統計的信頼区間を用いた局所例外部分データの効率的探索アルゴリズム

小笠原麻斗† 松本 拓海†† 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
†† 大阪大学工学部 〒 565–0871 大阪府吹田市山田丘 1-5
E-mail: †{ogasawara.asato,matsumoto.takumi,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし ビジネスデータの分析手法は，データの大規模化・多様化に伴い，従来の分析者が手動で分析を行う OLAP ベースの手法から，システム側が自動で分析し分析者に有用な分析結果を推薦する探索的データ解析手法に移行している．特に，販売データ等の地域性や時期性の局所的な偏りが大きいデータの分析においては，局所的な偏りを捉え，その偏りの中で例外的な傾向を示す部分データ（局所例外部分データ）を探索することが重要である．本稿では，局所例外部分データを効率的に探索するアルゴリズムを提案する．提案手法では，信頼区間推定技術に基づいて各部分データの例外度の上限・下限を推定し，例外度上位 n 件に入り得ない部分データを探索処理の途中で足切りすることにより，不要なデータ読み込み量を削減することで効率的に局所例外部分データを特定する．評価実験の結果，提案手法は，既存の局所例外部分データ探索手法の探索時間を最大 84% 削減することに成功し，更にデータサイズに対するスケーラビリティを有していることを確認した．

キーワード 探索的データ解析，外れ値検知，信頼区間

1. はじめに

ビッグデータ時代の到来に伴い，大規模化・多様化したビジネスデータを分析して有用な情報を抽出する需要が高まっている．ビジネスデータの分析には，データ可視化ツールを使用した OLAP（Online analytical processing）ベースの分析手法が用いられてきた．OLAP ベースの分析手法は，(1) 分析者が分析したいデータや可視化方法を選択し，(2) それらに基づいて出力されるデータの可視化結果を見て考察を行う，という 2 つのステップを繰り返し行う手法である．分析者は，出力した可視化結果の要因の調査によって有用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 ベースの分析手法では，有用な可視化結果を得られるまで上記の 2 つのステップを繰り返す必要があるため，特にデータが大規模・多様である場合，分析者の負担が大きいという問題がある．

上記の問題を解決するため，様々な探索的データ解析（Exploratory data analysis）手法が開発されている [1–6]．これらの手法は，例外的なクエリ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クエリ結果を特定し，そのクエリ結果を分析者に推薦する．但し，分析観点とは OLAP クエリにおける集約属性や group-by 属性を指し，部分データとは分析対象のデータ全体から関係代数における選択演算によって抽出された部分的なデータ（例えば，「商品カテゴリ＝ “T シャツ”」の条件で抽出されたタプルの集合）を指す．クエリ結果の有用性は，クエリ結果の例外度を算出することで評価し，例外度が高いほど有用性が高いと定義される．特に水野らの [5, 6] の手法では，ユーザが指定した OLAP クエリ q において，任意の部分データ t のクエリ結果 $q(t)$ の例外度は，比較対象のデータ c（しばしば，データ全体）に q を適用して得られたクエリ結果 $q(c)$ との乖離度として定義される．すなわち，この例外度は，部分データの傾向が比較対象のデータの傾向からどの程度乖離があるかという大域的な捉え方で算出される．一方我々は先行研究 [7] において，部分データの傾向がその部分データが所属するクラスタの傾向からどの程度乖離があるかという，局所的な傾向を考慮して各クエリ結果の例外度を評価する手法を提案した（以降，クラスタからの乖離度として計算されるこの例外度を局所例外度と呼び，高い局所例外度を持つクエリ結果を生成する元となった部分データを局所例外部分データと呼ぶ）．局所例外度は，特に販売データ等の地域性や時期性の偏りが大きいデータの分析において有用な例外度である．[7] の手法は，局所外れ値検知技術の主流な技術である LOF [8] を使用して，各部分データのクエリ結果の局所例外度（LOF 値）を計算し，自動的に局所例外部分データを探索する．しかし，[7] では，クエリ処理に対する効率化は行われていないため，データサイズとクエリ発行回数が増加する程データ読み込み量も増加し，探索時間が膨大になるという問題がある．

本稿では，クエリ処理に対する実行時間を効率化するため，LOF 値上位 n 件の局所例外部分データを効率的に探索するアルゴリズムを提案する．提案手法では，LOF 値上位 n 件に入り得ない部分データを探索処理の途中で足切りすることにより，不要なデータ読み込み量を削減することで高速化を図る．更に，部分データの足切りのため，信頼区間推定技術 [9] に基づいて各部分データのクエリ結果の信頼区間を推定し，その後各部分データの LOF 値の上限・下限を推定する．評価実験の結果，提案手法は，既存の局所例外部分データ探索手法の探索時間を最大 84% 削減することに成功し，更にデータサイズに対するスケーラビリティを有していることを確認した．

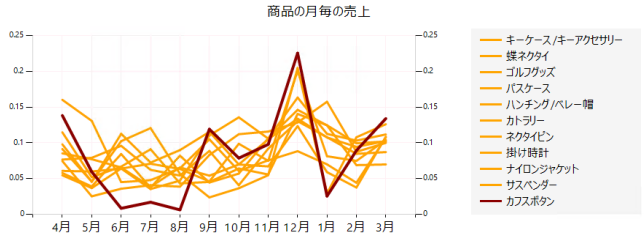

図 1　局所例外部分データ例

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説明し，3 章で提案手法の詳細について説明する．4 章で提案手法の評価を行い，5 章で関連研究について述べる．6 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

本章では，先行研究 [7] に基づき，局所例外部分データと局所外れ値検知技術 LOF（local outlier factor）[8]，局所例外部分データの自動探索フレームワークについて説明する．

2.1 局所例外部分データの具体例

局所例外部分データの重要性を，[7] の手法を実データに適用して得られた分析結果（図 1）を用いて説明する．この分析では，分析観点は月毎の売上であり，部分データ数は 210 であった．図 1 の赤色の線は，特定した局所例外部分データ “カフスボタン” の月毎の売上を表す．橙色の線は，部分データ 210 件の中で “カフスボタン” と売上傾向が最も類似した部分データ 10 件の月毎の売上を表す．つまり，図に表示されている部分データ 11 件は，全部分データの中で局所的な傾向を共有する部分データである．“カフスボタン” はこの部分データ 11 件の中において局所例外度が高く算出された部分データである．ここで，図から “カフスボタン” は “サスペンダー” や “ネクタイピン” 等の紳士服関連アクセサリーのクラスタに属していることが分かる．しかし，このクラスタ内の部分データの中で “カフスボタン” は唯一 6 月〜8 月の夏期の売上が低いという例外的な傾向を示す．よって，この要因を調査して販売戦略を打ち立てることにより，“カフスボタン” の売れ行きを改善させられる可能性がある．このように，部分データの中で，類似する傾向を持つ部分データ（この例では，売上好調期が似ている商品）を局所的にみてその中で例外的な傾向を持つ部分データ（局所例外部分データ）を特定することは重要である．

2.2 LOF (local outlier factor)

LOF は，密度に基づく最近傍ベースの代表的な外れ値検知手法である [8]．LOF では，各データ（例えば，部分データ 1 つ 1 つ）を N 次元距離空間上の 1 つの点とみなし LOF 値を計算する．任意の点 A は，0 以上の実数 a_i（$1 \leqq i \leqq N$）(注i) の N 個の組で表される座標として，次式で定義される．

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

LOF では，距離空間における局所的な範囲を表現するため，分析者が事前設定するパラメータ k を用いて，点毎に近傍（以後，k 近傍と呼ぶ）を特定する．A の k 近傍 $N_k(A)$ は，次式で定義される．

$$N_k(A) := \{\ B \in \mathbb{P} - \{A\} \mid d(A,B) \leqq k\text{-}distance(A)\ \} \tag{2}$$

但し，$\mathbb{P}$ は距離空間上の点の集合，$d(A,B)$ は A と B の間のユークリッド距離，$k\text{-}distance(A)$ は A から k 番目に近い点と A との間のユークリッド距離を表す．すなわち，$N_k(A)$ は，A から $k\text{-}distance(A)$ 以下の距離に位置する点を含む集合である．A の LOF 値は，$LOF_k(A)$ として次式で定義される．

$$LOF_k(A) := \frac{\sum_{B \in N_k(A)} lrd_k(B) \ /\ |N_k(A)|}{lrd_k(A)} \tag{3}$$

$LOF_k(A)$ は，A の k 近傍内の点 B との密度（lrd_k）の比で表される．$lrd_k(A)$ は次式で定義される．

$$lrd_k(A) := \frac{|N_k(A)|}{\sum_{B \in N_k(A)} reach\text{-}dist_k(A,B)} \tag{4}$$

但し，$reach\text{-}dist_k(A,B)$ は，B から A への到達可能距離（reachability distance）を表す．直感的には，B から A への到達可能距離は，距離空間において B と A がどの程度離れているかを表す．$reach\text{-}dist_k(A,B)$ は次式で定義される．

$$reach\text{-}dist_k(A,B) := \max\{d(A,B),\ k\text{-}distance(B)\} \tag{5}$$

但し，式 (5) において，$k\text{-}distance(B)$ は $lrd_k(A)$ が無限大になる事を防ぐための項である．k の値は 10 以上に設定すると統計的変動を小さく抑えることができる [8]．

2.3 局所例外部分データの自動探索フレームワーク

自動探索フレームワークで解く問題は，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局所例外度が高い上位 n 件の部分データを特定する問題である．各部分データの局所例外度は，LOF により算出された LOF 値である．

関係モデルに基づき，自動探索フレームワークで解く問題を定義する．分析対象のデータを D，D の部分集合を部分データ S とする．D はタプルの集合であり，各タプルはメジャー属性（売上金額等）の集合とディメンション属性（商品カテゴリ等）の集合で構成される．S は，全タプルの中から任意の条件（例えば,「商品名=“商品 A”」）で選択されるタプルの集合を表す．また，グループ化・集約処理を行う OLAP クエリを q とし，D 内の部分データ S に対して q を適用して得られるクエリ結果を $q(S,D)$ と表す．$q(S,D)$ は，N をグループ化属性が取り得る値（以後，グループ値と呼ぶ）の数とした時，グループ値と集約値の組を N 個持つ (注ii)．この問題では，S 毎に，$q(S,D)$ に含まれる N 個の集約値 $(W_1, W_2, \cdots, W_N)$ を N 次元距離空間上の 1 つの点の座標とみなし，各点の LOF 値を計算する．$q(S,D)$ の集約値から成るシーケンス $q'(S,D)$ は次式で定義さ

(注i)：負の値がある場合は，値域を正規化して全て正数にする．

(注ii)：クエリ結果においてグループ値と集約値の組の数が N 個より少ない場合（つまり，集約対象のタプルが存在しないグループ値が 1 つ以上ある場合），そのグループ値の集約値を 0 とすることで，組の数が N 個になるようにする．

れる．

$$q'(S, D) := [W_1, W_2, \cdots, W_N] \tag{6}$$

この $q'(S, D)$ が，式 (1) の A に相当する．

以上を踏まえ，このフレームワークで解く問題は以下のように定義される．

［定義 1］ 分析対象のデータを D，部分データ集合 $\mathbb{S}$，OLAP クエリ q，LOF 値計算における近傍の範囲 k，特定する局所例外部分データの件数 n を与え，部分データ集合 $\mathbb{S}$ に属する全ての S の中で $LOF_k(q'(S, D))$ 上位 n 件の S を特定する．

$$\underset{S \in \mathbb{S}}{\operatorname{argmax}^n} \ LOF_k(q'(S, D)) \tag{7}$$

3. 提 案 手 法

自動探索フレームワーク [7] では，データサイズが膨大になるほどデータ読み込みに要するクエリ処理時間がかかるという問題がある．そこで本稿では，LOF 値上位 n 件に入りえない部分データを探索処理の途中で足切りすることにより，不要なデータ読み込み量を削減することで効率的に LOF 値上位 n 件の部分データを特定する手法を提案する．

3.1 探索処理の概要

本節では，提案手法の探索処理の概要を説明する．提案手法では，統計的信頼区間推定の技術を用いることで，全てのデータを読み込む前に各部分データの LOF 値の上限・下限を判断し，最終的に LOF 値上位 n 件に入り得ない部分データを足切りする．具体的には，信頼区間推定の技術を適用するため，まず分析対象のデータをサンプルとそれ以外のデータに分割し，サンプルに対し OLAP クエリを実行する．その後，サンプルに対する OLAP クエリで得られた各部分データのクエリ結果に対して信頼区間を推定し，LOF 値の上限・下限を推定する．そして，導出した LOF 値の上限・下限を用い，LOF 値上位 n 件に入り得ない部分データを足切りする．

Algorithm 1 を用いて，提案手法の処理フローを説明する．Algorithm 1 は，入力として全体データ D，部分データ集合 $\mathbb{S}$，OLAP クエリ q，近傍の範囲 k，特定する件数 n，D に対するサンプルの割合 x を与え，LOF 値上位 n 件の部分データ $dataSlices$ を返却する．処理フローは以下の通りである．

（1） **サンプルへの OLAP クエリの実行**

データセット D のタプル集合の上から x%をサンプルのタプル集合（$samples$）とし，D を $samples$ と $(D - samples)$ に分割する（1 行目）．OLAP クエリ q を用い，部分データ集合 $\mathbb{S}$ 内の各部分データ S に対するクエリ結果を $samples$ から取得する（2 行目）．この時，信頼区間推定のための統計情報も取得する（詳細は 3.2 節）．各部分データのクエリ結果（$q(S, samples)$）は，$results$ として保持する．

（2） **信頼区間の推定**（詳細は 3.2 節）

全部分データのクエリ結果と統計情報から，各部分データのクエリ結果の信頼区間を推定する（3〜7 行目）．信頼区間は，クエリ結果のグループ値の数ぶんある集約値毎に推定する．

Algorithm 1 提案手法の処理フロー

Input D, $\mathbb{S}$, q, k, n, x
Output $dataSlices$

```
1: samples ← separateData(D, x)
2: results ← groupByAggregate(q, 𝕊, samples)
3: for each dataSlice ∈ results do
4:     for dimension ← 0 to |dataSlice| do //集約値毎に信頼区間を推定
5:         computeInterval(dataSlice, dimension)
6:     end for
7: end for
8: allLOFBounds ← computeAllLOFBounds(k)
9: keptDataSlices ← pruning(allLOFBounds, n)
10: results ← results ⊕ groupByAggregate(q, keptDataSlices, |D − samples|)
11: for each dataSlice ∈ keptDataSlices do
12:     LOFList[dataSlice] ← computeLOF(dataSlice, k)
13: end for
14: dataSlices ← getTopN(LOFList, n)
```

（3） **LOF 値の上限・下限の導出**（詳細は 3.3 節）

クエリ結果と推定した信頼区間を用いて，全部分データの LOF 値の信頼区間（上限・下限）を導出する（8 行目）．各部分データの LOF 値の上限・下限には，全体データ D に対するクエリ結果（$q(S, D)$）から計算される LOF 値が，指定した信頼係数に基づく誤差の範囲で含まれる．

（4） **部分データの足切り**（詳細は 3.4 節）

導出した LOF 値の上限・下限を用いて，LOF 値上位 n 件の候補になり得ない部分データの足切りを行い，残りのデータに対するクエリ処理をスキップする．足切り出来なかった部分データの集合を $keptDataSlices$ として保持する（9 行目）．

（5） **残りのデータへの OLAP クエリの実行**

足切り出来なかった部分データは，全体データに対するクエリ結果から LOF 値の計算を行う必要がある．よって，サンプルに対するクエリ結果は既に取得しているため，サンプル以外の残りのデータに対して OLAP クエリを実行し，クエリ結果を差分で更新する（10 行目）．[10] より，クエリ結果は差分更新が可能であるので，返されたクエリ結果（$q(S, D - samples)$）をステップ 1 で返されたクエリ結果（$q(S, samples)$）と結合（$\oplus$）する．但し，$\oplus$ は，与えられた部分データのクエリ結果の集約値を，対応する部分データのクエリ結果の集約値に加算する処理であり，[11] における加算処理に相当する．この OLAP クエリの適用対象のデータは，サンプル以外のデータ $(D - Sample)$ において，$keptDataSlices$ 内の部分データに該当する部分であり，足切りされた部分データに該当する部分はスキップされる．このステップでクエリが適用されるデータ量が減る程，自動探索フレームワークの実行時間の削減に繋がる．

（6） **LOF 値の計算**

$keptDataSlices$ 内の部分データ毎に LOF 値を正確に計算する（11〜13 行目）．その後，LOF 値上位 n 件の部分データを特定する（14 行目）．

以降の節で，提案手法で重要なステップであるステップ 2（信頼区間の推定），3（LOF 値の上限・下限の導出），4（部分データの足切り）で用いる技術を具体的に説明する．

3.2 信頼区間の推定

サンプルに対して OLAP クエリを実行した後，各部分データのクエリ結果の各集約値に対して信頼区間を推定する．各集約値に信頼区間を伴うクエリ結果は，各集約値の信頼区間を距離空間の各次元における存在領域とみなすことにより，距離空間上では超立方体として表現できる．クエリ結果の信頼区間を推定した後は，超直方体間の距離の上限・下限に基づき，LOF 値の上限・下限を導出する（3.3 節）．提案手法では，信頼区間推定の技術として，標本に関する情報のみで母数の値の範囲の推定が可能である中心極限定理に基づく技術を使用する [9]．提案手法においては，サンプルが標本に対応し，全体データが母数に対応する．この技術で推定される信頼区間は，信頼係数 $p \in (0,1)$ に基づく誤差の範囲内で母数の値を含む．信頼区間の幅は以下のように定義される．

$$\varepsilon_m := \left(\frac{Z_p^2 T_m}{m}\right)^{1/2} \tag{8}$$

但し，m はサンプル数，Z_p は信頼度に関する係数，T_m は集約値の不偏分散を表す．Z_p は，正規累積分布関数における $(p+1)/2$ 分位数であり，p の値と正の相関を持つ．部分データ S の母集団におけるクエリ結果 $q(S,D)$ の任意の集約値は，その部分データの標本におけるクエリ結果 $q(S,samples)$ の任意の集約値を $\bar{X}$ とすると，p に基づく誤差の範囲内で $[\bar{X}-\varepsilon_m, \bar{X}+\varepsilon_m]$ の範囲に含まれる．部分データ S に関して，標本において信頼区間を伴う OLAP クエリ結果 $q'(S,samples)$ は以下のように表現される．

$$\begin{aligned} q'(S,samples) = [&[\bar{X}_1-\varepsilon_{m_1}, \bar{X}_1+\varepsilon_{m_1}], \\ &\cdots, [\bar{X}_N-\varepsilon_{m_N}, \bar{X}_N+\varepsilon_{m_N}]]. \end{aligned} \tag{9}$$

但し，$\bar{X}_i$ $(1 \leqq i \leqq N)$ は各集約値を表し，ε_{m_i} $(1 \leqq i \leqq N)$ は各集約値に対する信頼区間の幅を表す．よって，各部分データのクエリ結果の各集約値の信頼区間を導出するために，Algorithm 1 の 2 行目において取得する統計情報は，各部分データのクエリ結果の各集約値に関する m と T_m である．ここで，$q'(S,samples)$ は，N 次元距離空間において，$(\bar{X}_i-\varepsilon_{m_i})$ 以上 $(\bar{X}_i+\varepsilon_{m_i})$ 以下（$1 \leqq i \leqq N$）の範囲を i 次元における存在領域とみなすことにより超立方体として表現できる．図 2 に，2 次元距離空間における $q'(S,samples)$ の存在領域を示す．図のように，2 次元距離空間における $q'(S,samples)$ は，1 次元目における変域が $(\bar{X}_1-\varepsilon_{m_1})$ 以上 $(\bar{X}_1+\varepsilon_{m_1})$ 以下，2 次元目における変域が $(\bar{X}_2-\varepsilon_{m_2})$ 以上 $(\bar{X}_2+\varepsilon_{m_2})$ 以下，中心の座標が $(\bar{X}_1, \bar{X}_2)$ である 2 次元超直方体として表現できる．これに基づき，LOF 値の上限・下限を導出するため，超直方体間の距離の上限・下限を導出する（3.3.1 項）．

3.3 LOF 上限・下限推定

各部分データのクエリ結果の各集約値に対して信頼区間を導出した後，各部分データの LOF 値の上限・下限を導出する．

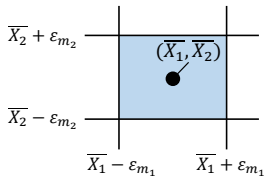

図 2　2 次元距離空間における $q'(S,samples)$ の存在領域

LOF 値の計算には，部分データのクエリ結果間の距離を導出することと，部分データの k 近傍を特定することが必要である．よって LOF 値の上限・下限を推定するためには，(1) 信頼区間を持つクエリ結果間の距離の上限・下限を導出することと，(2) クエリ結果が信頼区間を伴う状態において，各部分データの k 近傍候補を特定することが必要である．

初めに，LOF [8] をベースとした LOF 値の上限・下限の導出式を定義する．式 (3) より，点 A の LOF 値は，分母に A の密度 $lrd_k(A)$ を取り，分子に A の k 近傍内の各点 B の密度 $lrd_k(B)$ の平均を取る．そこで，式 (3) を拡張し，A の LOF 値の上限を，分母に $lrd_k(A)$ の下限を取り，分子に $lrd_k(B)$ の上限の平均を取るように定義する．また，クエリ結果が信頼区間を伴う状態において，各部分データの k 近傍は正確に特定できない．そこで，$lrd_k(B)$ の上限の平均を計算する際は，A の k 近傍候補内の各 B に対して $lrd_k(B)$ の上限を求め，その中から $lrd_k(B)$ の上限上位 k 件を選択しその平均を計算する．これにより，信頼区間の信頼度に基づく誤差の範囲で $LOF_k(A)$ の上限を定めることができる．A の LOF 値の下限の導出式は，同様の方法で定義する．式 (3) に基づき，A の LOF 値の上限・下限（$LOF_k(A).upper$，$LOF_k(A).lower$）を次式で定義する．

$$\begin{aligned} LOF_k(A).upper &:= \frac{\sum_{B \in L_k^{max}(A)} lrd_k(B).upper/k}{lrd_k(A).lower} \\ LOF_k(A).lower &:= \frac{\sum_{B \in L_k^{min}(A)} lrd_k(B).lower/k}{lrd_k(A).upper} \\ L_k^{max}(A) &:= \underset{B \in N_k'(A)}{\overset{k}{\operatorname{argmax}}}\ lrd_k(B).upper \\ L_k^{min}(A) &:= \underset{B \in N_k'(A)}{\overset{k}{\operatorname{argmin}}}\ lrd_k(B).lower \end{aligned} \tag{10}$$

但し，$N_k'(A)$ は A の k 近傍候補を表し，後述する定理 1 により規定される（詳細は 3.3.2 項）．$L_k^{max}(A)$ は，A の k 近傍候補内の B において，$lrd_k(B).upper$ 上位 n 件の B の集合を表し，$L_k^{min}(A)$ は，A の k 近傍候補内の B において，$lrd_k(B).lower$ 下位 n 件の B の集合を表す．$lrd_k(B).upper$ と $lrd_k(B).lower$ はそれぞれ $lrd_k(B)$ の上限と下限を表す．

次に，$lrd_k(B)$ の上限・下限の導出式を定義する．式 (4) より，$lrd_k(B)$ は，B の k 近傍内の各点 C から B への到達可能距離（$reach\text{-}dist_k(B,C)$）の平均の逆数を取る．そこで，式 (4) を拡張し，$lrd_k(B)$ の上限を，$reach\text{-}dist_k(B,C)$ の下限の平均の逆数を取るように定義する．また，クエリ結果が信頼区間を伴う状態においては，B の k 近傍候補を用いて $lrd_k(B)$ の上限を導出する必要がある．そこで，$reach\text{-}dist_k(B,C)$ の下限の平均を計算する際は，B の k 近傍候補内の各 C に対して $reach\text{-}dist_k(B,C)$

の下限を求め，その中から $reach\text{-}dist_k(B,C)$ の下限下位 k 件を選択しその平均を計算する．これにより，信頼区間の信頼度に基づく誤差の範囲で $lrd_k(B)$ の上限を定めることができる．$lrd_k(B)$ の下限の導出式は，同様の方法で定義する．式 (4) に基づき，$lrd_k(B).upper$ と $lrd_k(B).lower$ を次式で定義する．

$$
\begin{aligned}
lrd_k(B).upper &:= \frac{k}{\sum_{C\in N_k^{min}(B)} reach\text{-}dist_k(B,C).lower} \\
lrd_k(B).lower &:= \frac{k}{\sum_{C\in N_k^{max}(B)} reach\text{-}dist_k(B,C).upper} \\
N_k^{min}(B) &:= \underset{C\in N_k'(B)}{\overset{k}{\operatorname{argmin}}}\ reach\text{-}dist_k(B,C).lower \\
N_k^{max}(B) &:= \underset{C\in N_k'(B)}{\overset{k}{\operatorname{argmax}}}\ reach\text{-}dist_k(B,C).upper
\end{aligned}
\tag{11}
$$

但し，$reach\text{-}dist_k(B,C).upper$ と $reach\text{-}dist_k(B,C).lower$ はそれぞれ $reach\text{-}dist_k(B,C)$ の上限と下限を表す．$N_k^{max}(B)$ は，B の k 近傍候補内の各 C において，$reach\text{-}dist_k(B,C).upper$ 上位 n 件の C の集合を表し，$N_k^{min}(B)$ は，B の k 近傍候補内の各 C において，$reach\text{-}dist_k(B,C).lower$ 下位 n 件の C の集合を表す．

最後に，$reach\text{-}dist_k(B,C)$ の上限・下限の導出式を定義する．式 (5) より，$reach\text{-}dist_k(B,C)$ は，$d(B,C)$ と $k\text{-}distance(C)$ の内大きい方の値を取る．そこで，式 (5) を拡張し，$reach\text{-}dist_k(B,C)$ の上限を，$d(B,C)$ の上限と $k\text{-}distance(C)$ の上限の内大きい方の値を取るように定義する．これにより，信頼区間の信頼度に基づく誤差の範囲で $reach\text{-}dist_k(B,C)$ の上限を定めることができる．$reach\text{-}dist_k(B,C)$ の下限の導出式は，同様の方法で定義する．式 (5) に基づき，$reach\text{-}dist_k(B,C).upper$ と $reach\text{-}dist_k(B,C).lower$ を次式で定義する．

$$
\begin{aligned}
reach\text{-}dist_k(B,C).upper &:= \max\{d(B,C).upper, k\text{-}distance(C).upper\} \\
reach\text{-}dist_k(B,C).lower &:= \max\{d(B,C).lower, k\text{-}distance(C).lower\}
\end{aligned}
\tag{12}
$$

但し，$d(B,C).upper$ と $d(B,C).lower$ はそれぞれ $d(B,C)$ の上限と下限を表し，$k\text{-}distance(C).upper$ と $k\text{-}distance(C).lower$ はそれぞれ $k\text{-}distance(C)$ の上限と下限を表す．$k\text{-}distance(C).upper$ は，C 以外の各点 D に対する $d(C,D).upper$ の中で，下位 k 番目の $d(C,D).upper$ の値を取る．同様に $k\text{-}distance(C).lower$ は，C 以外の各 D に対する $d(C,D).lower$ の中で，下位 k 番目の $d(C,D).lower$ の値を取る．

以降の項で，$d(B,C).upper/lower$ 及び $N_k'(A)$ の導出方法を述べる．

3.3.1 $d(A,B)$ の上限・下限の導出

3.2 節で述べたように，部分データのクエリ結果の各集約値が信頼区間を伴う場合，その部分データのクエリ結果は距離空間上の超立方体として表現できる．つまり，信頼区間を伴う任意の 2 つの部分データのクエリ結果間の距離の上限・下限は，距離空間における 2 つの超立方体の間の最大・最小距離に相当する．2 つの部分データ A と B のクエリ結果間の距離の上限・下限（$d(A,B).upper$，$d(A,B).lower$）は，N次元空間における 2 つの物体の距離関数に基づき規定する．但し，A と B の距離空間上における存在領域に重なりがある場合は $d(A,B).lower = 0$ とする．図 3 に，2 次元距離空間上における，各集約値が信頼区間を伴う 2 つの部分データ A と B のクエリ結果間の距離の上限・下限（$d(A,B).upper/lower$）の値の取り方を示す．$d(A,B).upper$ は，A の超直方体の任意の頂点と B の超直方体の任意の頂点の間の距離の中で，最大となる距離を取り，同様に $d(A,B).upper$ は，最小となる距離を取る．但し，図のように 2 つの超直方体の存在領域が重なりがある場合は，$d(A,B).upper = 0$ となる．

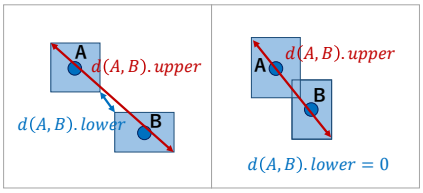

図 3　2 次元距離空間における $d(A,B).upper/lower$

3.3.2 k 近傍候補探索

LOF 値の計算には，各点 A につき k 近傍 $N_k(A)$ に含まれる点の特定が必要である．式 (9) より，サンプルにおける各部分データのクエリ結果 $q'(S, samples)$ は信頼区間を伴うため，k 近傍の候補は次の定理によって特定される．

［定理 1］　距離空間上の点の集合 $\mathbb{P}$ 内の点 A の k 近傍候補 $N_k'(A)$ は以下のように特定される．

$$
N_k'(A) := \{\ B \in \mathbb{P} - \{A\}\ \mid\ d(A,B).lower \leqq k\text{-}distance(A).upper\ \}
\tag{13}
$$

［証明 1］　A の k 近傍 $N_k(A)$ が k 近傍候補 $N_k'(A)$ に含まれることを説明する．式 (2) より，$N_k(A)$ には $d(A,B)$ が $k\text{-}distance(A)$ 以下の点 B が含まれる．ここで，信頼区間の信頼度に基づく誤差の範囲において，$d(A,B).lower \leqq d(A,B)$ であり，$k\text{-}distance(A) \leqq k\text{-}distance(A).upper$ であるため，$N_k(A) \subseteqq N_k'(A)$ が成り立つ．

図 4 に，$k = 2$ の時の 2 次元距離空間におけるある点 A の k 近傍候補（$N_2'(A)$）を示す．この図の距離空間において，B_i（$1 \leqq i \leqq 4$）が $N_2'(A)$ に含まれるかどうかの距離の閾値である $2\text{-}distance(A).upper$ は，各 B_i の位置関係から，$d(A,B_2).upper$ が選択される．よって $N_2'(A)$ には，$d(A,B_i).lower$ が $2\text{-}distance(A).upper$ 以下となる B_i，すなわち B_1，B_2，B_3 が含まれる．

3.3.3 LOF 値の上限・下限の導出

本項では，3.3.1 項で述べた $d(A,B)$ の上限・下限の導出技術や 3.3.2 項で述べた k 近傍候補の特定技術に基づいて，LOF 値の上限・下限を導出する．

LOF 値の上限・下限を導出するための前処理として，まず全点

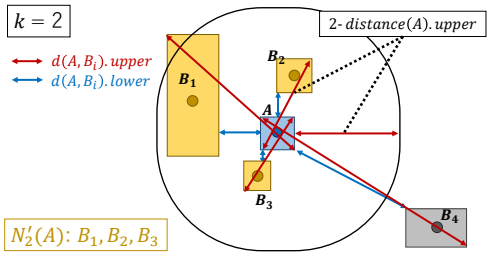

図 4　2 次元距離空間における A の k 近傍候補（$N_2'(A)$）

Algorithm 2 lrd_k の上限・下限の導出

Input A, $knnCandidates$
Output lrd_{upper}, lrd_{lower}

1: $lrd_{upper} \leftarrow 0$, $lrd_{lower} \leftarrow 0$, $i \leftarrow 0$
2: $reachDist_{uppers}, reachDist_{lowers}$
3: **for each** $B \in knnCandidates[A]$ **do**
4: $\quad reachDist_{lowers}[i] \leftarrow getReachDist_{lower}(A, B)$
5: $\quad reachDist_{uppers}[i] \leftarrow getReachDist_{upper}(A, B)$
6: $\quad i \leftarrow i + 1$
7: **end for**
8: $lrd_{upper} \leftarrow k$ / $getMinK(reachDist_{lowers}).Sum()$
9: $lrd_{lower} \leftarrow k$ / $getMaxK(reachDist_{uppers}).Sum()$

の k 近傍候補を特定し，その後全点の lrd_k の上限・下限を導出する．式 (11) に基づき，点 A の lrd_k の上限・下限を導出するアルゴリズムを Algorithm 2 に示す．Algorithm 2 では，入力として A，定理 1 で特定された k 近傍候補リスト（$knnCandidates$）を与える．式 (12) に基づき，$knnCandidates$ に含まれる A の k 近傍候補内の点 B 毎に，B から A への到達可能距離の上限・下限を導出する（4〜5 行目）．次に，式 (11) 内で定義された N_k^{max} を取得する関数 $getMaxK(reachDist_{uppers})$ と，N_k^{min} を取得する関数 $getMinK(reachDist_{lowers})$ を用いて，$lrd_k(A)$ の上限・下限を導出する（8〜9 行目）．

全点の lrd_k の上限・下限を導出した後，全点の LOF 値の上限・下限を導出する．式 (10) に基づき，点 A の LOF 値の上限・下限を導出するアルゴリズムを Algorithm 3 に示す．Algorithm 3 では，入力として A と lrd_k の上限・下限のリスト（lrd_{uppers}，lrd_{lowers}），k 近傍候補リスト（$knnCandidates$）を与える．まず，$knnCandidates$ に含まれる A の k 近傍候補内の点 B 毎に，$lrd_k(B)$ の上限・下限を取得し集約する（5〜6 行目）．次に，式 (10) 内で定義された L_k^{max} を取得する関数 $getMaxK(lrdB_{uppers})$ と，L_k^{min} を取得する関数 $getMinK(lrdB_{lowers})$ を用いて $lrd_k(B)$ の上限・下限の平均値を導出し（9〜10 行目），最終的に $LOF_k(A)$ の上限・下限を導出する（11〜12 行目）．

3.4　部分データの足切り

全ての部分データの LOF 値の上限・下限を導出した後，LOF 値上位 n 件に入り得ない部分データの足切りを行う．提案手法の処理フローにおいて，ステップ 4（**部分データの足切り**）の次のステップ 5（**残りのデータへの OLAP クエリの実行**）においては，サンプル以外のデータからタプルを取得する部分データは，足切り出来なかった部分データである．よってステップ

Algorithm 3 LOF 値の上限・下限の導出

Input A, lrd_{uppers}, lrd_{lowers}, $knnCandidates$
Output LOF_{upper}, LOF_{lower}

1: $lrdA_{upper} \leftarrow lrd_{uppers}[A]$, $lrdA_{lower} \leftarrow lrd_{lowers}[A]$, $i \leftarrow 0$
2: $lrdB_{upper} \leftarrow 0$, $lrdB_{lower} \leftarrow 0$
3: $lrdB_{uppers}, lrdB_{lowers}$
4: **for each** $B \in knnCandidates[A]$ **do**
5: $\quad lrdB_{uppers}[i] \leftarrow lrd_{uppers}[B]$
6: $\quad lrdB_{lowers}[i] \leftarrow lrd_{lowers}[B]$
7: $\quad i \leftarrow i + 1$
8: **end for**
9: $lrdB_{upper} \leftarrow getMaxK(lrdB_{uppers}).Sum()$ / k
10: $lrdB_{lower} \leftarrow getMinK(lrdB_{lowers}).Sum()$ / k
11: $LOF_{upper} \leftarrow lrdB_{upper}$ / $lrdA_{lower}$
12: $LOF_{lower} \leftarrow lrdB_{lower}$ / $lrdA_{upper}$

4（**部分データの足切り**）では，足切り技術によって足切り出来ない部分データを特定し，その部分データに関するクエリ結果，LOF の上限・下限等の計算結果，部分データの選択条件を保持する．また，LOF 値の計算には，近傍の点の情報が必要である．よって，LOF 値の上限・下限に基づいてまず LOF 値上位 n 件に入る可能性のある部分データを特定し，その後その部分データの LOF 値を最終的に計算する際に必要な近傍の部分データを特定する．特定されたこれらの部分データ以外の部分データが，不要な部分データとして足切りされる．

LOF 値上位 n 件に入り得ない部分データを足切りする方法を Algorithm 4 を用いて説明する．Algorithm 4 では，入力として全部分データ集合（$points$），LOF 値の上限・下限リスト（LOF_{uppers}，LOF_{lowers}），k 近傍候補リスト（$knnCandidates$）を与え，出力として足切り出来なかった部分データ集合（$keptPoints$）を返却する．全部分データの LOF 値の下限の中で，上位から n 番目の下限値を足切りの閾値として設定する（1 行目）．部分データ毎に，LOF 値の上限がその閾値以上かどうか判定し（3 行目），閾値以上である全ての部分データ（A）を保持する（4 行目）．そして，A の LOF 値を最終的に計算する際に必要な近傍である，$N_k'(A)$，$N_k'(B)$，$N_k'(C)$ に含まれる部分データを特定する（5〜13 行目）．

3.5　高速化性能を促進するための実装上の工夫

この節では，提案手法の高速化性能を向上させるため行った実装上の工夫について述べる．LOF は，LOF 値を計算するための k 近傍探索における計算コストが高く（$O(n^2)$（n はデータサイズ）），部分データ数が増大するほど計算コストも増大する．特に本提案手法では，k 近傍探索を行う局面が多く（LOF 値の上限・下限の計算時や，正確な LOF 値の計算時），k 近傍探索にかかる計算コストが増大するほど探索処理時間が増大する．よって，空間インデックス技術 R-Tree [12] を区間木として用いることで，各局面において k 近傍探索を効率的に行う．

4.　評 価 実 験

提案手法の高速化性能を評価するため，既存手法と探索処理全体の実行時間を比較した．既存手法として，複数クエリの

Algorithm 4 部分データの足切り

Input $points$, LOF_{uppers}, LOF_{lowers}, $knnCandidates$
Output $keptPoints$

```
1: threshold ← getTopNth(LOF_lowers)
2: for each A ∈ points do
3:   if threshold ≦ LOF_uppers[A] then
4:     keptPoints.add(A)
5:     for each B ∈ knnCandidates[A] do
6:       if keptPoints.contains(B) == false then
7:         keptPoints.Add(B)
8:       end if
9:       for each C ∈ knnCandidates[B] do
10:        if keptPoints.contains(C) == false then
11:          keptPoints.Add(C)
12:        end if
13:        for each D ∈ knnCandidates[C] do
14:          if keptPoints.contains(D) == false then
15:            keptPoints.Add(D)
16:          end if
17:        end for
18:      end for
19:    end for
20:  end if
21: end for
```

共有化を行った局所例外部分データの自動探索フレームワーク [7]，自動探索フレームワークに LOF 値計算効率化アルゴリズム [13] を適用した手法を用いた．[13] は，クラスタリングを用いて上位 n 件 LOF 値のみを効率的に計算する主要なアルゴリズムである．また，様々なデータサイズで調査を行うことでデータサイズに対するスケーラビリティも調査した．

データセット: ヘアサロンチェーンの販売データである実データ (注iii) と，理想的な状態で本提案手法を評価するため人工データ (注iv) を使用した．このデータセットを 10 倍，30 倍，50 倍に複製することで，スケーラビリティの評価を行った．

OLAP クエリ: 単純な状況における本提案手法の性能を評価するため，実データでは，グループ値が 2 つである性別毎に，商品 1 つ 1 つの売上傾向を分析するクエリを使用した．具体的には，グループ化属性を**性別**，部分データ集合のためのディメンション属性を**商品 ID**，集約関数を SUM，集約属性を**会計税込売上**とした．また，人工データでは，実データと同様に 2 次元のクエリ結果を返却するクエリを使用した．

パラメータ: 分析者の入力パラメータに関して，LOF 値計算時の近傍の範囲 k は，[8] より lrd_k の統計的変動が抑えられる最低数である 10 とした．また，本提案手法の最高性能を評価するため，特定する部分データ件数 n は 1 に設定した．更に，サンプリングにおける信頼係数 p は 0.95（すなわち，Z_p は 1.96 となる），サンプル率 X は 10%とした．

(注iii)：経営科学系研究部会連合協議会主催平成 29 年度データ解析コンペティションで提供されたデータである．
(注iv)：一般に，商品の売上はロングテールと称される power-law 分布に従うため，部分データを構成するタプル数の分布が power-law 分布に従うように人工データを作成した [14]．

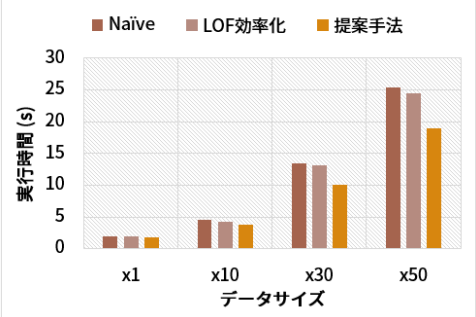

図 5　手法間の実行時間（実データ）

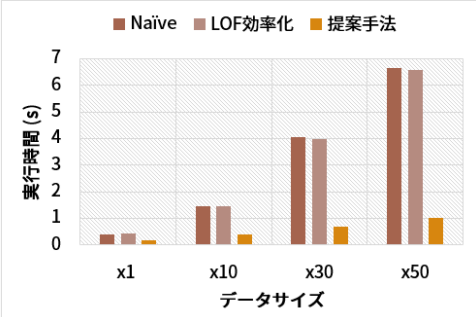

図 6　手法間の実行時間（人工データ）

図 5，6 は，データセット毎に，複数クエリの共有化を行った自動探索フレームワーク（**Naive**），自動探索フレームワークに LOF 値計算効率化アルゴリズム [13] を適用した手法（**LOF 効率化**），そして自動探索フレームワークに提案手法を適用した手法（**提案手法**）の 3 つの手法に関して，複製したデータサイズを変更して探索処理全体の実行時間を計測した結果である．図 5 の実データに対する実験結果において，**提案手法**では，どのデータサイズにおいても部分データの足切り率は 63%であり，部分データの足切りによって読み込まれなかったタプルの割合（タプルの足切り率）は 15%であった．図 5 では，**提案手法**は，**Naive** 手法と **LOF 効率化**手法に比べてデータサイズの増加に対し一貫して実行時間が短く，かつ最大で既存手法の実行時間を 22%削減できていることが分かる． 一方，図 6 の人工データに対する実験結果において，**提案手法**では，どのデータサイズにおいても部分データの足切り率は 80%であり，タプルの足切り率は 87%であった．図 6 では，**提案手法**は，図 5 と同様に **Naive** 手法と **LOF 効率化**手法に比べてデータサイズの増加に対し一貫して実行時間が短く，かつ最大で既存手法の実行時間を 84%削減できていることが分かる．また，精度に関して，図 5，6 双方において，各手法で特定した上位 n 件の結果は一致していた．実験より，提案手法は，クエリ処理の効率化による探索時間の高速化が可能であり，データサイズへのスケーラビリティも備えることが分かった．特に，タプル数の分布の偏りが異なる 2 種類のデータセットに対する結果から，部分データを構成するタプル数の偏りが比較的小さいと，足切りされるタプル数が増えるため，高速化の効果は高まりやすいことが分かった．更に，部分データを構成するタプル数が power-law 分

布に従うような，一般的な販売データに対して本提案手法は有効であることも判明した．

5. 関 連 研 究

本章では，本稿に関わる 3 つの研究領域について，各研究領域の主要な研究を概説することにより俯瞰する．

5.1 データ可視化ツール

Spotfire [15] は，散布図をベースに複数種類の可視化ビューを描画できる．Polaris [16] は，多次元データベースのインターフェースとして主流な Pivot Table を拡張しているシステムであり，多角的な視点からデータの特徴を捉える機能を備える．Tableau [17] は，Polaris をベースに開発された，商用のデータ可視化ツールであり，直感的なユーザインターフェースと多様な可視化機能を備える．これらの可視化ツールは，描画可能な可視化の多様性は高いが，使用するユーザは有用性の高い可視化結果が得られるまで試行錯誤の解析作業を行わなければならないため，ユーザの負担が大きい．対照的に，本稿で提案する効率的探索フレームワークは，ユーザにとって有用性の高い分析結果を効率的に探索する．

5.2 探索的データ解析技術

SEEDB [1,2] は，多次元データベースにおいて様々な OLAP クエリを探索し，全体データに OLAP クエリを適用したクエリ結果と部分データに適用したクエリ結果の乖離が最も大きくなる OLAP クエリを特定する．水野らの研究 [5,6] では，全体データに OLAP クエリを適用したクエリ結果と部分データに適用したクエリ結果の乖離を計算する事により例外的な部分データを探索する．Zenvisage [3,4] は，SEEDB の OLAP クエリ探索技術と水野らの部分データ探索技術をハイブリッドに組み合わせたシステムを開発している．これらの研究では，大域例外部分データの探索を効率的に行う手法が提案されているが，本稿では，局所例外部分データの探索を効率的に行う手法を提案する．

5.3 外れ値検知技術

Wu らの手法 [18] は，サンプリング技術を用いて最近傍探索工程を効率化する手法であり，その計算コストを $O(MN)$（M はサンプル数）に抑える．Jin らの手法 [13] は，マイクロクラスターの探索と各マイクロクラスターの LOF 値の上限・下限の計算を行い，LOF 値上位 n 件に入りえない点を含むマイクロクラスターを計算途中において足切りすることで，上位 n 件の異常値のみを効率的に探索する．本稿では，座標に幅を持つデータ点において LOF 値の上限・下限を推定し，LOF 値上位 n 件の点を特定する点が特徴的である．

6. お わ り に

本稿では，局所例外部分データの自動探索フレームワークにおいて，データサイズに対するスケーラビリティを向上させるアルゴリズムを提案した．提案手法は，例外度上位 n 件に入りえない部分データを探索処理の途中で足切りする技術により不要なデータ読み込み量を削減することで自動探索フレームワークを高速化する．評価実験の結果，提案手法は，既存の局所例外部分データ探索手法の探索時間を最大 84% 削減することに成功し，更にデータサイズに対するスケーラビリティを有していることも確認した．今後の課題として，1) 信頼係数と結果の有効性の関係性の調査や，2) サンプリング後の各部分データの信頼区間を伴うクエリ結果を超直方体として捉えた距離空間上での部分データの分布の仕方により，足切り率とそれによる高速化性能が大きく変動する問題への対処がある．2) に関して，足切り率が高くならない分布のケースは，(i) 存在領域（信頼区間）が重なり合う部分データの量が大きいケースや，(ii) 集約値に欠損値を持つ部分データが多く距離空間上の原点付近に密集するケース等が考えられる．

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] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Automatically generating query visualizations. *PVLDB*, Vol. 7, No. 13, pp. 1581–1584, 2014.
[2] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: efficient data-driven visualization recommendations to support visual analytics. *PVLDB*, Vol. 8, No. 13, pp. 2182–2193, 2015.
[3] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran. Effortless data exploration with zenvisage: An expressive and interactive visual analytics system. *PVLDB*, Vol. 10, No. 4, pp. 457–468, 2016.
[4] T. Siddiqui, J. Lee, A. Kim, E. Xue, C. Wang, Y. Zou, L. Guo, C. Liu, X. Yu, K. Karahalios, et al. Fast-forwarding to desired visualizations with zenvisage. In *CIDR*, 2017.
[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. In *DEIM*, 2016.
[6] Y. Mizuno, Y. Sasaki, and M. Onizuka. Efficient data slice search for exceptional view detection. In *DOLAP*, 2017.
[7] 小笠原麻斗, 水野陽平, 佐々木勇和, 鬼塚真. 局所例外部分データの自動探索. *DBSJ Journal*, Vol. 16, No. 13, 2018. to appear.
[8] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander. Lof: Identifying density-based local outliers. *SIGMOD Record*, Vol. 29, No. 2, pp. 93–104, 2000.
[9] P.J. Haas. Large-sample and deterministic confidence intervals for online aggregation. In *SSDBM*, pp. 51–63, 1997.
[10] A. Gupta and I.S. Mumick. Materialized views. chapter Maintenance of Materialized Views: Problems, Techniques, and Applications, pp. 145–157. 1999.
[11] J. A. Blakeley, N. Coburn, and P.A. Larson. Updating derived relations: Detecting irrelevant and autonomously computable updates. *TODS*, Vol. 14, No. 3, pp. 369–400, 1989.
[12] A. Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD Record*, Vol. 14, No. 2, pp. 47–57, June 1984.
[13] W. Jin, A.K.H. Tung, and J. Han. Mining top-n local outliers in large databases. In *KDD*, pp. 293–298, 2001.
[14] C. Anderson. *The long tail: Why the future of business is selling more for less.* Hyperion, 2006.
[15] C. Ahlberg. Spotfire: An information exploration environment. *SIGMOD Record*, Vol. 25, No. 4, pp. 25–29, 1996.
[16] C. Stolte, D. Tang, and P. Hanrahan. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *TVCG*, Vol. 8, No. 1, pp. 52–65, 2002.
[17] Tableau. `https://www.tableau.com`.
[18] M. Wu and C. Jermaine. Outlier detection by sampling with accuracy guarantees. In *KDD*, pp. 767–772, 2006.