Vol.2021-DBS-174 No.2
2021/12/27

シノプシスに基づく近似問合せ処理における誤差保証の検討

倪　天嘉[1,a)]　杉浦 健人[1,b)]　石川 佳治[1,c)]　陸 可鏡[1,d)]

概要：データ量の増加と分析の要求の高度化に伴い，データベースにおける問合せ処理がより重要となってきている．膨大なデータに基づく問合せを効率的に実行するための技術として，近年近似的問合せ処理（approximate query processing, AQP）が着目されている．AQP では，要約データやサンプルを用いて効率的に問合せ処理を行う．本研究では既存の近似的問合せ処理研究 BAQ のフレームワークを改善し，誤差閾値以内でシノプシスを生成できる近似計算手法を提案する．BAQ では，簡単な SQL による集約問合せに焦点を当てたシノプシス構築技術であり，シノプシスを用いて誤差を保証した問合せ処理を実現する．ただし，COUNT，SUM，AVG 集約計算について対象データが実数全体である場合に厳密な誤差保証を提供できないという問題がある．本稿では BAQ の問題を解決し，提案手法による近似的問合せ処理システムを実装し，誤差の保証について議論する．

1. はじめに

近年，データ量の増加と分析の要求の高度化に伴い，データベースにおける問合せ処理がより重要となってきている．データベース全体ではなく，データベースの一部分のデータや要約データを用いて効率的に問合せ処理を行う技術として，近似的問合せ処理（approximate query processing, AQP）が着目されている [1,3,5]．AQP ではシノプシスを用いた問合せが行われる．シノプシス（synopsis）とは，対象となるデータをコンパクトに集約したデータを指す一般的な概念であり，さまざまなアプローチがある [4,6]．AQP からのシノプシスの問合せ結果には誤差が含まれるので，誤差を小さくすること，また，誤差を推定することが重要となる [2]．

それで Li ら [3] は，一部のデータを保持できて，問合せ結果の誤差保証ができるように，データベース上にシノプシスを構築する手法（BAQ）を提案した．BAQ では，ユーザが設定する誤差の閾値と典型的なワークロードを示す問合せ集合を用いてオフライン処理でデータベースからシノプシスを生成し，そのシノプシスを使用して効率的にオンライン問合せに回答する．他の要約データを使用するアプローチ (サンプリング，分位数など) と比べて，BAQ は一般的な集計関数（COUNT，SUM，AVG，MIN，MAX）に対して効率的に誤差を保証できる特徴がある．例えば，TPC-H の `lineitem` テーブルを使用する場合，以下のような問合せを含むワークロードを処理できる．

```
SELECT AVG(price) FROM lineitem
```

ただし，BAQ では，正のデータのみに対する誤差を保証できる．同時に，MIN/MAX 関数を計算する際に，誤差は小さいので無視できるという結論が出ているが，実際の実装には無視できない推定誤差が存在する．実際のデータセットを考えると，実数全体についても誤差保証がある近似的問合せ処理技術が必要である．また，BAQ で対応できるのは構造が簡単な問合せのみであり，例えば結合などの操作を含む問合せへの対応方法は明示されていない．つまり，BAQ には結合や自己結合などの操作を含む問合せに対するシノプシスの生成手法など，検討と改善の余地がある．

そこで，本研究では以上の問題を解決するために，BAQ のアプローチを拡張した近似的問合せ処理手法を提案する．加えて，対応可能な問合せを対象に，誤差の分析と保証について検討する．

2. 準備

本章では本稿の議論に必要な概念について説明する．以下では，対象とする問合せ，誤差の指標である相対誤差について述べる．

2.1 対象問合せ

本研究では以下の演算を用いた OLAP 問合せを想定する．

- COUNT，MIN，MAX，SUM，および AVG での集約
- 一つ以上のカテゴリ属性に対する $=$，$\neq$ と集約計算
- 一つ以上の数値属性に対する $=$，$\neq$，$>$，$\geq$，$<$，およ

[1] 名古屋大学大学院情報学研究科
[a)] ni@db.is.i.nagoya-u.ac.jp
[b)] sugiura@i.nagoya-u.ac.jp
[c)] ishikawa@i.nagoya-u.ac.jp
[d)] lu@db.is.i.nagoya-u.ac.jp

び $\leq$ を用いた条件での選択と集約計算
- グルーピングとランキング

2.2 相対誤差

本研究では与えられた OLAP 問合せに対し，集約結果の真値と近似値との誤差 err として相対誤差を用いる．2 つの値 $x, y \in R$ の相対誤差を以下の式で定める．

$$err(x,y) = \begin{cases} \frac{|x-y|}{\epsilon} & (x=0) \\ \left|\frac{x-y}{x}\right| & (\text{otherwise}) \end{cases} \quad (1)$$

ただし，x を真値，y を近似値とし，$\epsilon \in R$ を非常に小さな正の値とする．シノプシスを生成するとき，バケット集合 B_i はある数値属性 $A_i \in R$ を重複なしで分割した範囲の集合であり，各バケット中における平均値と任意の値の相対誤差が δ 以内となるように生成する．つまり，各バケット $b \in B_i$ の平均値 p を代表値として考える，各バケットは，以下の式が成り立つように定める．

$$p = \frac{\sum_{x \in b} x}{|b|} \quad (2)$$
$$\forall x \in b, err(x,p) \leq \delta$$

本研究では，事前にユーザから相対誤差のしきい値 δ とワークロードが与えられると想定する．この想定のもとで，誤差以内で式 (2) でバケットを生成し，シノプシスを計算する．問合せ処理時には，ユーザはシノプシスを使用し，問合せを高精度かつ効率的な近似的処理が実行できる．

3. 提案手法

ここでは，[3] をもとに拡張した手法を示す．よく使われる問合せの集合（ワークロード）が与えられると，ユーザから指定された相対誤差のしきい値をもとに，各問合せに関連するデータを要約し，新たなバケットを計算する．3.1 節で詳しく説明する．次に，シノプシスまでの生成過程について 3.2 に示している．オンラインの問合せ処理について，3.3 で述べる．

3.1 数値属性のバケット分割

バケットへの分割は，BAQ [3] で述べられたとおり，各数値属性の正の最小値を基準とした方法で行える．例えば，正のゼロでない最小値 x_1 を基準としたとき，以下の式を満たす x_n を x_1 と同一のバケットに含める．

$$x_n \leq (1+\delta)x_1 \quad (3)$$

具体的に，数値データ $\{120, 130, 137, 140, 143, 146, 150, 155, 158, 161, 185, 190\}$ と $\delta = 0.2$ が与えられた場合，1 番目のタプル 120 をグループ 1(つまりシノプシスのレコード 1) の基準として式 (3) で判定する．5 番目のタプル 143 まで $120 * (1 + 0.2)$ より小さく，6 番目のタプル 146 から 144 より大きいので，$\{120, 130, 137, 140, 143\}$ をグループ 1 として保存する．146 がグループ 2 の基準として 161 まで式 (3) を満たすので，$\{146, 150, 155, 158, 161\}$ をグループ 2 として保存する．それで三つのグループを取得できて，各グループの先頭の値，最小値，最大値及び要素数をバケットとして保存する．たとえばグループ 1 の場合，最後に $\{120, 120, 143, 5\}$ となる．

Algorithm 1: is_possible_to_add

```
Data: データ集合 array，誤差のしきい値 δ
Result: true or false
begin
    mean = array.mean
    min = array.min
    if |(mean − min)/min| ≤ δ then
        return true
    else
        return false
```

Algorithm 2: BucketGeneration

```
Data: 昇順ソート済み元データ T，誤差のしきい値 δ
Result: バケット blk, mean, count
begin
    n := len(T)
    blk := T[0] // 新たなバケットを生成
    count := count + 1 // blk に含まれる要素数
    mean := T[0] // blk の平均値
    for i := 1 → n − 1 do
        if is_possible_to_add(T[i], δ) then
            blk := blk ∪ T[i]
            count := count + 1
            mean := (count ∗ mean + T[i])/(count + 1)
             // 平均値を更新
            if i == n − 1 then
                return blk, mean, count
            else
                return blk, mean, count
                new_block(T[i])
                count := 1
                mean := T[i]
```

以上のバケット分割方法で生成したシノプシスのサイズは元データのサンプルより小さくなるが，より小さなシノプシスを得るには，バケットの分割には改善の余地がある．

本研究では，各バケット b の平均値 p を代表値として考える，各バケットは，式 (2) が成り立つように定める．

式 (2) を用いてアルゴリズム 2 でバケットを分割する．

表 1　データベースのスキーマ

lineitem.tbl	price, discount, returnflag, linestatus

前処理として，元データのタプルを数値属性に基づいて昇順にソートしたデータを T とする．相対誤差のしきい値 δ とソートされたデータ T を入力データし，データのサイズ n，blk 及び要素数 $count$ と $mean$ を初期化する (2–5 行目)．続いて，順番に元データ T のタプルをアルゴリズム 1 により，式 (2) に基づいてバケットを生成する (7–14 行目)．具体的には，バケットへの分割は，各数値属性の正及び負の最小値を基準として行える．

例えば，正のゼロでない最小値 x_1 を基準としたとき，以下の式でバケットが生成できる．

$$p_n \leq (1+\delta)x_1 \tag{4}$$

具体的に，数値データ$\{120, 130, 137, 140, 143, 146, 150, 155, 158, 161, 185, 190\}$ と $\delta = 0.2$ が与えられた場合，1 番目のタプル 120 をグループ 1(つまりシノプシスのレコード 1) の基準として式 (4) で判定する．10 番目のタプル 161 までの平均値 144 は $120 * (1 + 0.2)$ より大きくなく，11 番目のタプル 185 までの平均値 148 は 144 より大きいことから，$\{120, 130, 137, 140, 143, 146, 150, 155, 158, 161\}$ をグループ 1 として保存される．185 がグループ 2 の基準として 190 まで式 (4) を満たすので，$\{185, 190\}$ をグループ 2 として保存する．それで二つのグループを取得できて，各グループの平均値，最小値，最大値及び要素数をバケットとして保存する．たとえばグループ 1 について，最後に $\{144, 120, 161, 10\}$ が得られる．

BAQ と比較して，バケットの対応する範囲は次第に大きくなっていく．なお，負のバケットも同様の方法で生成し，値がゼロのタプルが存在する場合はゼロのみのバケットも追加で生成する．

3.2　シノプシスの生成

一つ以上の数値属性とカテゴリ属性に関するシノプシスの生成過程を説明する．TPC-H ベンチマークのデータベースを想定する．

```
SELECT linestatus, MIN(price)
FROM lineitem
WHERE price > 100000
AND discount > 0.2
GROUP BY linestatus
```

以上の問合せを例として，元データ T として lineitem の (linestatus, returnflag, discount, price) に関するレコードをシノプシスの処理対象とする．まず，カテゴリ属性による元データをグルーピングする．例で linestatus と returnflag の属性による各値に基づいてグルーピングとランキングを行う．次に，各カテゴリ属性の値による数値属性のデータ集合についてバケットを計算する．つまり，linestatus の各値に対し，price について 3.1 のアルゴリズム 1 と 2 でバケットを計算し，(linestatus, price_mean, price_max, price_min, SF1) の要約データ，つまりシノプシスを得る．数値属性 returnflag と数値属性 discount による (returnflag, discount_mean, discount_max, discount_min, SF2) のスキーマに関するシノプシスを取得できる．最後に，二つのシノプシスを元テーブルとマッチングし，組み合わせて最後のシノプシス (price, discount, returnflag, linestatus, price_mean, price_max, price_min, SF1, discount_mean, discount_max, discount_min, SF2) を生成する．シノプシス中には SF1=0, SF2=0 となるレコードは含まない．

3.3　問合せ処理

カテゴリ属性による選択・グルーピングに加え，選択条件に数値属性が用いられる場合について述べる．つまり，以下のような問合せ 2 が対象となる．

```
SELECT linestatus, COUNT(*)
FROM lineitem
WHERE price > 100
GROUP BY linestatus
```

この問合せに対し，3.1 と 3.2 の手順に従い (linestatus, price_max, price_min, SF1) のシノプシスを生成する．シノプシス中には SF1=0 となるレコードは含まない．つまり，選択・グルーピング条件で指定されたカテゴリ属性に加え，選択条件で指定された数値属性のバケットをカテゴリ属性とみなしてグルーピングを行い，各グループに属するタプル数をシノプシスのレコードとする．

バケットに分割された数値属性に対して選択条件が与えられたとき，各バケットは 1) バケット全体が条件を満たす，2) バケット全体が条件を満たさない，3) バケットの一部が条件を満たすの 3 通りに分けられる．1) のバケットについては，元の問合せを書き直して以下の問合せになって効率的に計算する．

```
SELECT linestatus, SUM(SF)
FROM synopsis
WHERE price_min > 100
GROUP BY linestatus
```

3) のバケットの場合，100 を含むバケットの上限値を取得する必要があるので，部分的条件を満たすバケットの最大値 price_max を返し，問合せの選択条件に設定して元テーブルに対する計算する．

```
SELECT linestatus, COUNT(*)
FROM lineitem
WHERE price > 100
AND price < price_max
GROUP BY linestatus
```

条件を部分的に満たすバケットのうち条件を満たすタプル数を元テーブルを用いて正確に計算し，誤差をゼロとできて，全体として，ゼロの誤差で COUNT 集約を計算できる．各集約関数について一つ以上の数値属性に対する選択・グ

ルーピングについて，提案手法は BAQ より小さい誤差となる．4 章に詳しい説明する．

4. 誤差の保証

4.1 COUNT 集約

COUNT 集約における誤差保証について述べる．ただし，選択条件がない場合，つまり元テーブル中の全タプルが対象となる際の結果は明らかであるため省略する．以下では，選択・グルーピング条件に数値属性を含まない場合と含む場合のそれぞれについて説明する．

4.1.1 選択・グルーピング条件が数値属性を含まない場合

まず，問合せにいずれの条件も与えられない場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT COUNT(*)
FROM lineitem
WHERE linestatus = 'URGENT'
```

この問合せに対して，任意の linestatus を含むシノプシスについて，タプル数の総和（つまり SF の総和）を計算する．数値属性に関するデータを要約しないので，ここで誤差ゼロで COUNT 集約を処理できる．

4.1.2 選択条件に一つの数値属性を含む場合

カテゴリ属性による選択・グルーピングに加え，選択条件に数値属性が用いられる場合について 3.3 節に説明したので省略する．選択条件に一つの数値属性がある場合がゼロの誤差で計算できる．

4.1.3 選択条件に二つ以上の数値属性を含む場合

選択条件に二つ以上の数値属性が用いられる場合 COUNT 関数の誤差も保証できる．以下の問合せ例に対しては，(price, discount, linestatus, price_mean, price_max, price_min, SF1, discount_mean, discount_max, discount_min, SF2) というシノプシスを生成する．

```
SELECT linestatus, COUNT(*)
FROM lineitem
WHERE price > 100000
AND discount > 0.2
GROUP BY linestatus
```

なお，シノプシスには，同時に $SF1 = 0$ と $SF2 = 0$ となるレコードは含まない．つまり，選択・グルーピング条件で指定されたカテゴリ属性に加え，選択条件で指定された数値属性のバケットをカテゴリ属性とみなしてグルーピングを行い，各グループに属するタプル数をシノプシスのレコードとする．

単一の問合せの処理によると同様に，条件を満たすバケットと部分的な条件を満たすバケットをそれぞれ計算する．条件を部分的に満たすバケットの結果は元テーブルをもとに計算したため誤差がゼロであり，条件を全部満たすバケットも誤差がゼロである．よって一つ以上の数値属性について選択する場合誤差ゼロの保証がある．

4.2 MIN/MAX 集約

MIN/MAX 集約における誤差保証について述べる．MIN/MAX 集約はほぼ同様の手順で誤差保証が考えられるため，以下では MIN 集約についてのみ述べる．以下では選択・グルーピング条件が与えられない場合，条件にカテゴリ属性のみが用いられる場合，及び条件に数値属性を含む場合それぞれについて説明する．

4.2.1 選択・グルーピング条件がない場合

まず，問合せにいずれの条件も与えられない場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT MIN(price)
FROM lineitem
```

この問合せに対し，対応するシノプシスのスキーマを (price_min, price_max, SF) とし，元テーブルを用いてインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．つまり，集約対象の数値属性のバケットをカテゴリ属性とみなしてグルーピングを行い，各バケット $b \in B$ の最小値 min，最大値 max 及び各バケット中のタプル数をシノプシスのレコードとする．シノプシスを用いることで，MIN(price_min) で効率的に計算できて，相対誤差はゼロである．

4.2.2 条件がカテゴリ属性のみの場合

選択・グルーピング条件にカテゴリ属性が用いられる場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT linestatus, MIN(price)
FROM lineitem
WHERE returnflag = 'O'
GROUP BY linestatus
```

ここで，対応するシノプシスのスキーマを (linestatus, returnflag, price_min, price_max, SF) とし，元テーブルを用いてインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．つまり，選択・グルーピング条件で使用されたカテゴリ属性及び集約対象の数値属性のバケットを用いてタプルのグルーピングを行い，各グループ中のバケットの代表値及び属するタプル数をシノプシスのレコードとする．シノプシスを構築したとき，誤差保証は MIN(price_min) で，条件を含まない場合と同様に行える．上記のシノプシスは，選択・グルーピング条件で使用されたカテゴリ属性によって元テーブルを分割し，分割後の部分テーブルで集約対象の数値属性をバケットに分割したものとみなせる．つまり，分割後の部分テーブルでは上記と同様の議論ができ，結果の相対誤差はゼロである．

4.2.3 選択条件に一つの数値属性を含む場合

カテゴリ属性による選択・グルーピングに加え，選択条件に数値属性が用いられる場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT linestatus, MIN(price)
```

```
FROM lineitem
WHERE price > 100
GROUP BY linestatus
```

この問合せに対し，スキーマ（linestatus, price_mean, price_min, price_max, SF）を持つシノプシスを考え，元テーブルを用いてそのインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．つまり，選択・グルーピング条件で指定されたカテゴリ属性，選択条件で指定された数値属性のバケット，及び集約対象の数値属性のバケットを用いてグルーピングを行い，各グループの集約対象の代表値及び属するタプル数をシノプシスのレコードとする．

3.3 のような問合せを処理する場合，各バケット中の任意の二つの値の相対誤差が δ 以内であることは変わらないため，得られる回答と最小値の真値との相対誤差は δ 以内である．

4.2.4 選択条件に二つ以上の数値属性を含む場合

カテゴリ属性による選択・グルーピングに加え，選択条件に二つの数値属性が用いられる場合について述べる．つまり，3.2 の問合せが対象となる．

二つ以上の数値属性が選択条件に存在する場合においても，MIN 集約の結果の相対誤差は δ 以内に保証できる．COUNT 集約と同様に選択条件によって二つの数値データを絞り込む．数値属性の条件によって同時に部分的に条件を満たすバケットが処理対象とする．次に，グルーピング条件によって分割された各グループ内のレコードに対し，最小値の代表値を結果として返す．部分的に条件を満たすバケットの中も任意の二つの値の相対誤差が δ 以内であることにより，上記の手続きで得られる回答の誤差も δ 以内である．

4.3 SUM/AVG 集約

SUM/AVG 集約における誤差保証について述べる．ただし，SUM/AVG 集約はほぼ同様の手順で誤差保証が考えられるため，以下では SUM 集約についてのみ述べる．以下では選択・グルーピング条件が与えられない場合，条件にカテゴリ属性のみが用いられる場合，及び条件に数値属性を含む場合それぞれについて説明する．

4.3.1 選択・グルーピング条件がない場合

まず，問合せにいずれの条件も与えられない場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT SUM(price)
FROM lineitem
```

この問合せに対し，対応するシノプシスのスキーマを (price_mean,price_min, price_max, SF) とし，元テーブルを用いてインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．各グループの集約対象の代表値，最小値，最大値及び属するタプル数をシノプシスのレコードとする．シノプシスに対し，price_mean $\times$ SF の総和を近似的な総和として返す．BAQ [3] で提案された方法では集約対象の数値属性のドメインが正ないし負いずれかの場合のみ相対誤差 δ 以内を保証できるが，提案手法では代表値として各バケットの平均値を用いることで誤差 0 で回答できる．以下では，それぞれの誤差保証について順に述べる．

まず，BAQ における誤差保証について説明する．シノプシス中に k 個のバケットがあるとする．対象の数値属性の各値を v，各バケットの代表値を p で表すとき，合計の真値 s と近似値 s' との相対誤差は以下の式で表せる．

$$err(s,s') = \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{k} SF_i \cdot p_i}{\sum_{i=1}^{n} v_i} \right| \tag{5}$$

$$= \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{k} \sum_{j=1}^{SF_i} p_i}{\sum_{i=1}^{n} v_i} \right| \tag{6}$$

k 個のバケットは元々テーブルを分割したものであるため，各バケット b_i 中の各真値を v_{ij} と表すことで以下のように表せる．

$$err(s,s') = \left| \frac{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} v_{ij} - \sum_{i=1}^{k} \sum_{j=1}^{SF_i} p_i}{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} v_{ij}} \right| \tag{7}$$

$$= \left| \frac{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} (v_{ij} - p_i)}{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} v_{ij}} \right| \tag{8}$$

ここで，BAQ 中では暗黙のうちに行われていたが，**対象の集計属性のドメインが正ないし負のみである**と仮定すると以下の式が導ける．

$$err(s,s') = \frac{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} |v_{ij} - p_i|}{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} |v_{ij}|} \tag{9}$$

$$\leq \frac{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} \delta \, |v_{ij}|}{\sum_{i=1}^{k} \sum_{j=1}^{SF_i} |v_{ij}|} = \delta \tag{10}$$

一方で，代表値をバケット毎の平均値とすることで**実数全体においても誤差保証が可能**であり，誤差はゼロとなる．

$$err(s,s') = \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{k} SF_i \cdot p_i}{\sum_{i=1}^{n} v_i} \right| \tag{11}$$

$$= \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{k} SF_i \cdot \frac{\sum_{j=1}^{SF_i} v_{ij}}{SF_i}}{\sum_{i=1}^{n} v_i} \right| \tag{12}$$

$$= \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{k} \sum_{j=1}^{SF_i} v_{ij}}{\sum_{i=1}^{n} v_i} \right| \tag{13}$$

$$= \left| \frac{\sum_{i=1}^{n} v_i - \sum_{i=1}^{n} v_i}{\sum_{i=1}^{n} v_i} \right| = 0 \tag{14}$$

4.3.2 条件がカテゴリ属性のみの場合

選択・グルーピング条件にカテゴリ属性が用いられる場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT linestatus, SUM(price)
FROM lineitem
WHERE linestatus = 'URGENT'
GROUP BY linestatus
```

ここで，対応するシノプシスのスキーマを (linestatus, price_mean, price_min, price_max,SF) とし，元テーブルを用いてインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．選択・グルーピング条件で使用されたカテゴリ属性及び集約対象の数値属性のバケットを用いてタプルのグルーピングを行い，各グループ中のバケットの平均値及び属するタプル数をシノプシスのレコードとする．シノプシスにより，SUM(price_mean×SF) で SUM 集約に対する誤差保証が，条件を含まない場合と同様に行える．

生成したシノプシスは，選択・グルーピング条件で使用されたカテゴリ属性によって元テーブルを分割し，分割後の部分テーブルで集約対象の数値属性をバケットに分割したものとみなせる．つまり，分割後の部分テーブルでは 4.3.1 節と同様の議論ができ，結果の相対誤差がゼロであることを導ける．

4.3.3 選択条件に一つの数値属性を含む場合

カテゴリ属性による選択・グルーピングに加え，選択条件に数値属性が用いられる場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT linestatus, SUM(price)
FROM  lineitem
WHERE price > 100
GROUP BY linestatus
```

この問合せに対し，スキーマ (linestatus, price_mean, price_min,price_max,SF) を持つシノプシスを考え，元テーブルを用いてインスタンスを生成する．なお，シノプシス中には $SF = 0$ となるレコードは含まない．つまり，MIN/MAX 集約の場合と同様にシノプシス 5 を構築し，代表値のみ各バケットの平均値を使用し，問合せを処理して近似結果を返す．条件を部分的に満たすバケットのうち条件を満たすタプル数を元テーブルを用いて正確に計算し，前節で述べたとおり，条件を完全に満たすバケットの合計値は誤差 0 で計算できる．一方で，条件を満たすタプルの正確な数を計算しかつ各バケットが正ないし負の数しか含まないよう生成することで，部分的に条件を満たすバケットの合計は式 (6) から式 (10) に示した議論のとおり誤差 δ 以内で計算できる．

4.3.4 選択条件に二つ以上の数値属性を含む場合

カテゴリ属性による選択・グルーピングに加え，選択条件に二つの数値属性が用いられる場合について述べる．つまり，以下のような問合せが対象となる．

```
SELECT linestatus, SUM(price),
SUM(discount)
FROM  lineitem
WHERE price > 100000
AND discount > 0.2
GROUP BY linestatus
```

ここで，3.2 の内容による (price, discount, returnflag, linestatus, price_mean, price_max, price_min, SF1, discount_mean, discount_max, discount_min, SF2) のスキーマに関するシノプシスを取得できる．なお，シノプシス中同時には $SF1 = 0$, $SF2 = 0$ となるレコードは含まない．

次に，COUNT 集約と同様に選択条件によって二つの数値データを絞り込む．二つの数値属性の条件によって条件を満たすバケットと部分的に条件を満たすバケットが処理対象として計算する．条件を部分的に満たすバケットのうち条件を満たすタプル数を元テーブルを用いて正確に計算し，誤差をゼロとできて，4.3.3 に示した条件を全部満たすバケットについても δ 以内の誤差保証があるので，一つ以上の数値属性について問合せする場合，δ 以内の誤差で計算できる．

5. まとめと今後の課題

本稿では，誤差の保証がある近似的問合せ処理に対して，データを要約する提案手法について誤差の保証を議論した．今後の課題としては，今回議論した提案手法に基づく複雑な集計計算について新しい指針の考案とその実装が実現する．

謝辞 本研究の一部は科研費（16H01722, 20K19804, 21H03555）による．

参考文献

[1] B. Mozafari, N. Niu: A handbook for building an approximate query engine. IEEE Data Engineering Bulletin **38**(3), 3–29 (2015)
[2] K.Li, G.Li: Approximate query processing: What is new and where to go?. Data Science and Engineering **3**, 379–397 (2018)
[3] K. Li, Y. Zhang, G. Li, W. Tao, Y. Yan: Bounded approximate query processing. IEEE TKDE **31**(12), 2262–2276 (2019)
[4] B. Walenz, S. Sintos, S. Roy, J. Yang,: Learning to sample: Counting with complex queries. PVLDB **13**(3), 389–401 (2019)
[5] S. Chaudhuri, B. Ding, S. Kandula: Approximate query processing: No silver bullet. In: Proc. SIGMOD, pp. 511–519. ACM (2017)
[6] Q. Ma, P. Triantafillou: DBEST: Revisiting approximate query processing engines with machine learning models. In: Proc. SIGMOD, pp. 1553–1570. ACM (2019)
[7] N. Potti, J. Patel: DAQ: A new paradigm for approximate query processing. PVLDB **8**(9), pp. 898–909. ACM (2015)