲のセルを示す．図によると，A の k 近傍探索範囲のセルは，A が属するセル $[x_1, x_2]$ を含む $[x_1-1, x_2-1]$，$[x_1-1, x_2]$，$[x_1-1, x_2+1]$，$[x_1, x_2-1]$，$[x_1, x_2]$，$[x_1, x_2+1]$，$[x_1+1, x_2-1]$，$[x_1+1, x_2]$，$[x_1+1, x_2+1]$ の 9 個のセルである．

Algorithm 1 に，効率的な k 近傍探索可能点を特定する部分のアルゴリズムを示す．Algorithm 1 は，グリッド分割後の全セルのリストである $CellList$，セルの 1 辺の長さである $CellSize$ を入力として，効率的な k 近傍探索可能点のリストである $PointList$ を出力する．各セルに対して，定理 1 の条件 (i) が成立するか確認し（2 行目），そのセルが持つ各点に対して定理 1 の条件 (ii) が成立するか確認し（6 行目から 9 行目），2 つの条件を満たす点を $PointList$ に追加する（12 行目）．

Algorithm 1 効率的な k 近傍探索可能点の特定

Input $CellList, CellSize$
Output $PointList$

```
1:  for each Cell ∈ CellList do
2:    if Cell > k then
3:      for each Point ∈ Cell.Points do
4:        for each OtherPoint ∈ Cell.Points − Point do
5:          Flug = True
6:          if Distance(Point, OtherPoint) > CellSize then
7:            Flug = False
8:            Break
9:          end if
10:       end for
11:       if Flug == True then
12:         PointList ← Point
13:       end if
14:     end for
15:   end if
16: end for
```

4.2　部分データの足切り

LOF 値の上限・下限の推定により，LOF 値上位 n 件に入りえない部分データの LOF 値の正確な計算を省く．ある点 A

の LOF 値の計算には，A の到達可能密度 $lrd_k(A)$ と，A の k 近傍 $N_k(A)$ 内の各点 B の到達可能密度 $lrd_k(B)$ が必要である．$lrd_k(A)$ の計算には，A から各 B への到達可能距離の計算で必要な $k\text{-}distance(B)$ が必要であり，そのためには各 B の $N_k(B)$ の特定が必要である．同様に $lrd_k(B)$ の計算には，B から各 C への到達可能距離の計算で必要な $k\text{-}distance(C)$ が必要であり，そのためには各 C の $N_k(C)$ の特定が必要である．つまり LOF は，A1 つにつき $|N_k(A)| \times |N_k(B)|$ の計算量がかかる $N_k(C)$ の特定が必要であり，コストが大きい．よって本高速化手法では，4.1 節で計算したセルの情報を利用し，$k\text{-}distance(C)$ の上限・下限の推定により A の LOF 値の上限・下限を推定し，LOF 値上位 n 件に入りえない A の足切りを行うことで，$N_k(C)$ の特定必要回数を減らすという高速化のアプローチをとる．

A の LOF 値 $LOF_k(A)$ の上限・下限は式（2）より，次式で定義できる．

$$LOF_k(A) \geqq \frac{\sum_{B \in N_k(A)} lrd_k(B).lower}{|N_k(A)|} / lrd_k(A)$$
$$LOF_k(A) \leqq \frac{\sum_{B \in N_k(A)} lrd_k(B).upper}{|N_k(A)|} / lrd_k(A) \qquad (11)$$

但し，$lrd_k(B).upper$ と $lrd_k(B).lower$ はそれぞれ，B の局所到達可能密度 $lrd_k(B)$ の上限と下限を表す．B の局所到達可能密度 $lrd_k(B)$ の上限・下限は式（3）より，次式で定義できる．

$$lrd_k(B) \geqq 1/(\frac{\sum_{C \in N_k(B)} reach\text{-}dist_k(B,C).upper}{|N_k(B)|})$$
$$lrd_k(B) \leqq 1/(\frac{\sum_{C \in N_k(B)} reach\text{-}dist_k(B,C).lower}{|N_k(B)|}) \quad (12)$$

但し，$reach\text{-}dist_k(B,C).upper$ と $reach\text{-}dist_k(B,C).lower$ はそれぞれ，B の到達可能距離 $reach\text{-}dist_k(B,C)$ の上限と下限を表す．式（12）はさらに式（4）により，次式で定義できる．

$$lrd_k(B) \geqq 1/(\frac{\sum_{C \in N_k(B)} \max\{d(B,C), k\text{-}distance(C).upper\}}{|N_k(B)|})$$
$$lrd_k(B) \leqq 1/(\frac{\sum_{C \in N_k(B)} \max\{d(B,C), k\text{-}distance(C).lower\}}{|N_k(B)|}) \qquad (13)$$

但し，$k\text{-}distance(C).upper$ と $k\text{-}distance(C).lower$ はそれぞれ，C の k 距離 $k\text{-}distance(C)$ の上限・下限を表す．

次の節で，C の k 距離 $k\text{-}distance(C)$ の上限・下限の計算方法を説明する．

4.2.1 $k\text{-}distance(C)$ の上限・下限推定

C の k 距離 $k\text{-}distance(C)$ とは，C から k 番目に近い点（以後，C_k とする）から C までの距離である．本高速化手法では，C_k を正確に特定せず，距離空間において C_k が存在する範囲を特定することで $k\text{-}distance(C)$ の上限・下限を計算する．C_k が存在する範囲の特定には，C が属するセル $Cell_C$ と $Cell_C$ の周囲のセルの情報を利用する．セルの情報とは，セルの重心点，重心点からセル内で最も遠い点までの距離（以後，セル内部点半径と呼ぶ）の 2 つである．C_k が存在する範囲と $k\text{-}distance(C)$ の上限・下限は，セルの情報を利用した次の定理により決定される．

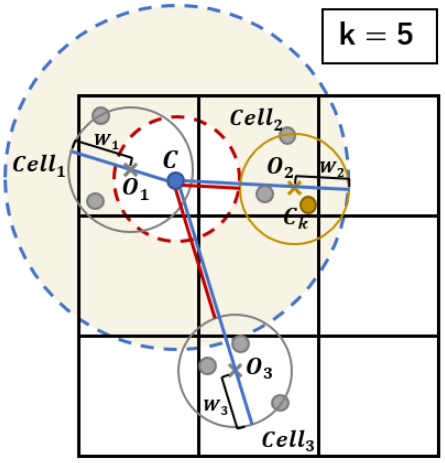

図 4　C_k の存在範囲と $k\text{-}distance(C)$ の上限・下限

定理 2　C の k 近傍 $N_k(C)$ が存在する可能性のあるセルを $Cell_1, Cell_2, \cdots, Cell_J$ とする．またそれらのセルについて，重心点をそれぞれ $O_1, O_2, \cdots, O_J$，セル内部点半径をそれぞれ $R_1, R_2 \cdots, R_J$，内包する点の数をそれぞれ $n_1, n_2, \cdots, n_J$ とする．この時次の 2 つの定理が成り立つ．

上限　$(d(C,O_1)+R_1), (d(C,O_2)+R_2), \cdots, (d(C,O_J)+R_J)$ を昇順ソートしたとき，$k\text{-}distance(C)$ の上限は $(d(C,O_i)+R_i)$ である．但し i は，$n_1, n_2, \cdots, n_i \geqq k$ かつ $n_1, n_2, \cdots, n_{i-1} < k$ を満たす．

下限　$(d(C,O_1)-R_1), (d(C,O_2)-R_2), \cdots, (d(C,O_J)-R_J)$ を昇順ソートしたとき，C_k は C から半径 $(d(C,O_i)-R_i)$ 以内の範囲には位置せず，$k\text{-}distance(C)$ の下限は $(d(C,O_i)-R_i)$ である．但し i は，$n_1, n_2, \cdots, n_i \geqq k$ かつ $n_1, n_2, \cdots, n_{i-1} < k$ を満たす．

図 4 に，C_k が存在する範囲と $k\text{-}distance(C)$ の上限・下限を示す．$Nk(C)$ が存在する可能性のあるセルが $Cell_1$，$Cell_2$，$Cell_3$ であり，重心点がそれぞれ O_1, O_2, O_3，セル内部点半径がそれぞれ R_1, R_2, R_3，内包する点の数がそれぞれ n_1, n_2, n_3 である．定理 2 を適用すると，図のように $k\text{-}distance(C)$ の上限は $d(C,O_2)+R2$ となり，下限は $d(C,O_2)-R2$ となる．

4.3　高速化手法の処理フロー

高速化手法全体の処理フローは以下の通りである．

（1）　**グリッド分割**

距離空間全体を小さなセルに分割し，全点に対して，その点が所属するセルを特定する．具体的には，まずセル 1 辺の長さを分析者が指定する．この情報と分析対象データの存在範囲により，1 次元あたりのセル分割数 x が自動的に決まる．これにより，距離空間全体を x^N（N は次元数）個のセルの空間とみなすことができる．よって各点につき，その点の座標を元に所属するセルを特定する．また，以降のステップで使用するため，点を 1 つ以上持つ全てのセルに対して，**重心の座標とセル内部点半径**の計算を行う．

（2）　**近傍探索**

効率的な近傍探索が行える点を特定し，特定した点の近傍を特

定する．具体的には，まず定理 1 の条件 (i) を満たすセルが持つ点において，定理 1 の条件 (ii) を満たす点を特定する．そして，特定した点につき，隣接セル内の全ての点までの距離を計算し，近傍を特定する．なお，隣接セル数は $3^N - 1$ である．

（3） **LOF 値の上限・下限計算**

近傍内の点までの距離の上限・下限の推定が行える点を特定し，それらの点の LOF 値の上限/下限を計算する．具体的には，まず前のステップで特定した点につき，その点の近傍内の全ての点が，前のステップで特定した他の点である点を特定する．そして，特定した点につき，lrd，近傍の lrd の上限・下限を計算し，LOF の上限・下限を計算する．

（4） **初期上位 n 件 LOF 値の計算**

次の足切りステップの前処理として，LOF 値上位 n 件に入るかどうかの判定に必要な閾値を求める．最終的に上位 n 件に入るポテンシャルのある点 n 件を選択し，それらの LOF 値を正確に計算し，その中で最も低い LOF 値を閾値とする．ポテンシャルのある点の選択であるが，LOF の性質（LOF 値はその点が孤立しているほど大きい）を踏まえ，内包する点の数が少ないセルに属する点ほどポテンシャルが高いと仮定し，そこから選択する．

（5） **足切り**

LOF 値の上限・下限を推定した点に対して 2 種類の足切りを行う．1 つ目の足切りでは，点の中で，その点の LOF 値の上限値が前のステップで計算した LOF の閾値を下回る点を足切りする．2 つ目の足切りでは，1 つ目の足切りで足切りされていない点を下限の高い順に並べその n 番目の下限を新たな閾値にし，上位 n 番目に入りえない点を足切りする．

（6） **全ての点の LOF 値の計算**

このステップでは，全ての点の正確な LOF 値を計算する．LOF 上位 n 件の閾値導出のために計算した初期 n 件の点と足切りステップで足切り対象であった点以外の全ての点の LOF を計算する．

5. 評価実験

本章では，3.2 節のステップ 3「LOF の計算」において，本自動探索手法に高速化技術を用いる場合が高速化技術を用いない場合に対してどの程度高速化が有効であるのか検証した実験について述べる．使用したデータセットは，経営科学系研究部会連合協議会主催 平成 28 年度データ解析コンペティションで提供された，ファッション EC サイトの販売データである．実験では，本来の LOF の高コスト化の要因であった「LOF 計算対象の部分データの総数」を変化させた場合の，「実行時間」，「足切りに成功した部分データ数」，「効率的な近傍探索に成功した部分データ数」への影響を調査する．実験では，本自動探索手法の事前入力パラメータを，部分データ集合としてディメンション属性が「商品番号」の部分データ（つまり，商品 1 つ 1 つが部分データ），OLAP クエリの集約関数を「SUM」，メジャー属性を「注文金額」，グループ化属性を「性別」に設定する．また，LOF 計算時の近傍の範囲 k を 20，結果として得たい部分データ件数 n を 30 に設定する．実験に使用した PC は，

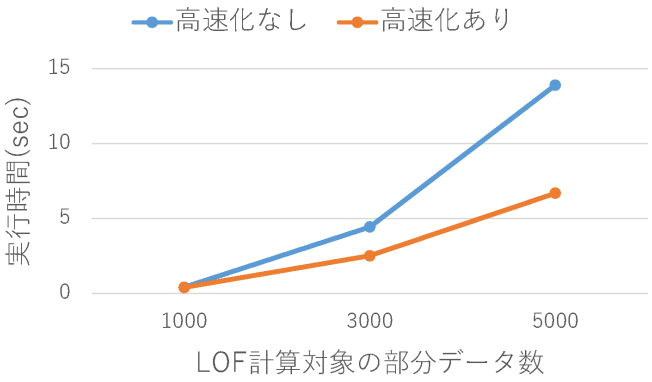

図 5　部分データ数変更時の LOF 計算にかかる実行時間

CPU が Intel(R) Core(TM) i5-6600CPU，クロック周波数が 3.30GHz，コア数が 4，メモリは 16GB の性能を持つ．LOF 計算対象の「商品番号」部分データの総数は，取得するクエリ結果を絞り，1000 個，3000 個，5000 個の 3 つに設定して実験を行う．

図 5 は，高速化技術を用いる場合と用いない場合における，LOF 計算対象部分データの総数変更時に LOF 計算にかかる総実行時間を示す．このグラフによると，LOF 計算対象の部分データ数が増加するほど LOF 計算にかかる実行時間が減少し，高速化技術を用いた場合は用いない場合に対して最大で約 2 倍の速度で計算が行えることがわかる．これは，LOF 計算対象部分データの総数が増加するほど効率的な近傍探索成功件数，部分データの足切りの成功件数が増えるためである．また，部分データ数が増加するほど実行時間の差が大きくなる傾向があることから，部分データの量が多くなるほど大幅に性能改善する．図 6 は，高速化技術を用いる場合における，LOF 計算対象部分データの総数変更時の「効率的な近傍探索に成功した部分データ数」と「足切りに成功した部分データ数」の変動を表す．このグラフによると，LOF 計算対象部分データの総数が増加するほど効率的な近傍探索成功件数，部分データの足切りの成功件数が増加している．これは，LOF 計算対象部分データの総数が増加するほど 1 セルあたりの点の数が増え，定理 1 を満たす点が増えるためである．定理 1 を満たす点が増えることで，効率的な近傍探索成功件数が増え，それにより部分データの足切りの成功件数も増える．これら 2 つのグラフから，LOF 計算対象部分データの総数が増加するほど，足切り・効率的な近傍探索に成功する部分データが増加し，結果的に LOF 計算に要する実行時間は減少していくことがわかる．

6. 分析結果

本章では，実際のデータセットに対し本自動探索手法を適用して得られた分析結果の例と，分析結果の実用的な活用方法について説明する．使用したデータセットは，5. 章の評価実験で使用したデータセットと同じファッション EC サイトの販売データである．本自動探索の事前入力パラメータのうち LOF 計算時の近傍の範囲 k を 20，部分データ集合としてディメンション属性「商品カテゴリ小」の部分データ 226 件を用いて得

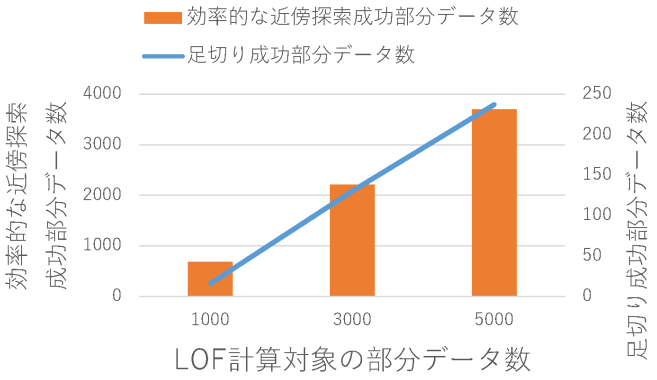

図 6　足切り成功・効率的な近傍探索に成功した部分データ数

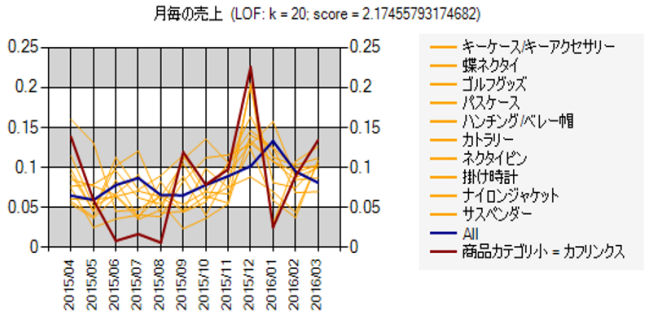

図 7　LOF 値上位部分データ例「カフリンクス」

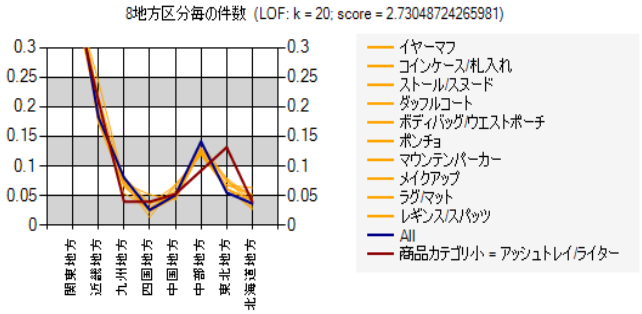

図 8　LOF 値上位部分データ例「アッシュトレイ/ライター」

られた分析結果のうち，LOF 値が上位の部分データの例を図 7 と図 8 に示す．これらのグラフには，特定した部分データを赤色の線，特定した部分データの近傍の部分データ 10 件を橙色の線，全部分データの平均「All」を青線で示している．

図 7 は，集約関数を「SUM」，メジャー属性を「注文金額」，グループ化属性を「注文月」という観点での分析で得られた，部分データ「カフリンクス」に関する分析結果であり，グラフは月毎の売上金額を表す．このグラフによるとカフリンクスの近傍にはサスペンダーやネクタイピン等の紳士服関連のアクセサリーが含まれているが，カフリンクスだけ 6 月から 8 月の売上が著しく低いという他と比べて例外的な傾向を示す．この原因としては気温が高い夏場はワイシャツをあまり着用しないという点挙げられるが，夏季でも特に暑い 7 月の売上が 6 月と 8 月より大きいことから夏場の需要も少なくないと判断できる．よって，清涼感のある薄手のワイシャツと共にユーザーに推薦を行う等の方法により夏場の売上を伸ばせる可能性がある．

図 8 は，集約関数を「$COUNT$」，メジャー属性を「オーダー数」，グループ化属性を「8 地方区分」という観点での分析で得られた，部分データ「アッシュトレイ/ライター」に関する分析結果であり，グラフは月毎の売上件数を表す．このグラフからは，アッシュトレイ/ライターは東北地方での売上件数が他の地方と比べて唯一多いという気付きが得られる．この原因として都道府県別の喫煙率は北日本ほど高い（厚生労働省「国民生活基礎調査の概況」より）という点が挙げられる．よって，ユーザーに対して地域別の商品推薦を行いたい場合等に，居住地が北日本のユーザーにはアッシュトレイ/ライターを推薦することにより増益が狙える．

7. 関 連 研 究

OLAP クエリ結果の可視化が可能なデータ解析ツールには，TIMBER [8]，Spotfire [9]，Polaris [10] などがある．TIMBER はデータ構造に基づいてユーザーを補助する可視化ツールである．Spotfire は散布図をベースとした可視化システムである．Polaris は基本的なデータベースクエリとテーブル代数による可視化の仕様を統合したシステムである．Spotfire と Polaris はデータセットに最適な可視化設定を自動的に選択するが，分析者が設定することも可能である．これらの可視化ツールは分析者が着目したい全ての属性を手動で選択する必要がある．

OLAP クエリ結果を自動で可視化する機能を持つデータ解析ツールには，Profiler [11]，Vizdeck [12] などがある．Profiler はデータの異常を自動で検出し，いくつかの可視化結果を表示する．Vizdeck はダッシュボード上に 2 次元で表示し得る全ての可視化結果を表示する．

複数のソースからデータを収集・統合・可視化という一連の処理を自動化する技術として Google Fusion Tables [13]，DEVise [14] などがある．Google Fusion Tables は，web 上から様々なデータを収集し，統合することによりテーブルを作成し，その分析結果を可視化する．

有用な分析結果の獲得を目指し分析工程を自動化する技術の研究として MuVE [4] がある．MuVE は，SEEDB が非対応であった数的な次元でのグループ化処理を可能にするため複数の有用性関数を用いて有用性の定量化を行い，ある部分データにおいて有用性の大きい OLAP クエリを特定する．この研究に加えて SEEDB [2,3] や水野らの研究 [5,6] は，分析工程自動化のアプローチは異なるが，大域的な有用性の評価関数を用いている点で類似している．

将来的な OLAP クエリ結果可視化ツールである Ermacs [15] は，データベースの最適化機能とデータベースの可視化機能を統合するための新しい宣言的な可視化言語を有している．また，データの事前読み込みを用いた多次元データキューブ分析に関する研究 [16,17] では，OLAP キューブから特異的な単一のセルを探索する手法を提案している．

本研究では，[5,6] と同様に例外的な部分データの自動探索を行うが，局所的な有用性の評価方法を用いている点が特徴的である．

LOF 関連の研究として，LOF 値上位 n 件を効率的に探索す

る Jin らの手法 [18] がある．この手法では，事前処理として距離空間上の点に対しクラスタリングを行った後，クラスタ毎にクラスタに所属 LOF の上限・下限値を推定することで LOF 値上位 n 件に入りえない点を含むクラスタの足切りを行う．初めの点をグルーピングする部分の計算量は，クラスタリングの場合は $O(M^2)$ であるが，本研究はセルによる方法であるため計算量が $O(M)$ である点で異なる．

8. おわりに

本研究では，局所例外部分データを自動探索する手法と LOF 計算の高速化手法を提案した．また，計算対象部分データの総数に対する本高速化手法の効果を調査し，LOF 計算対象部分データの総数が増加するにつれ，部分データの足切りや効率的な近傍探索に効果があり，LOF 計算における実行時間の短縮にも効果あることを確認した．さらに，実際のデータに本自動探索手法を適用すると，実用的な利用方法を持つ分析結果が得られることが判明した．今後は，LOF 計算高速化手法の他の関連研究と比べた性能評価，本手法で特定した例外部分データの有用性に関する議論を行う予定である．

謝　辞

本研究は JSPS 科研費 JP16K00154 の助成を受けたものです．

文　献

[1] Jiawei Han, Jian Pei and Micheline Kamber, “Data Mining: Concepts and Techniques,” Elsevier, 2011.

[2] Manasi Vartak, Samuel Madden, Aditya Parameswaran and Neoklis Polyzotis, “SeeDB: Automatically Generating Query Visualizations,” Proc. VLDB Endow., Vol. 7, No. 13, pp. 1581–1584, 2014.

[3] Manasi Vartak, Sajjadur Rahman, Samuel Madden, Aditya Parameswaran and Neoklis Polyzotis, “SEEDB: efficient data-driven visualization recommendations to support visual analytics,” Proc. VLDB Endow., Vol. 8, No. 13, pp. 2182–2193, 2015.

[4] Humaira Ehsan, Mohamed A. Sharaf and Panos K. Chrysanthis, “MuVE: Efficient Multi-Objective View Recommendation for Visual Data Exploration,” 32nd IEEE International Conference on Data Engineering, pp. 1–12, 2016.

[5] 水野陽平, 鬼塚真, “統計的信頼区間を用いた特徴的な部分データの効率的探索,” データ工学と情報マネジメントに関するフォーラム (DEIM), D3–4, 2016.

[6] Yohei Mizuno, Yuya Sasaki and Makoto Onizuka, “Efficient Data Slice Search for Exceptional View Detection,” DOLAP ’17, 2017.

[7] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander, “LOF: identifying density-based local outliers,” ACM sigmod record, Vol. 20, No. 2, pp. 93–104, 2000.

[8] Michael Stonebraker and Joseph Kalash, “TIMBER: A sophisticated relation browser,” Proceedings of the 8th International Conference on Very Large Data Bases, pp. 1–10, 1982.

[9] Christopher Ahlberg, “Spotfire: An Information Exploration Environment,” SIGMOD Rec., Vol. 25, No. 4, pp. 25–29, 1996.

[10] Chris Stolte, Diane Tang and Pat Hanrahan, “Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases,” Commun. ACM, Vol. 51, No. 11, pp. 75–84, 2008.

[11] Sean Kandel, Ravi Parikh, Andreas Paepcke, Joseph M. Hellerstein and Jeffrey Heer, “Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment,” Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 547–554, 2012.

[12] Alicia Key, Bill Howe, Daniel Perry and Cecilia Aragon, “VizDeck: Self-organizing Dashboards for Visual Analytics,” Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 681–684, 2012.

[13] Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen and Jonathan Goldberg-Kidon, “Google Fusion Tables: Web-centered Data Management and Collaboration,” Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 1061–1066, 2010.

[14] M. Livny, R. Ramakrishnan, K. Beyer, G. Chen, D. Donjerkovic, S. Lawande, J. Myllymaki and K. Wenger, “DEVise: Integrated Querying and Visual Exploration of Large Datasets,” SIGMOD Rec., Vol. 26, No. 2, pp. 301–312, 1997.

[15] Eugene Wu, Leilani Battle and Samuel R. Madden, “The Case for Data Visualization Management Systems: Vision Paper,” Proc. VLDB Endow., Vol. 7, No. 10, pp. 903–906, 2014.

[16] Sunita Sarawagi, Rakesh Agrawal and Nimrod Megiddo, “Discovery-driven exploration of OLAP data cubes,” EDBT’98, pp. 168–182, 1998.

[17] Sunita Sarawagi, “User-Adaptive Exploration of Multidimensional Data,” VLDB, pp. 307–316, 2000.

[18] Wen Jin, Anthony KH Tung, Jiawei Han, “Mining top-n local outliers in large databases,” Proceedings of the seventh ACM SIGKDD, pp. 293–298, 2001.

[19] Wei Wang, Jiong Yang, and Richard Muntz, “STING: A statistical information grid approach to spatial data mining,” VLDB, Vol. 97, pp. 186–195, 1997.