大域的・局所的データ分析を両立した効率的なフレームワーク

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　〒 5650871 大阪府吹田市山田丘 1-5
E-mail: †{matsumoto.takumi,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし　探索的データ分析は，ビッグデータ時代における重要な研究の 1 つである．探索的データ分析の典型的なアプローチは，分析観点（グループ化属性や集約属性）や分析対象となるデータを変更することによって自動的に生成される多数の OLAP クエリの結果の中から興味深い結果を特定する．多くの分析観点や部分データが存在するため，分析に膨大な時間を要するという課題が存在する．既存研究では，色々な方法で高速化を行っているが，対象としている分析手法が大域的なデータか局所的なデータのどちらかにしか対応していない．本稿は，大域的なデータ分析と局所的なデータ分析を両立した探索的データ分析のフレームワークを提案する．また，中心極限定理を用いた top-k 枝刈りとデータキューブを用いたクエリ共有化を行うことによって処理の最適化を行う．更に，複数の実データに対して適用した結果を報告する．

キーワード　OLAP，探索的データ分析，Data Cube

1. はじめに

John Tukey は 1977 年にデータから隠れた知見を発見するために仮説を立てることなく分析する手法 [1] を提唱し，探索的データ分析 (EDA: Exploratory Data Analysis) [2] として知られている．一方，統計的な仮説検証型のデータ分析 (CDA: Confirmatory Data Analysis) は，ビックデータは 3V (Variety, Velocity, Volume) という特徴によって統計的な仮説を立てることが難しい．そのため，仮説検証型のデータ分析ではなく探索的データ分析 がビックデータに適しており，現在の研究のトレンドの 1 つである．探索的データ分析の主な目的は，統計的仮説を定義することなくデータを探索することで興味深い観測結果を得ることである．興味深い観測結果とは，データセット内に存在する平均的な動作を計算することや，例外的な点や異常傾向を特定することで計測される．

実際，OLAP クエリ結果から異常傾向を特定する探索的データ分析の研究が数多く存在する [3–10]．これらの研究は，データ分析の観点においてクエリ探索と部分データ探索の 2 つに大別することができる．分析対象のデータを D とすると，クエリ探索は，データ D の部分データ S を入力し，クエリ結果 $q(D)$ と $q(S)$ が最も乖離する OLAP クエリ q を特定する．一方，部分データ探索は，OLAP クエリ q を入力し，クエリ結果 $q(D)$ と $q(S)$ が最も乖離する部分データ S を特定する．ほとんどの既存手法 [4–6, 10] は，複数のクエリ結果間の距離を計算することで大域例外 (Global Outlier Factor) を特定することに効果的であるが，局所例外 (Local Outlier Factor) を特定することは出来ない．局所例外は多くのアプリケーション領域でよく知られている概念である．例えば，クレジットカードの異常な使用を検出する不正検出，顧客の異常行動を特定するカスタマイズマーケティング，あるいは様々な治療の異常な反応を発見する医用分析で用いられている [11]．小笠原らの研究 [12] では，部分データ探索において局所例外の特定を実現している

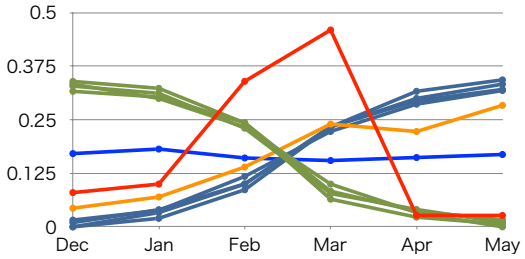

図 1　商品別正規化した月毎の売上．このグラフには 3 つのクラスターが存在するが，赤色の線が大域例外であり，黄色の線が局所例外のデータである．

が，大域例外を特定することはできない．以下の例では，大域・局所例外の外れ値検出技術が特定できる興味深い観測について説明する．

Example 1: 大域例外の異常傾向
図 1 において，緑色や青色の線は比較的同様の傾向を示す部分データが存在する．しかし，赤色の部分データでは，Mar, Apr, May 以外の月に関して他の部分データなどと比較して乖離が大きいため，x 軸全体を通して例外的なデータであると言える．

Example 2: 局所例外の異常傾向
図 1 において，橙色の線と青色の線の傾向は x 軸全体を通して似た傾向を示している．しかし，Apr に関して値が低く，局所的な傾向の中での異常値であり，局所例外度が高く算出される部分データであるため，局所的な例外データであると言える．

本研究は，大域例外と局所例外を両立している既存技術は存在していないという問題に対し，部分データ探索問題に対して大域的・局所的な 異常傾向の検出を両立した，探索的データ分析のための効率的なフレームワークを提案する．top-k 枝刈りとクエリ共有化による最適化を行うことで高速化を行う．クエリや部分データの候補は膨大に存在するため，入力データ

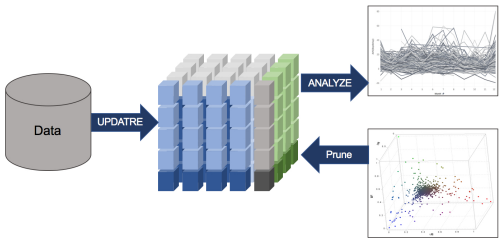

図 2　提案フレームワークの概要

をストリーム（レコード毎に）処理することでデータキューブと Join 処理し，データキューブを段階的に維持する効率的なアプローチを行う [9,13]．データキューブのセルが top-k の候補に含まれないと判定すると，データキューブから対象のセルを削除する．さらに，効率的な top-k 枝刈りのために Online Aggregation 技術 [14,15] を採用する．

図 2 は提案するフレームワークの概要を表す．入力データに対するストリーム処理に User Defined Aggregation Function (UDAF) を使用して，Spark の上に提案フレームワークを実装する．

本稿の構成は，以下の通りである．2. 章にて事前知識について説明し，3. 章において提案フレームワークの詳細を示し，4. 章にて評価実験について説明し，5. 章にて関連研究について述べ，6. 章にて本稿をまとめ，今後の課題を論ずる．

2. 事 前 知 識

本章では，提案フレームワークに用いた信頼区間と信頼区間の推定に必要である確率不等式，分析アプリケーションについて説明する．2.1 節では中心極限定理を用いた OLAP クエリ結果の区間推定について 2.2 節では LOF (Local Outlier Factor) について説明する．そして 2.3 節で分析アプリケーションに関する問題定義や有用性の定義を説明する．

2.1　中心極限定理を用いたクエリ結果の区間推定

信頼区間とは，母数の値がどのような範囲に存在するかを確率的に表す方法である．中心極限定理より，真の集約値 μ の信頼区間の式を導出する．標本集約値を $\bar{Y_n}$，標本集約値分散を θ_2 とすると，標本集約値が真の集約値 μ の信頼区間に含まれる確率は以下のように定義される．

$$Pr(\epsilon) := Pr[|\bar{Y_n} - \mu| \leqq \epsilon] \tag{1}$$

また，サンプルサイズ n が大きい場合，分散 θ_2 の値は不変分散 $T_{n,2}$ 値に近似する．中心極限定理より，式 (1) は以下のように表される．

$$P\{|\bar{Y_n} - \mu| \leqq \epsilon\} = P\{|\frac{\sqrt{n}(\bar{Y_n} - \mu)}{\sqrt{T_{n,2}(v)}}| \leqq \frac{\epsilon\sqrt{n}}{\sqrt{T_{n,2}(v)}}\} \approx 2\Phi(\frac{\epsilon\sqrt{n}}{\sqrt{T_{n,2}(v)}}) - 1 \tag{2}$$

但し，$\epsilon \geqq 0$ であり，Φ は標準正規分布の確率変数の累積分布関数である．母集団の平均値 μ の信頼係数が p である確率点 z_p は累積分布関数を使用すると，$\Phi(z_p) = \frac{p+1}{2}$ で計算することができる．以上より，クエリ結果の信頼区間は式 (3) で求められる．

$$\bar{Y_n} - \epsilon_n \leqq \mu \leqq \bar{Y_n} + \epsilon_n,\ \ \epsilon_n = \sqrt{\frac{z_p^2 T_{n,2}(v)}{n}} \tag{3}$$

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り，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 $(1 \leqq i \leqq N)$ の N 個の組で表される座標として，次式で定義される．

$$A := [a_1, a_2, \cdots, a_n] \tag{4}$$

距離空間における局所的な範囲を決定するため，ユーザが事前に設定するパラメータ k を用いて，点毎の近傍（以後，k 近傍）を特定する．点 A の k 近傍 $N_k(A)$ は，次式で定義される．

$$N_k(A) := \{\ B \in \mathbb{P} - \{A\}\ |\ d(A,B) \leqq k\text{-}distance(A)\ \} \tag{5}$$

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．

$$LOF_k(A) := \frac{\sum_{B \in N_k(A)} lrd_k(B)\ /\ |N_k(A)|}{lrd_k(A)} \tag{6}$$

$LOF_k(A)$ 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．

$$lrd_k(A) := \frac{|N_k(A)|}{\sum_{B \in N_k(A)} reach\text{-}dist_k(A,B)} \tag{7}$$

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．$reach\text{-}dist_k(A,B)$ は次式で定義される．

$$reach\text{-}dist_k(A,B) := \max\{d(A,B),\ \ k\text{-}distance(B)\} \tag{8}$$

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．

2.2.1　LOF 上限・下限推定

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．従って 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・下限を導出し，(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 [16] をベースとした LOF 値の上限・下限の導出式を定義する [12]．式 (6) 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では，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，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 (6) 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．

$$LOF_k(A).upper := \frac{\sum_{B \in L_k^{max}(A)} lrd_k(B).upper/k}{lrd_k(A).lower}$$

$$LOF_k(A).lower := \frac{\sum_{B \in L_k^{min}(A)} lrd_k(B).lower/k}{lrd_k(A).upper}$$

$$L_k^{max}(A) := \underset{B \in N_k'(A)}{\overset{k}{\arg\max}}\ lrd_k(B).upper$$

$$L_k^{min}(A) := \underset{B \in N_k'(A)}{\overset{k}{\arg\min}}\ lrd_k(B).lower \qquad (9)$$

但し，$N_k'(A)$ は A の k 近傍候補を表し，$L_k^{max}(A)$ 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．

2.3 分析アプリケーション

有用性の定義に基づく評価関数を作成することで，ユーザが発見したい分析結果を定量的に評価することが可能となる．ここで全体データを D とし，D の部分集合を部分データ S で表現し，部分データの集合を $\mathbb{S}$ とする．全体データ D はレコード $d_1, d_2, \cdots, d_{|D|}$（$|D|$ は全体データのレコード数）から構成される．レコード $d_i (1 \leqq i \leqq |D|)$ は集約属性と次元属性の集合で構成される．集約属性の集合 A は，集約属性 $a_1, a_2, \cdots, a_{|A|}$（$|A|$ は集約属性の総数），次元属性の集合 B は，次元属性 $b_1, b_2, \cdots, b_{|B|}$（$|B|$ は次元属性の総数）で構成される．

次元属性 $b_i (1 \leqq i \leqq |B|)$ の値が Y である部分データ S と部分データ集合 $\mathbb{S}$ は，関係代数における選択演算 σ を使用すると式 (10),(11) で定義される．

$$S := \sigma_{b_i = Y}(D) \qquad (10)$$

$$\mathbb{S} := \bigcup_{i=1}^{|B|} \{\sigma_{b_i = Y}(D) \mid Y \in b_i\} \qquad (11)$$

単一の集約属性 a_i と次元属性 b_i から成る OLAP クエリ q は式 (12) ように定義される．

$$q := {}_{b_i}G_{w=f(a_i)} \qquad (12)$$

但し，G は各レコードを次元属性 b_i でグループ化し各グループ毎に集約関数 f を a_i に適用する処理，w はその集約値である．

2.3.1 大域・局所例外データの特定

ユーザが単一の集約属性 a_i と次元属性 b_i，次元属性を指定し，有益な上位 k 件のデータを探索する具体的な分析例として，大域例外部分データの探索問題 [10] や局所例外部分データ

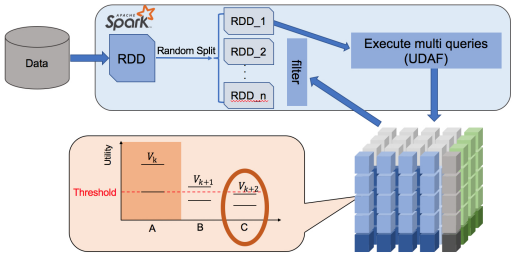

図 3　フレームワークの概要

の探索問題 [12] が存在する．任意の条件から得られる全ての部分データの OLAP クエリ結果に対し乖離度を数値化する必要がある．有用性が高い分析結果を生み出す部分データとは，大域例外部分データでは全体データの分析結果との乖離が大きいと仮定し，局所例外部分データでは近傍点と密度差が大きいと仮定する．これらの探索タスクでは，部分データ集合と単一の集約属性，グループ化属性を事前にユーザが指定し，部分データ集合 $\mathbb{S}$ から乖離度の大きい上位 k 件の部分データ S を求める．複数クエリ結果間の乖離度を数値化する関数として，大域例外データの特定の場合はユークリッド距離を，局所例外データの特定の場合は LOF 値を採用すると，上位 k 件の例外部分データの探索問題は以下のように定義される．

定義 1. 部分データ集合 $\mathbb{S}$，OLAP クエリ q，特定する例外部分データの件数 k を指定し，部分データ集合 $\mathbb{S}$ に属する全ての S の中で $\mathbb{D}(q'(S))$ 上位 k 件の S を特定する．

$$\underset{S \in \mathbb{S}}{\overset{k}{\arg\max}}\ \mathbb{D}(q(S))$$

大域例外部分データを特定する際の $\mathbb{D}(q(S))$ は以下の式で表される．但し，$\mathbb{X}$ は x 軸の属性値集合を表す．

$$\mathbb{D}(q(S)) = \Sigma_{x \in \mathbb{X}} |q_x(S) - q_x(D)|$$

局所例外部分データを特定する際の $\mathbb{D}(q(S))$ は以下の式で表される．但し，LOF 値計算における近傍の範囲を n とする．

$$\mathbb{D}(q(S)) = LOF_n(q(S))$$

3. 提案フレームワーク

本章では，複数部分データに対するクエリ処理を同時実行すると共に，上位 k 件検索による探索空間の削減手法を両立したフレームワークを説明する．このフレームワークは，Apache Spark 上に UDAF を用いて実装した．3.1 節でフレームワークの動作，3.2 節でデータキューブの更新方法，3.3 節でフレームワークのアルゴリズム，3.4 節では本稿で用いた分析アルゴリズムについて説明する．

3.1 フレームワークの概要

ある部分データは式 (10) で得られ，部分データに対してクエリ式 (12) を実行する．探索的データ分析では，様々なパターンで集約処理を実行するため，膨大な部分データに対してクエリを実行する必要がある．提案するフレームワークの処理の概

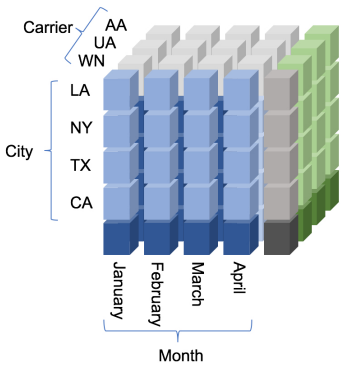

図 4　データキューブの例

要は以下のとおりである．データを読み込む都度そのレコードが対象となる複数の部分データに対するクエリを同時実行し，クエリ結果をデータキューブの形式で保持する．これによりデータ D を 1 スキャンする事で複数の OLAP クエリの結果を同時に作成することができる．更に，統計的信頼区間の技術 [14, 15, 17, 18] を利用することで，一部のデータに対するクエリ結果をもとに全体データに対するクエリ結果を推定し，上位 k 件の候補に成り得ないデータキューブを枝刈りする．上位 k 件の枝刈り判定は分析アルゴリズムによって異なるため，枝刈り判定自体は分析アルゴリズムが行い，フレームワーク側はその判定に従いデータキューブの更新をスキップすることで高速化を図る．

提案するフレームワークの処理の流れは以下の通りである．step.2, 3 の詳細は 3.2 節で述べる．

step.1　統計的信頼区間の技術を適用するためには入力データ D をランダム順に格納する必要がある．そのため，レコードを選択し格納先のデータブロックをランダムに選択する (注1) ことでランダム性を保証する．

条件が成立するまで step.2-5 を実行　（条件）上位 k 件が確定する，あるいは，分割したデータ集合を全て処理

step.2　分割したデータブロックごとに，レコードを逐次的に読む込むことで複数部分データに対するクエリ $q(S)$ を同時実行し，データキューブの基本単位であるキューボイドを差分更新する．

step.3　統計的信頼区間推定技術を適用し，クエリ $q(S)$ の結果の式 (3) より信頼区間をキューボイドの値を用いて推定する．

step.4　クエリ $q(S)$ 結果の信頼区間を用いて，クエリ結果の有用性の上限値と下限値を計算し，有用性の上限値と下限値から上位 k 件に成り得ないクエリ $q(S)$ を特定する．

step.5　特定した上位 k 件に成り得ないクエリ $q(S)$ の枝刈りを行う．

3.1.1　データ構造

定義 1 のクエリ q の group-by 属性と部分データの選択条件の属性集合を全て含む属性集合を入力して，各属性を 1 辺とするデータキューブを構築する．そして，クエリ q とデータキューブについて，クエリ $q(S)$ の結果は group-by 属性によって分割された複数のキューボイドで表される．データキューブは OLAP のクエリ $q(S)$ 結果の区間推定に必要な統計情報（標本数と標本集約値平均，標本集約値分散）を内部で保持する．このデータキューブはキューブの最小単位であるベースキューボイドから構成されている．具体的には，図 4 のデータキューブは，$Month, City, Carrier$ の 3 つの属性をもとに構築されており，それぞれの属性毎に特定の値を指定することによって具体的なベースキューボイドを特定することができる．例えば，$(LA, AA, April)$ は 1 つのベースキューボイドの例である．

3.2　データキューブの更新

データキューブの更新は次の 3 つのステップから構成される．まず入力レーコードに対して差分更新するベースキューボイドを特定する．次に，特定したキューボイドに対しデータの差分更新する．そして，データキューブの信頼区間の計算を式 (3) を用いて行う．以降各ステップの詳細を説明する．

入力レコードに含まれるキューブを構成する各属性の属性値を key とし，入力レコード毎に既存のキューブから差分更新するベースキューボイドを特定する．次に特定したキューボイドが保持している標本数と標本集約値平均，標本集約値分散に関して，レコード毎に計算して以下のように差分更新する．過去に処理した全てのデータブロックである A に，新たに処理しているデータブロック B を差分更新する場合を考える．データブロック A, B のそれぞれの要素数を n_a, n_b，標本集約値平均を $mean_a, mean_b$，標本集約値分散を var_a, var_b とすると，データブロック $A \cup B$ の要素数 n_{ab} と平均 $mean_{ab}$ は以下のように計算する事ができる．

$$n_{ab} = n_a + n_b$$

$$mean_{ab} = \frac{mean_a n_a + mean_b n_b}{n_{ab}}$$

$$var_{ab} = \frac{n_a(var_a + mean_a^2) + n_b(var_b + mean_b^2)}{n_{ab}} - mean_{ab}^2$$

差分更新した結果を用いて，クエリ $q(S)$ 結果の信頼区間の計算を行う時に，中心極限定理の式 (3) を適用する．具体的には，キューボイドが保持しているデータ（標本数 N，標本集約値平均 $Average$，標本集約値分散 Var）からクエリ結果 $q(S)$ の信頼区間は以下のように表される．但し，$\bar{Y}$ が推定したい集約値である．

$$Y\_interval = [\bar{Y} - \epsilon, \bar{Y} + \epsilon], \quad \epsilon = z_p \sqrt{\frac{Var}{N-1}} \qquad (13)$$

3.3　フレームワークのアルゴリズム

Algorithm1 を用いて，本システムのフレームワーク部の動作の詳細を説明する．ユーザは入力として，探索した結果として得たい有用性の高い部分データの件数 k，中心極限定理における信頼係数 z_p，データセットを分割する割合 $x\_array = [x_0\%, x_1\%, \cdots, x_l\%]$，データキューブを作成する際に使用する属性集合 $Dimension$，使用するデータセット D，クエリ q を入力し，上位 k 件の有用性の高い部分データを出

(注1)：分割個数はパラメータで指定

Algorithm 1: フレームワークの動作

Input : k,z_p,x_array,Dimension,DataSet
Output: Results

```
1  DataRDDs ← RandomSampling(Dataset,x_array)
2  CUBE ← ∅
3  Subset_key ← ∅
4  foreach records ∈ DataRDDs do
5      if Subset_key.size > k or Cube is empty then
6          records.filterling(Subsetkey)
7          Cube ++ = QueryExecute(records)
8          foreach (subsetkey,value) ∈ Cube do
9              Cube1[subsetkey].Upper = value + ϵ
10             Cube1[subsetkey].Lower = value - ϵ
11         end
12         Results, Subset_key ← ComputeUtility(Cube1)
13         foreach (subset, value) ∈ Cube do
14             if subset ∉ Subset_key then
15                 Cube.Remove(subset)
16             end
17         end
18     end
19 end
```

力する．最初にユーザが指定した分割割合 x_array のレコードを保持する l 個のデータブロックに入力データ D を分割する（1 行目）．分割されたデータブロック毎に枝刈りされていない $q(S), S \in \mathbb{S}$ に合致しないレコードをスキップする（6 行目）[注2]．その後，入力となる各レコードごとに UDAF を適用することで複数部分データに対するクエリ処理を同時に処理してその結果を用いてデータキューブを差分更新する (7 行目)．データキューブのすべてのキューボイドに対して上限値と下限値を式 (3) を用いて計算する（9, 10 行目）．上限値と下限値を保持したキューボイドを用いて，クエリ $q(S), S \in \mathbb{S}$ の有用性を計算し，有用性の低い部分データ S の枝刈り判定を行う（12 行目）．有用性の計算 · 枝刈り判定は分析アプリケーションが決定する（詳細は 3.4 節で述べる）．枝刈り判定の結果をもとにデータキューブから有用性の低いクエリ $q(S)$ に関するキューボイドの削除を行う（13 行目から 16 行目）．

3.4 分析アルゴリズムと枝刈り

集約関数は AVG か SUM に限定して議論を進める．大量の部分データを全て計算するのはコストが大きい．分析アルゴリズムに使用する有用性にそぐわない部分データも存在するため，統計的信頼区間推定技術を適用し上位 k 件の候補になり得ない部分データを早期に特定し，その部分データの探索を枝刈りすることで高速化を図る．各標本に対し統計的信頼区間の技術を適用したクエリ結果を用いて，全体データからの乖離度の上限値と下限値を推定し，上位 k 件の候補になり得ない部分データの枝刈りを行う．

(注2)：但し，最初のデータブロックでは枝刈り条件が無いためレコードのスキップは行わない．

Algorithm 2: 有用性の計算

Input : k，DataCube // k: 探索する件数，DataCube: クエリ結果の信頼区間を保持するデータキューブ
Output: Subset_key

```
1  DevianceArray ← ∅ // 乖離度を保持する変数
2  All ← All_GroupByAggregate(DataCube)
3  SubsetCube ← Subset_GroupByAggregate(DataCube)
4  foreach (key, XY) ∈ SubsetCube do
5      deviance ← CalucDevianceInterval(XY)
       DevianceArray[key] ← deviance
6  end
7  threshold ← GetTop_k(DevianceArray, k)
8  Subset_key ← {}
9  foreach (key, deviance) ∈ DevianceArray do
10     if threshold ← ≦ deviance.Upper then
11         Subset_key + = {key}
12     end
13 end
```

データキューブから例外部分データを特定する部分データとグループ化属性の粒度のキューボイドを作成し，全体データに関する OLAP クエリ結果のデータキューボイドは複数のベースキューボイドを合算し作成する．部分データ集合 $\mathbb{S}$，OLAP クエリ q，特定する例外部分データの件数 k，閾値は乖離度の上限値が k 番目に大きい部分データの乖離度の下限値を式 (14) とすると，上位 k 件に成り得る部分データの集合 $N_k(S)$ を式 (15) のように定義する．

$$threshold := Top\_k(\mathbb{D}(q(S)).lower) \tag{14}$$

$$N_k(S) := \{S \in \mathbb{S} \mid \mathbb{D}(q(S)).upper \geqq threshold\} \tag{15}$$

Algorithm 2 を用いて探索空間の削減手法の詳細について説明する．Algorithm 2 では，データキューブ $DataCube$，探索する部分データの件数 k を入力 (Input) として，上位 k 件に成り得る部分データ名を出力 (Output) する．入力として与えられるデータキューブの値は既に統計的信頼区間推定の技術を適用したクエリ結果の上限値と下限値を保持している状態である[注3]．まず，データキューブから全体データに関するクエリ結果を保持するデータキューブを作成し（2 行目），同様に部分データに関するデータキューブを使用する粒度のデータキューブを作成する（3 行目）．部分データ毎に乖離度 $\mathbb{D}(q(S))$ の上限値と下限値を計算する（4 行目から 6 行目）．ここで使用する有用性の定義にあった評価関数を用いて，各グループ化属性の値での乖離度の最大値と最小値を算出する．そして，乖離度の上限値が k 番目に大きい部分データの乖離度 $\mathbb{D}(q(S))$ の下限値を閾値に設定する (7 行目)．上位 k 件の候補集合を探索し（8 行目から 12 行目），候補集合 $N_k(S)$ と乖離度を出力する．

(注3)：部分データに関して，グループ化属性値の中に集約対象のレコードが存在しない場合，そのグループ化属性値の集約値を 0 とすることで，グループ化属性値の数を統一

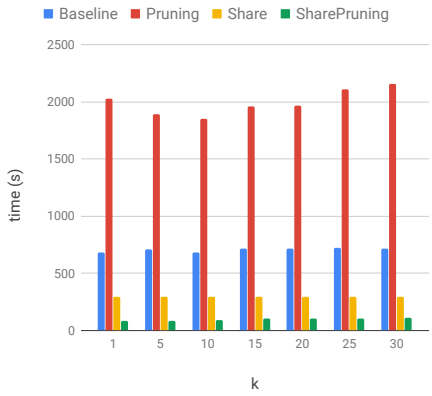

図 5 （大域例外）探索件数を変化させた場合の実行時間

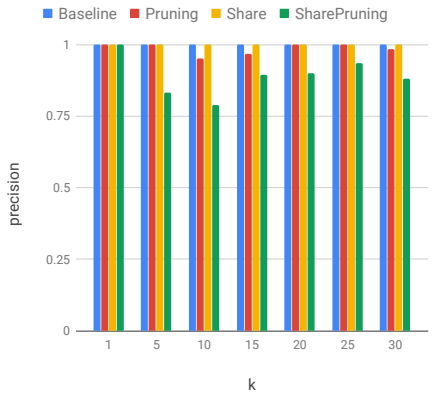

図 6 （大域例外）探索件数を変化させた場合の Precision

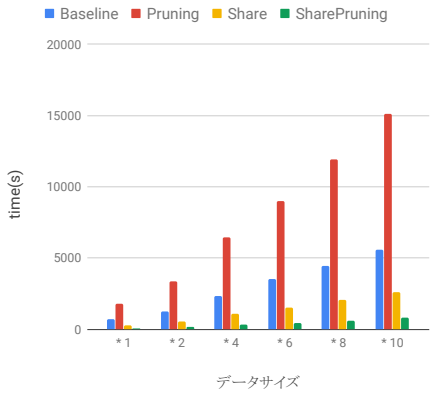

図 7 （大域例外）データサイズを変化させた場合の実行時間

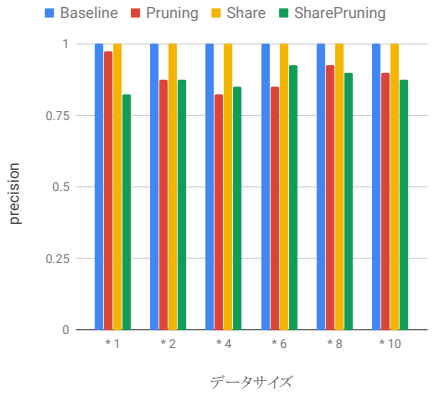

図 8 （大域例外）データサイズを変化させた場合の Precision

4. 評 価 実 験

本章では提案したエンジンにおいて複数クエリの同時実行と探索空間削減手法を両立したフレームワークの有効性を評価した．

4.1 Setup and Setting

提案フレームワークの有効性を検証するために，Table.4. のデータセットに対して，以下の 4 手法の実行時間と分析結果の精度を，探索件数 k を変化させた場合とスケーラビリティに対して比較する実験を行った．

- Baseline
- Pruning
- Query sharing
- Pruning and Query sharing (**Proposal Method**)

また分析結果の精度の指標として，Precision を採用した．

本実験には，CPU が Intel(R) Core(TM) $i5-6600$，クロック周波数は $3.30GHz$，コア数は 4，メモリは $16GB$ の PC を使用した．

FlightData. このデータは，アメリカの航空に関するデータ (注4) であり，出発/到着日時・出発/到着予定時刻・空港・距離やキャリアなど属性が 111 種類存在する．

表 1 実験で用いたデータセットの概要

Dataset	Columns [#]	Row [#]
FlightData	111	5,674,621

4.2 実験結果と考察

パラメータ: 分析者の入力パラメータは，サンプリングにおいて 95% 信頼区間を用い (すなわち，Z_p は 1.96 となる)，データの分割割合を 10% のデータ集合 10 個とし，それぞれのパラメータで 5 回実行した平均値で比較を行う．

図 5,6 は，探索する部分データの件数 k を $k = [1, 5, 10, 15, 20, 25, 30]$ と変化させた場合の大域的例外データの探索の実験結果である．Precision 0.83 の時，最大 88.1% の実行時間の削減が確認できた．また，精度である Precision は最低でも 0.785 であり，一定の精度を保てていると考えられる．図 7,8 は，データのレコードサイズを $[1, 2, 4, 6, 8, 10]$ 倍と変化させた場合の実験結果である．図 7 より，提案手法が比較手法に比べスケーラビリティに優れていることがわかる．

以上のことから，データサイズや探索件数を変化させても高精度の分析結果を出力できていることがわかる．しかし，Pruning のみを行っている手法では大幅に実行時間が伸びているのは，データの読み込み回数が膨大になっていることが原因であると考えられる．また，Precision の値が低くなっている箇所は上位 k 番目付近の部分データの乖離度が枝刈りによって

(注4)：https://transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time

全体平均の値と共に変化することで不正解の回答を出力をしているためである．

5. 関 連 研 究

SEEDB [6,8] は，クエリ探索問題を定義しそれを自動化したフレームワークとデータドリブンで分析するためのデータ構造を提案している．水野ら [10] は，部分データ探索において大域例外を自動で特定するフレームワークを提案している．ユーザが入力したグラフ設定を使用して，Hoeffding の確率不等式を用いて top-k の大域例外を効率的に特定する．小笠原ら [12] は，分析工程の自動化アプローチが SEEDB や水野らとは異なり，事前計算を必要とせず top-k に入り得ない部分データを枝刈りすることで高速化を行い効率的に局所例外を特定する．

Zenvisage [5] は，多数の OLAP クエリを共有化する組み合わせパターンが膨大に存在するため，最適な組み合わせを高速に求めるのは NP 困難であるとされている問題に取り組むことで高速化を行っている．また，最適な組み合わせの OLAP クエリを生成するための言語である ZQL を提案している．そして，3 次元のデータ構造を使用することでクエリ探索と部分データ探索を両立している．

EKI [4] は現存する OLAP ツールが単一の Insights に対し一度の集約演算しか行うことができない問題とツールを使用するユーザからの入力 (group-by attribute や OLAP cell or dimensions) を必要としている問題に取り組んでいる研究である．具体的には，従来の OLAP ツールと比べより複雑な演算処理が可能であり，Insights 間の結果を比較するための評価関数を提案している．ユーザの入力としてデータと演算回数を行う回数のみを必要とし，複雑な演算を行うために演算子と演算を行う軸となる属性の組を自動的に作成し指定された演算回数の結果の top-k 件を出力する．さらに，基本的な集約関数，ランキング評価，%，平均からの乖離値，x 軸の前の値からの増減などの処理を行う演算子を追加している．そして，異なる演算子間の結果をランキング付けするための新たな評価関数として，Impact Measure と Significance Measure の積を提案している．Impact Measure はビジネスの観点では対象としている部分データが Sibiling Group 内で占める割合を表している．Significance Measure は p-value に基づいた演算子毎に適切な帰無仮説を設定し，異常度などの数値である．Data Cube を使用している点，複数の Insights を考慮している点は同一であるが，局所例外探索を適応していないこと，部分データの探索の枝刈りを実施していないことなどの点において本研究と異なる．

6. ま と め

本稿は，中心極限定理を用いた top-k 枝刈りとデータキューブを用いたクエリ共有化を行うことによって処理の最適化を行った，大域的なデータ分析と局所的なデータ分析を両立した探索的データ分析のフレームワークを提案した．

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] J.W. Tukey, EXPLORATORY DATA ANALYSIS, Addison-Wesley, 1977.
[2] W.L. Martinez, A.R. Martinez, and A. R, Exploratory Data Analysis with MATLAB, vol.169, 2004.
[3] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, “Overview of Data Exploration Techniques,” Proceedings of the ACM SIGMOD, 2015.
[4] B. Tang, S. Han, M. Lung, Y. Rui, and D. Dongmei, “Extracting Top-K Insights from Multi-dimensional Data,” Proceedings of the ACM SIGMOD, 2017.
[5] T. Siddiqui, A. Kim, J. Lee, K. Karahalios, and A. Parameswaran, “Effortless Data Exploration with zenvisage: An Expressive and Interactive Visual Analytics System,” Proceedings of the VLDB Endowment, 2016.
[6] M. Vartak, and S. Madden, “SEEDB : Automatically Generating Query Visualizations,” Proceedings of the VLDB, vol.7, no.13, pp.1581–1584, 2014.
[7] A. Parameswaran, N. Polyzotis, and H. Garcia-Molina, “SeeDB: Visualizing Database Queries Efficientl,” Proceedings of the VLDB Endowment, vol.7, no.4, pp.325–328, 2013.
[8] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis, “SEEDB : Efficient Data-Driven Visualization Recommendations to Support Visual Analytics,” Proceedings of the VLDB Endowment, 2015.
[9] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi, “Distributed and interactive cube exploration,” Proceedings of the ICDE, pp.472–483, 2014.
[10] M. Yohei, S. Yuya, and O. Makoto, “Efficient Data Slice Search for View Detection,” Proceedings of the DOLAP, 2017.
[11] L.O. Factor, “Mining Top-n Local Outliers in Large Databases,” Proceedings of the ACM SIGKDD, pp.293–298, 2001.
[12] 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，” Proceedings of the DEIM，2017.
[13] B. Mozafari, “Approximate Query Engines: Commercial Challenges and Research Opportunities,” Proceedings of the ACM SIGMOD, 2017.
[14] P.J. Haas, and S. Jose, “Large-Sample and Deterministic Confidence Intervals for Online Aggregation,” Proceedings of the SSDBM, 1997.
[15] J.F. Naughton, “Technical Perspective: Optimized Wandering for Online Aggregation,” Proceedings of the ACM SIGMOD, vol.46, 2017.
[16] M.M. Breunig, H.P. Kriegel, R.T. Ng, and J. Sander, “LOF: Identifying Density-Based Local Outliers,” Proceedings of the ACM SIGMOD, no.October 2017, pp.1–12, 2000.
[17] J.M. Hellerstein, P.J. Haas, and H.J. Wang, “Online aggregation,” Proceedings of the ACM SIGMOD, 1997.
[18] F. Li, B. Wu, K. Yi, and Z. Zhao, “Wander Join: Online Aggregation via Random Walks,” Proceedings of the ACM SIGMOD, 2016.