

誤差を保証する近似的問合せについて

倪 天嘉[†] 杉浦 健人[†] 石川 佳治[†]

[†] 名古屋大学大学院情報学研究科 〒 464-8601 愛知県名古屋市千種区不老町

E-mail: ni@db.is.i.nagoya-u.ac.jp, sugiura@db.is.i.nagoya-u.ac.jp, ishikawa@i.nagoya-u.ac.jp

あらまし 大量のデータに基づくデータベース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 (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た問合せが行われる。本研究では、Li ら [1] により提案された Bounded Approximate Query (BAQ) と呼ばれる、リレーショナルデータベース上のシノプシス構築とそれに基づく問合せの技術に着目する。BAQ は、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。本研究では BAQ の問題点を指摘し、その一般化を行い、実装について議論する。

キーワード 近似的問合せ処理, シノプシス

1 背景

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た。大量のデータを対象としたデータベース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 [2–4]。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正確ではないものの効率的に問合せ処理を行う技術については、これまでも多くの研究があるが、さらに着目されている。ただし、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。そして、数値データについて近似計算が大事なことになる。

数値データを近似するアプローチは大きくデータを一切保持しないと一部を保持する手法の二つに分類できる。ヒストグラム、ワープレットなど関数による近似手法、統計モデルを使用する研究 [5–7] で、大規模なデータを近似計算できる。ただし、そういう手法が元データを保持しないので、取得した結果がユーザにとって把握性が弱い。それに対して、元データを一部を保持手法について、サンプリング、分位数と BAQ [1, 6, 8, 9] など研究がある。

それここで Li ら [1] 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、リレーショナル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。問合せ処理では、与えられた問合せに対してシノプシスを用いて誤差を保証した結果を返すことができる。元のデータを一部保存するので、ユーザがデータを把握しやすいという利点がある。加えて、他の一部の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 が COUNT, SUM, AVG 関数のみならず、MIN と MAX 関数についても効率的に誤差の保証ができる点が挙げられる。

ただし、BAQ にはいくつかの問題がある。

(1) BAQ では、正のデータのみに対する誤差を保証でき

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。

(2) BAQ で正のデータを考えたとき、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。

(3) BAQ 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、より削減するために、シノプシスの生成アルゴリズムに改善の余地がある。

そこで本研究では、以上の制限を克服するために、[1] のアプローチを拡張した誤差を保証する近似的問合せ処理フレームワークを提案する。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が与えられると、データベースからオフライン処理でシノプシスを生成し、そのシノプシスを使用して効率的なオンライン問合せに回答する。

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、対象とするデータ及び問合せ、誤差の指標である相対誤差、及び数値属性のバケット分割について述べる。

2.1 対象データ及び問合せ

本研究では単一のリレーションスキーマ $R = (A_1, A_2, \dots, A_m)$ を対象とし、そのインスタンスとして n 個のタプルを持つテーブル T を扱う。各 $A_i \in R$ は数値属性またはカテゴリ属性であるとし、そのドメインを D_i で表す。

上記のようなテーブル T に対し、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, 及び AVG での集約。
- カテゴリ属性に対する $=$ 及び \neq での、数値属性に対する $=$, \neq , $>$, \geq , $<$, 及び \leq を用いた条件での選択。
- 上記条件に対する論理和 \vee , 論理積 \wedge , 及び否定 \neg を用いた選択。
- カテゴリ属性に対するグルーピング。

例えば、以下のような問合せを処理対象として扱う。

```
SELECT status, SUM(price)
FROM T
WHERE priority = 'URGENT' AND price > 100
GROUP BY status
```

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 を相対誤差で考える。2つの値 $x, y \in \mathbb{R}$ 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、 $\epsilon \in \mathbb{R}$ を非常に小さな正の値とする。

本研究では、事前にユーザから相対誤差のしきい値 δ 及び処理のワークロードを表す問合せの集合 Q が与えられると想定する。この想定のもとで、問合せ集合 Q 内で使用された集約演算に対し、近似値の相対誤差を δ 以内に収める削減されたタプルの集合を生成する。この削減されたタプル集合を、本研究ではシノプシス (synopsis) と呼ぶ。

2.3 数値属性のバケット分割

シノプシスの構築と誤差保証において重要となる数値属性のバケット分割について述べる。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任意の二つの値の相対誤差が δ 以内となるよう生成する。つまり、各バケット $b \in B_i$ に対して以下の式が成り立つ。

$$\forall x, y \in b, err(x, y) \leq \delta \quad (2)$$

バケットへの分割は、BAQ [1] 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 0.2 の δ が与えられる場合、一番目のタプル 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。五番目のタプル 143 まで $120 * (1 + 0.2)$ より小さく、六番目のタプル 146 から 144 より大きいだから、{120, 130, 137, 140, 143} をグループ 1 として保存される。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 として保存される。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。グループ 1 の場合、最後に {120, 120, 143, 5} を取得できる。

ただし、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取得するために、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、以下の式が成り立つように定める。

$$p = \frac{\sum_{x \in b} x}{|b|} \quad (4)$$

$$\forall x \in b, err(x, p) \leq \delta$$

式 (4) を用いてアルゴリズム 2 でバケットを分割する。

Algorithm 1: is_possible_to_add

Data: データ集合 $array$, 誤差のしきい値 δ

Result: $true$ or $false$

```
1 begin
2   mean = array.mean
3   min = array.min
4   if |(mean - min)/min| ≤ δ then
5     return true
6   else
7     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 { $blk, mean, count$ }

```
1 begin
2   n := len(T)
3   blk := T[0] // 新たなバケットを生成
4   count := count + 1 // blk に含まれる要素数
5   mean := T[0] // blk の平均値
6   for i := 1 → n - 1 do
7     if is_possible_to_add(T[i], δ) then
8       blk := blk ∪ T[i]
9       count := count + 1
10      mean := (count * mean + T[i]) / (count + 1) // 平均値を更新
11    if i == n - 1 then
12      return blk, mean, count
13    else
14      return blk, mean, count
15      new_block(T[i])
16      count := 1
17      mean := T[i]
```

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 を入力データとする。次に、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 で式 (4) 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。

$$p_n \leq (1 + \delta)x_1 \quad (5)$$

具体的に、数値データ {120, 130, 137, 140, 143, 146, 150, 155, 158, 161, 185, 190} と 0.2 の δ が与えられる場合、一番目のタプル 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 5 で判定する。十番目のタプル 161 までの平均値 144 が $120 * (1 + 0.2)$ より大きくなって、十一月番目のタプル 185 までの平均値 148 が 144 より大きいだから、{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 5 を満たすので、185, 190 をグループ 2 として保存される。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。グループ 1 について、最後に 144, 120, 161, 10 を取得できる。

BAQ に対して、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 単一の間合せにおける誤差保証

本章では、一つの間合せに対するシノプシスの構築方法について述べ、その誤差保証について議論する。さらに、構築したシノプシスをを用いた集約間合せ処理についても説明する。以下では、与えられる間合せの種類に応じて場合分けを行い、それぞれ記述する。なお、混同を避けるために、以下では元テーブル T 中の要素をタプル、シノプシス中の要素をレコードと記す。

3.1 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、つまりテーブル T 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む場合それぞれについて説明する。カテゴリ属性のみが用いられる場合、BAQ [1] と同様となるので省略する。

3.1.1 選択条件に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、以下のような間合せが対象となる。

```
SELECT status, COUNT(*)
FROM T
WHERE price > 100
GROUP BY status
```

この間合せに対し、対応するシノプシスのスキーマを (status, price_min, price_max, SF) とし、テーブル T 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。

ここで、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三つに分けられる。まず、1) のバケットについて、元の間合せを書き直して以下の間合せになって効率的に計算できる。

```
SELECT status, SF
FROM synopsis
WHERE price_min > 100
```

3) のバケットの場合、以下の間合せで対応できる。

```
SELECT status, SF
FROM synopsis
WHERE price_min < 100 AND price_max > 100
```

ここで、COUNT 関数を計算する時、 $\left\lceil SF * \frac{\text{price_max} - 100}{\text{price_max} - \text{price_min}} \right\rceil$ を計算し、選択条件を満たすレコードの数 SF として返す。

ただし、数値属性が選択条件に存在する場合、COUNT 集約の誤差を δ 以下に保証することはできない。これは、前述した条件を部分的に満たすバケットの中から条件を満たすタプル数を誤差保証付きで推定できないためである。つまり、誤差を δ 以内に収めるには部分的に条件を満たすバケットに対してのみ正確な COUNT を計算するなど、追加の対応が必要となる。なお、BAQ ではバケット内のタプルの分布が一様分布に従うと仮定し、部分的に条件を満たすバケット内のタプル数を推定している。

3.2 MIN/MAX 集約

MIN/MAX 集約における誤差保証について述べる。ただし、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。

3.2.1 選択・グルーピング条件がない場合

まず、間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、以下のような間合せが対象となる。

```
SELECT MIN(price)
FROM T
```

この間合せに対し、対応するシノプシスのスキーマを (price_min, price_max, SF) とし、テーブル T 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、以下の問合せで効率的に計算できて、相対誤差はゼロである。

```
SELECT MIN(price)
FROM synopsis
```

3.2.2 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 status, MIN(price)
FROM T
WHERE priority = 'URGENT'
GROUP BY status
```

このとき、対応するシノプシスのスキーマを ($status$, $priority$, $price_min$, $price_max$, SF) とし、テーブル T 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、誤差保証は以下の問合せで、条件を含まない場合と同様に行える。

```
SELECT MIN(price)
FROM synopsis
WHERE priority = 'URGENT'
```

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テーブル T 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、回答の相対誤差は 0 以内である。

3.2.3 選択条件に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 status, MIN(price)
FROM T
WHERE price > 100
GROUP BY status
```

この問合せに対し、スキーマ ($status$, $price_mean$, $price_min$, $price_max$, SF) を持つシノプシスを考え、テーブル T 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。

数値属性が選択条件に存在する場合においても、MIN 集約の回答の相対誤差は δ 以内に保証できる。まず、回答する際は選択条件によって処理対象のレコードを絞り込む。このとき、数値属性の条件によって部分的に条件を満たすレコードも処理対象に含める。その後、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、最小値の真値を持つレコード（上記問合せであれば最小の $price_min$ を持つレコード）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を含むとしても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、上記の手続きで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。

3.3 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。

3.3.1 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 T
```

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、テーブル T 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。

```
SELECT SUM(price_mean * SF)
FROM synopsis
```

シノプシスに対し、以上の問合せで $price_mean \times SF$ の総和を近似的な総和として返す。提案手法では、BAQ [1] 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、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 における誤差保証の導出を説明する。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_i 、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。

$$\text{err}(s, s') = \left| \frac{\sum_{i=1}^n v_i - \sum_{i=1}^k SF_i \cdot p_i}{\sum_{i=1}^n v_i} \right| \quad (6)$$

$$= \left| \frac{\sum_{i=1}^n v_i - \sum_{i=1}^k \sum_{j=1}^{SF_i} p_i}{\sum_{i=1}^n v_i} \right| \quad (7)$$

k 個のバケットは元々テーブル T 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。

$$\text{err}(s, s') = \left| \frac{\sum_{i=1}^k \sum_{j=1}^{SF_i} v_{ij} - \sum_{i=1}^k \sum_{j=1}^{SF_i} p_i}{\sum_{i=1}^k \sum_{j=1}^{SF_i} v_{ij}} \right| \quad (8)$$

$$= \left| \frac{\sum_{i=1}^k \sum_{j=1}^{SF_i} (v_{ij} - p_i)}{\sum_{i=1}^k \sum_{j=1}^{SF_i} v_{ij}} \right| \quad (9)$$

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。**

$$\text{err}(s, s') = \frac{\sum_{i=1}^k \sum_{j=1}^{SF_i} |v_{ij} - p_i|}{\sum_{i=1}^k \sum_{j=1}^{SF_i} |v_{ij}|} \quad (10)$$

$$\leq \frac{\sum_{i=1}^k \sum_{j=1}^{SF_i} \delta |v_{ij}|}{\sum_{i=1}^k \sum_{j=1}^{SF_i} |v_{ij}|} = \delta \quad (11)$$

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。

$$\text{err}(s, s') = \left| \frac{\sum_{i=1}^n v_i - \sum_{i=1}^k SF_i \cdot p_i}{\sum_{i=1}^n v_i} \right| \quad (12)$$

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

$$= \left| \frac{\sum_{i=1}^n v_i - \sum_{i=1}^k \sum_{j=1}^{SF_i} v_{ij}}{\sum_{i=1}^n v_i} \right| \quad (14)$$

$$= \left| \frac{\sum_{i=1}^n v_i - \sum_{i=1}^n v_i}{\sum_{i=1}^n v_i} \right| = 0 \quad (15)$$

3.3.2 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 status, SUM(price)
FROM T
WHERE priority = 'URGENT'
GROUP BY status
```

このとき、対応するシノプシスのスキーマを (status , priority , price_mean , price_min , price_max , SF) とし、テーブル T 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、以下の問合せで 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。

```
SELECT SUM(price_mean*SF)
FROM synopsis
WHERE priority = 'URGENT'
```

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テーブル T 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 3.3.1 節と同様の議論ができ、回答の相対誤差がゼロであることを導ける。

3.3.3 選択条件に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 status, SUM(price)
FROM T
WHERE price > 100
GROUP BY status
```

この問合せに対し、スキーマ (status , price_mean , price_min , price_max , SF) を持つシノプシスを考え、テーブル T 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、以下の二つの問合せで結果を近似結果として返す。

全体のレコードが条件を満たすバケットについて、以下の問合せで効率的に SUM 関数を計算できる。

```
SELECT status, SUM(price_mean*SF)
FROM synopsis
WHERE price_min > 100
```

部分的に条件を満たすバケットの場合、元の問合せが以下の問合せで計算する。

```
SELECT status, SUM(price_mean*SF)
FROM synopsis
WHERE price_min < 100 AND price_max > 100
```

SUM 集約の場合、COUNT 集約と同様に、部分的に条件を満たすバケットの影響で単純には誤差を保証できない。しかし、部分的に条件を満たすバケットに対しそのバケット中で条件を満たすタプルの正確な数を計算することで、SUM 集約の近似的な計算が可能となる。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に生成することで、部分的に条件を満たすバケットの合計は 7 から式 (11) に示した議論のとおり誤差 δ 以内で計算できる。

ただし、誤差保証が可能なのは、**部分的に条件を満たすバケットが集約対象の数値属性上で正ないし負の範囲にのみ存在するとき**である。これは、条件を部分的に満たすバケットがそれぞれ負の側と正の側とに存在するとき、それらの合計が互いに打ち消し合う可能性があるためである。このとき、一般的に相対誤差が δ 以内になることは示せず、厳密な値を計算するなど誤差を保証するには他の方法を取る必要がある。

4 統合シノプシスの構築

オンラインで効率的に問合せを処理できるように、最も単純な考え方は問合せごとにシノプシスを作成し、複数のシノプシスでオンライン的に問合せを回答することである。ただし、複数のシノプシスを保存する必要がある、加えて、処理速度が遅い可能性も高い。

保存されるシノプシスのサイズを最小化するために冗長なシノプシスを削除する方法については、4.1 節で説明する。後、残ったシノプシスに基づく最終的なシノプシスの構築手法を4.2 節で述べる。

混同を避けるために、以下では元テーブル T 中の要素をタプル、シノプシス中の要素をレコードと記す。

4.1 シノプシスサイズの最小化

3 章で述べた、複数のシノプシスの組合せに基づくアプローチでは、シノプシスのサイズが大きくなり、時間もかかるという問題が存在する。ここで、複数の問合せのシノプシスの間の関連を用いた最小化操作を行う。

具体的には、問合せ i について関連列セット Q_i を持つシノプシス s_i があり、問合せ j に関連列セット Q_j を持つシノプシス s_j があるとする。ここで、 $Q_i \subseteq Q_j$ の場合、シノプシス s_j で問合せ i も回答できると判断する。

ここで、3 章の問合せを例として説明する。問合せ 1 から問合せ 7 について、

- (status, price_min, price_max)
- (price_min, price_max)
- (status, priority, price_min, price_max)
- (status, price_mean, price_min, price_max)
- (price_mean, price_min, price_max)
- (status, priority, price_mean, price_min, price_max)
- (status, price_mean, price_min, price_max)

というシノプシスを生成した。問合せ 7 のシノプシスが他のシノプシスの列を全部含むことから、シノプシス 6 で他の問合せも処理できる。これにより、問合せ 1 から 7 まで与えられる場合、シノプシス 6 のみ保存して問合せを処理できる。

4.2 シノプシスの構築

BAQ [1] で述べられたとおり、最小化されたシノプシス集合に基づいて、元データをスキャンし、シノプシス集合のレコードをカバーするタプルを選択し、統合シノプシスとするアイデアで行える。

まず、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ソートしたデータを T とする。シノプシス集合の全体レコードを持つ集合 S とソートされたデータ T を入力データとする。次に、すべてのレコードの状態 *record.flag*、統合シノプシス S' を初期化する。全レコードの状態に false を、統合シノプシス S' に空集合をそれぞれ代入する。続いて、最小化されたシノプシス集合の全体レコードについて順番に元データ T にマッチングできる 1 番目のタプルの選択を行う。具体

的には、カテゴリ属性の値が同じで、数値属性の値がレコードの $[min, max]$ の範囲に含まれるタプルを選択し、統合シノプシスのレコードとして保存する。後、ソート済みデータ T から、タプルがシノプシスのレコード *record* とマッチングするかをチェックしながらループ処理を行う。マッチングできた場合、ループではレコードの状態 *record.flag* とシノプシス S' の更新を行う。シノプシスのレコードにすべてマッチングを行ってループ処理を終了する。スキーマ T に N 個のレコードがあり、シノプシス集合に全体的に M 個のタプルがある場合、時間の複雑さは $O(MN)$ である。

5 実験と評価

本章では、提案手法に関する評価を行う。他の代表的な AQP 手法に対し、相対誤差、生成したシノプシスのサイズ、オンライン問合せ処理時間、およびオフラインのシノプシス生成時間の 4 点で比較を行う。

5.1 準備

5.1.1 データセット

本研究では二つのデータセットを使用した。一つ目は 7 列（6 つのカテゴリ列と 1 つの数値列）を持つ Kaggle の temperature データセットである。このデータセットには 321 個都市の 36 年間の気温の統計が含まれている。二つ目はよく知られている TPC-H データセットであり、実験では 4 列（3 つのカテゴリ列と 1 つの数値列）の orders テーブルを使用した。

5.1.2 問合せ

与えられる問合せ集合について、各データセットについて 5 節の各種類による 310 個の問合せを生成した。これには、選択と条件がない MIN/MAX 問合せ 2 個、SUM/AVG 問合せ 2 個、条件がカテゴリ属性のみの COUNT 問合せ 50 個、MIN/MAX 問合せ 50 個、SUM/AVG 問合せ 50 個、選択条件に数値属性を含む COUNT 問合せ 52 個、MIN/MAX 問合せ 52 個、SUM/AVG 問合せ 52 個である。以上の 310 個の問合せ集合によりオフラインにシノプシスを生成して、シノプシスのサイズと生成時間を評価できる。

後、オンラインに問合せの処理時間と相対誤差を比較しよう。310 個の問合せの属性セットに基づいて、選択と条件がない MIN/MAX 問合せ 2 個、SUM/AVG 問合せ 2 個、条件がカテゴリ属性のみの COUNT 問合せ 20 個、MIN/MAX 問合せ 20 個、SUM/AVG 問合せ 20 個、選択条件に数値属性を含む COUNT 問合せ 5 個、MIN/MAX 問合せ 10 個、SUM/AVG 問合せ 10 個を生成して、既存研究の BAQ と比較した。BAQ は最新の SAQP (Blink, Seek), Sketch, Wavelet よりも優れていることが証明されているので [1], SAQP 系の手法とは比較しなかった。

5.1.3 評価項目

提案手法と BAQ を四つの角度から比較する。(1) 推定の結果と真値の間の誤差、(2) シノプシスのサイズ、(3) オンラインに問合せを処理する時間 (ms)、(4) オフラインのシノプシスの生成時間 (s)。評価実験では、提案手法と BAQ をそれぞれ 20 回実行し、平均値を結果とする。

5.2 誤差限界

誤差限界 δ を 0.05, 0.1, 0.15, 0.2, 0.25 を設定して, 提案手法と BAQ を比較する.

5.2.1 推定誤差

数値属性に正の数のみを持っている orders および正と負の数の両方を持っている temperature データについて, 全体問合せの平均値について提案手法と BAQ の誤差計算の結果は図 1 のようになる.

提案手法の相対誤差ははるかに小さく, orders データを利用する時提案手法が BAQ の 10-20 %, temperature の場合に BAQ の 10-30 % となった. 例えば, temperature データの場合誤差範囲が 0.05 から 0.25 の間では, 提案手法の誤差は 0.002-0.01 であるが, BAQ の誤差は 0.01-0.05 である. その理由は, 提案手法は選択条件に数値属性を含まない場合, 100 % の信頼度を持っているためである. 誤差限界の増加に伴い, 推定誤差も増加するが, 限界 δ は最悪の推定に対応するため, 実際の推定誤差は常に指定された誤差限界よりも小さくなり, 実際にははるかに小さくなっている.

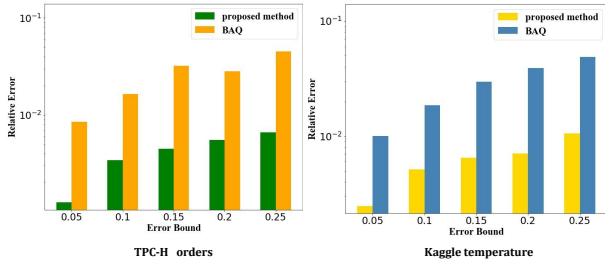


図 1 全体問合せの誤差

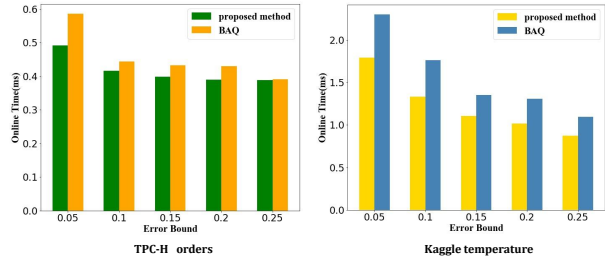


図 2 オンラインに問合せの処理時間

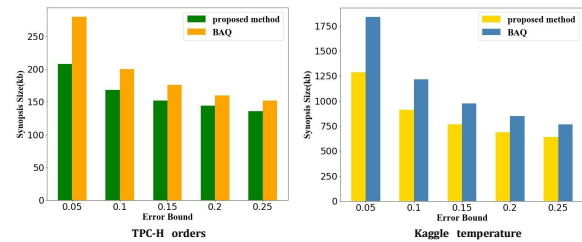


図 3 シノプシスのサイズ

5.2.2 シノプシスのサイズ

提案手法と BAQ で生成したシノプシスのサイズを図 3 に示す.

orders データについて提案手法のシノプシスのサイズは BAQ に対して 60-90 %, temperature の場合が 50-80 % となっている.

ここで簡単な分析を行う. BAQ は最小値を基準として, 任意値の値を代表値として, 式 (3) でグルーピングしてシノプシスを生成する. 提案手法は正及び負の最小値を基準として, 平均値を代表値として, 数値属性についてグルーピングして, 各バケットについて得られる. 各バケットのレコード数は, 提案手法の方が BAQ よりも大きくなり, シノプシスのサイズが小さくなる. さらに, 誤差の限界を増加させると, δ を満たすために生成するバケットが少なくなるため, シノプシスのサイズが減少する.

5.2.3 オンライン問合せ処理時間

orders と temperature データについて, オンラインに問合せを処理する時間を計測した. 提案手法と既存研究 BAQ の結果を図 2 に示す.

提案手法のシノプシスのサイズは BAQ よりも小さかったため, 提案手法は BAQ よりも高速であって, orders の場合が 0.5 ミリ秒以内, temperature の場合が 1.8 ミリ秒以内に問合せに答えることができた. 誤差限界の増加に伴い, 問合せ処理の時間は減少した. 誤差限界が大きくなったことで, シノプシスも小さくなったので, シノプシスをスキャンする時間が短くなったためである.

5.2.4 シノプシス生成時間

オフラインにシノプシスを生成する時間について, orders と temperature データセットに対する計測結果を図 4 に示す.

orders データについて, 提案手法は BAQ より最大 2 倍の時間を要した. δ が 0.25 の場合, 提案手法は 3800 秒が必要があるに対して, BAQ は 2000 秒である. しかし, temperature については提案手法と BAQ 間の差は小さくなり, いずれも 800s 以内で処理できる.

提案手法は各バケットの平均値を計算する必要があるので, 既存研究より時間を要した. シノプシスの生成はオフラインで実行するので, ユーザに対する影響は小さいと考えられる.

5.3 選択条件を持つ問合せ

推定誤差について, 問合せの種類ごとの詳しい分析の必要があるため, 本節では, 各集計関数について, 選択条件に数値属性を含む場合と含まない場合について深く分析する.

WHERE 句の選択条件に数値属性がある問合せと選択条件に数値属性がない問合せに分けて考える. ここで, PM_COUNT は提案手法で数値属性に対する選択条件がない場合, PM_COUNT_Num は数値属性に対する選択条件がある場合を表す. BAQ についても同様である.

まず, 二つのデータセットに対し COUNT 関数を考えたときの, 提案手法と BAQ の推定誤差は図 5 のようになる.

以下の観察が行える. 提案手法と BAQ の両者とも選択・グルーピング条件がない場合と選択条件にカテゴリ属性のみがあ

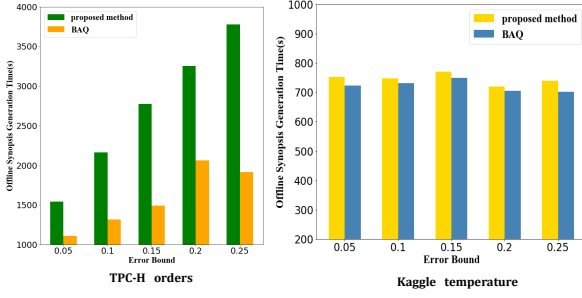


図4 シノプシスの生成時間

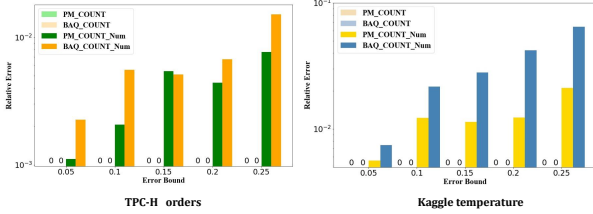


図5 選択問合せの誤差 (COUNT 関数の場合)

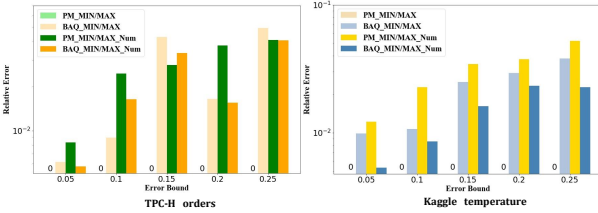


図6 選択問合せの誤差 (MIN/MAX 関数の場合)

る場合、100 %の信頼度となる。選択条件に数値属性を含む場合、提案手法が BAQ より大きいことがあるが、誤差限界以内を保証できる。

次、MIN/MAX 集約計算する場合の推定誤差の評価結果を図6に示す。この場合、数値属性が選択条件に含まれていないなら、誤差はゼロと保証されている。しかし、無視できない推定誤差が観測できる。選択条件に数値属性を含む場合、提案手法の方が BAQ に劣っている。ただし、提案手法は誤差限界以内には収まっている。

最後、SUM と AVG 関数を含む問合せに対する結果を図7に示す。提案手法では、条件がない場合、誤差0を保証できて、WHERE 句選択条件にカテゴリ属性がある、数値属性に関する集計結果を返す場合、理論的には誤差がゼロであるが、実装では実数を浮動小数点で表現したときの丸め誤差により小さな誤差が現れる。BAQ の場合、推定誤差が提案手法より大きくなった。選択条件に数値属性を含む場合には、提案手法は BAQ より小さい誤差となる。

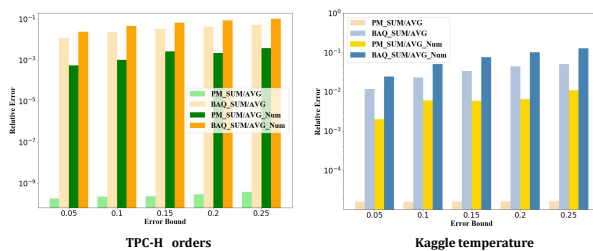


図7 選択問合せの誤差 (SUM/AVG 関数の場合)

以上より、提案手法は 0.05 から 0.25 までの誤差限界の範囲において推定誤差、シノプシスのサイズおよびオンライン問合せ処理時間の点で既存研究を上回った。

6 おわりに

本稿では、近似的問合せ処理において、誤差上限以内でシノプシスを生成し、効率的にオンライン問合せ処理を行うアプローチを提案した。また、上記の誤差保証を持つ近似的問合せ処理フレームワークを実装し、二つのデータセットを用いて評価実験を行った。実験結果により、提案手法は既存研究 BAQ より推定誤差、オンラインの問合せ処理時間、シノプシスのサイズにおいて優れており、実数データにも誤差以内で効率的に適用できることを示した。

今後は、二つ以上の数値属性を持つ複雑な問合せを効率的に処理するための近似的問合せ処理方法、及びまとめシノプシスの最適化について検討する予定である。

謝 辞

本研究は JSPS 科研費 (16H01722, 19K21530, 20K19804) および文部科学省委託事業「地球環境情報プラットフォーム構築推進プログラム (DIAS)」の助成による。

文 献

- [1] K. Li, Y. Zhang, G. Li, W. Tao, and Y. Yan, "Bounded approximate query processing," *IEEE TKDE*, vol. 31, no. 12, pp. 2262–2276, 2019.
- [2] S. Chaudhuri, B. Ding, and S. Kandula, "Approximate query processing: No silver bullet," in *Proc. SIGMOD*, pp. 511–519, 2017.
- [3] B. Mozafari and N. Niu, "A handbook for building an approximate query engine," *IEEE Data Engineering Bulletin*, vol. 38, no. 3, pp. 3–29, 2015.
- [4] K. Li and G. Li, "Approximate query processing: What is new and where to go?," *Data Science and Engineering*, vol. 3, pp. 379–397, 2018.
- [5] K. Zeng, S. Gao, J. Gu, B. Mozafari, and C. Zaniolo, "ABS: A system for scalable approximate queries with accuracy guarantees," in *Proc. SIGMOD*, pp. 1067–1070, 2014.
- [6] A. Pol and C. Jermaine, "Relational confidence bounds are easy with the bootstrap," in *Proc. SIGMOD*, pp. 587–598, 2005.
- [7] S. Agarwal, H. Milner, A. Kleiner, A. Talwalkar, M. I. Jordan, S. Madden, B. Mozafari, and I. Stoica, "Knowing when you're wrong: Building fast and reliable approximate query processing systems," in *Proc. SIGMOD*, pp. 481–492, 2014.
- [8] Q. Ma and P. Triantafillou, "DBEST: Revisiting approximate query processing engines with machine learning models," in *Proc. SIGMOD*, pp. 1553–1570, 2019.
- [9] B. Walenz, S. Sintor, S. Roy, and J. Yang, "Learning to sample: Counting with complex queries," *PVLDB*, vol. 13, no. 3, pp. 389–401, 2019.