

シノプシスの最適化に基づく近似問合せ処理の高速化

倪 天嘉[†] 杉浦 健人[†] 石川 佳治[†] 陸 可鏡[†]

[†] 名古屋大学大学院情報学研究科

Email: {ni,lu}@db.is.i.nagoya-u.ac.jp, {sugiura,ishikawa}@i.nagoya-u.ac.jp

あらまし 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、近似的問合せ処理 (approximate query processing, AQP) が着目されている。AQP では対象データを要約した構造 (シノプシス) やサンプルを用いた問合せ処理が行われ、処理結果は一般に誤差を含む。既存研究で提案された bounded approximate query (BAQ) フレームワークではシノプシスを用いた近似処理において厳密な誤差保証を提供したが、BAQ には正のデータのみに対する誤差しか保証できないという問題がある。加えて、BAQ で生成したシノプシスのサイズは元データより小さくなるが、より小さなシノプシスを取得するためにシノプシスの生成アルゴリズムには改善の余地がある。本稿ではこれらの問題に対応すると共に、メモリに格納されるシノプシスの最小化に焦点を当て BAQ を拡張する。TPC-H ベンチマークにおける主要な問合せを用いた実験により BAQ と比較し、提案手法 BAQ_± の有効性を示す。

キーワード 近似的問合せ処理, シノプシス, 問合せ処理・問合せ言語

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。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]。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, 5]。AQP からのシノプシスの問合せ結果には誤差が含まれるため、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。

Li ら [6] は一部のデータを用いた問合せ処理において誤差保証を行うために、誤差保証付きの近似問合せフレームワークとして bounded approximate query (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) [7–10] 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。

ただし、BAQ では、正の数値データのみに対する誤差しか保証できない。また [6] では、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 で生成したシノプシスのサイズは元データより小さくなるが、より小さなシノプシスを取得するためにシノプシスの生成アルゴリズムには改善の余地が

ある。加えて、BAQ で対応できるのは構造が簡単な問合せのみであり、例えば複数の数値属性に関する集約計算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 には複数の数値属性を含む集約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 BAQ_± を提案する。加えて、対応可能な問合せを対象に、メモリに格納されるシノプシスを最小化するためにワークロードの最適化とシノプシスの生成アルゴリズムについて検討する。

2 既存研究

Li ら [6] はシノプシスに基づき近似問合せの誤差を厳密に保証するフレームワークとして bounded approximate query (BAQ) を提案した。BAQ の処理は 1) サンプルとして与えられた問合せ集合からシノプシスを構築するオフライン処理、および 2) 構築したシノプシスを用いてユーザからの問合せに答えるオンライン処理の 2 段階に分かれる。BAQ の特徴は誤差の指標として相対誤差を用いたことであり、互いの相対誤差がユーザに与えられたしきい値以下となるようレコードをバケットに分割することで、各集約問合せに対して厳密な誤差保証を可能としている。以下、2.1 節では BAQ における誤差保証の基礎となるバケット分割、2.2 節ではオフラインでのシノプシス構築および 2.3 節ではシノプシスを用いたオンラインでの問合せ処理についてそれぞれ述べる。

2.1 数値属性のバケット分割

まず、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。BAQ では与えられた問合せに対し、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. 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 2 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 [6] で述べられたとおり, 各数値属性の正の最小値を基準値とした方法で行える. 例えば, 正のゼロでない最小値 x_{\min} を基準としたとき, 以下の式を満たす x を x_{\min} と同一のバケットに含める.

$$x \leq (1 + \delta)x_{\min} \quad (3)$$

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) の基準値として式 (3) で判定する. 5 番目のタプル 143 までは $120 \cdot (1 + 0.2)$ より小さく, 6 番目のタプル 146 からは 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. 次に, 146 をグループ 2 の基準値として 161 までは式 (3) を満たすため $\{146, 150, 155, 158, 161\}$ をグループ 2 として保存する. それにより 3 つのグループが得られ, 各グループの先頭の値, 最小値, 最大値および要素数をバケットとして保存する. 例えばグループ 1 の場合, 最後に $\{120, 120, 143, 5\}$ となる.

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.

2.2 シノプシスの生成

BAQ は 1 つの数値属性と複数のカテゴリ属性に対し選択やグルーピングを含む問合せをサポートする. まず, シノプシスを初期化し, 元データにカテゴリ属性のそれぞれの一意の値と対応する数値属性のデータ集合を直積で組み合わせる. ここで, カテゴリ属性の一意の値の集合を V で表し, 相関する数値属性のデータ集合を G で表す. では直積を示すペア (V, G) を生成できる. 次に, ペアにカテゴリ値 $v \in V$ による数値属性のデータ集合 $G(v)$ について式 (3) でバケットを計算し $(v, G(v)_{\text{head}}, G(v)_{\text{min}}, G(v)_{\text{max}}, \text{count})$ で表す. $\text{count}(v, G(v)) > 0$ の場合 $(v, G(v)_{\text{head}}, G(v)_{\text{min}}, G(v)_{\text{max}}, \text{count})$ をシノプシスに追加する. カテゴリ属性の一意の値を全部処理すると, シノプシスの生成が終わる.

表 1 のような商品情報を登録する販売商品表を例として説明する. 会社の管理者は, 1000 以上収益がある商品分類, 商品名と商品数を知りたい場合, 以下の問合せを実行する.

表 1 販売商品表 product

productID	name	category	price	revenue
1	jacket	clothes	14350	1330
2	shirt	clothes	2450	998
3	jacket	clothes	23450	3120
4	ballpoint pen	stationery	2400	1320
5	jacket	clothes	16320	1370
6	fork	kitchenware	1230	234
7	jacket	clothes	16320	1300
8	shirt	clothes	2670	1002

```
SELECT name, category, COUNT(*)
FROM product
WHERE revenue > 1000
GROUP BY name, category
```

対象データとして product の (name, category, revenue) 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 name, category により対象元データをグルーピングする. つまり, name と category の値の各組合せに基づいてグルーピングを行って $\{(jacket, clothes), (shirt, clothes), \dots, (fork, kitchenware)\}$ のカテゴリ値グループ集合を得る. 次に, 各グループについて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 $(name, category)$ の各値の組合せに対し対応する revenue 属性値の集合について式 (3) でバケットを計算し, シノプシス (name, category, revenue_head, revenue_min, revenue_max, count) を得る. 具体的に $(name, category)$ は (jacket, clothes) の場合, 対応する revenue の 3 個の数値データについて式 (3) でバケットを計算し $\{(jacket, clothes, 1300, 1300, 1300, 1), (jacket, clothes, 1330, 1330, 1330, 1), (jacket, clothes, 1370, 1370, 1370, 1)\}$ のレコード集合を得る. 各レコード集合がシノプシスのレコードとして保存される.

2.3 問合せ処理

ここでは, BAQ における問合せ処理を, 2.2 節の問合せを例として説明する. 前節で示した (name, category, revenue_head, revenue_min, revenue_max, count) のシノプシスに基づく, 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, BAQ ではバケットの最小値が 1000 より大きいという条件のもとで $SUM(count)$ を計算する. 3) のバケットの場合, $\left\lfloor count \times \frac{revenue_max - 1000}{revenue_max - revenue_min} \right\rfloor$ を計算し, 選択条件を満たすレコードの数 $count_0$ として返す. BAQ の性質により, この集約結果には δ 以内の誤差保証がある.

ただし, 問合せの選択条件に数値データが存在する場合, COUNT 集約の誤差を δ 以下に保証できない. 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て、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 提案手法 BAQ±

ここでは、[6] をもとに拡張した提案手法の対象問合せおよびアイデアを示す。

3.1 対象問合せ

BAQ± 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 =, ≠ と集約計算
- 一つ以上の数値属性に対する =, ≠, >, ≥, <, および ≤ を用いた条件での選択と集約計算
- グルーピング

3.2 数値属性のバケット分割

BAQ± 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、BAQ とは異なり、式 (4) で誤差を評価する。

$$\bar{x} = \frac{\sum_{x \in B} x}{|B|} \quad (4)$$

$$\forall x \in B, \text{err}(x, \bar{x}) \leq \delta$$

バケット生成処理をアルゴリズム 1 に示す。

Algorithm 1: GenerateBuckets(T, δ)

Data: 昇順ソート済のタプル配列 T , 誤差の閾値 δ

Result: バケット集合 S

```

1  $B := \emptyset, \text{count} := 0$ 
2  $S := \emptyset$ 
3 for  $i := 0$  to  $\text{len}(T) - 1$  do
4   if  $\text{IsPossibleToAdd}(B, T[i], \delta) = \text{false}$  then
5      $S := S \cup \{B\}$ 
6      $B := \emptyset, \text{count} := 0$ 
7    $B := B \cup \{T[i]\}$ 
8    $\text{mean} := (\text{count} \times \text{mean} + T[i]) / (\text{count} + 1)$ 
9    $\text{count} := \text{count} + 1$ 
10  $S := S \cup \{B\}$ 
11 return  $S$ 
```

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 (タプルの配列) 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 を入力データとし、バケット B 、バケット集合 S および要素数 count を初期化する (1–2 行目)。続いて、順番に元データ T のタプルをアルゴリズム 2 により、式 (4) に基づいてバケットを生成する (3–9 行目)。具体的には、バケットへの分割は、各数値属性の正と負の最値を基準として行える。

Algorithm 2: IsPossibleToAdd($B, \text{count}, \text{mean}, v, \delta$)

Data: バケット B , 追加したい値 v , しきい値 δ

Result: v を B に追加可能なら true , そうでなければ false

```

1 if  $B \neq \emptyset$  then
2    $\text{mean} := (\text{count} \times \text{mean} + v) / (\text{count} + 1)$  //  $v$  を追加後の平均を求める
3   if  $(\text{err}(\min(B), \text{mean}) > \delta)$  or  $(\text{err}(v, \text{mean}) > \delta)$  then
4     // 最小値  $\min(B)$ , 最大値  $v$  のいずれかが誤差閾値を超えた
5     return  $\text{false}$ 
6 return  $\text{true}$  //  $B = \emptyset$  の場合も  $\text{true}$  を返す
```

例として、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 から式 (4) でバケット 1 への所属を判定する。10 番目のタプル 161 までの平均値 144 は最小値 $120 \cdot (1 + 0.2)$ より小さく、最大値 $161 \cdot (1 - 0.2)$ より大きいことから、バケット 1 に追加する。11 番目のタプル 185 までの場合、平均値 147.7 は最小値 $120 \cdot (1 + 0.2)$ より大きいことから、式 (4) にしたがってバケット 1 に追加せず、バケット 2 の最小値として判定する。つまり、10 番目までのタプル { 120, 130, 137, 140, 143, 146, 150, 155, 158, 161 } をバケット 1 として保存される。185 から 190 までが式 (4) を満たすので { 185, 190 } をバケット 2 として保存する。このように生成した各バケットの平均値、最小値、最大値および要素数をレコードとして保存する。例えばバケット 1 については (144, 120, 161, 10) が得られる。

BAQ と比較して、BAQ± ではバケットの対応するデータの範囲はより大きくなる。加えて、BAQ± では 2 つの数値属性に対する選択やグルーピングを含む問合せをサポートする。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。

3.3 シノプシスの生成

1 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。

```

SELECT name, category, MIN(revenue)
FROM product
WHERE revenue > 10000
AND price < 200000
GROUP BY name, category
```

以上の問合せを例として、対象データ T として product の (name, category, price, revenue) のインスタンスに関するレコードをシノプシスの処理対象とする。まず、カテゴリ属性によりデータをグルーピングする。name と category の属性値の組合せに基づいてグルーピングを行う。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、name の各値に対し、revenue について 3.2 節のアルゴリズム 1 と 2 でバケットを計算して (name, revenue_mean, revenue_max, revenue_min, count1) のシノプシスを得る。数値属性 category と数値属性 price による (category, price_mean, price_max, price_min,

count2) のスキーマに関してもシノプシスを求める。最後に、2つのシノプシスを元テーブルと組み合わせて最後のシノプシス (name, category, revenue_mean, revenue_max, revenue_min, count1, price_mean, price_max, price_min, count2) を生成する。シノプシス中には count1 = 0, count2 = 0 となるレコードは含まない。

3.4 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 (name, category, revenue_min, revenue_max, count1) のシノプシスを生成することを考える。シノプシス中には count1 = 0 となるレコードは含まない。2.1 節の product を使う以下の問合せを例として説明する。

```
SELECT name, category, COUNT(*)
FROM product
WHERE revenue > 1000
GROUP BY name, category
```

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。ここでは数値属性 revenue に関する全体が 1000 より大きいバケットが 6 つ、一部が条件を満たすバケットが 1 つある。1) のデータ全体が条件を満たすバケットについては、元の問合せを以下のように書き直して効率的に計算する。

```
SELECT name, category, SUM(count1)
FROM synopsis
WHERE revenue_min > 1000
GROUP BY name, category
```

3) のデータの一部が条件を満たすバケットの場合、バケットの最値を取得する必要があるため、ここではシノプシスから最大値 revenue_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をする。

```
SELECT name, category, COUNT(*)
FROM product
WHERE revenue > 1000
AND revenue < synopsis.revenue_max
GROUP BY name, category
```

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することで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 1 つ以上の数値属性に対する選択・グルーピングを含む問合せについて、[11] の分析により提案手法は BAQ より小さい誤差となる。

4 保存されるシノプシスの最小化

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り、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、4.1 節で説明する。

4.1 ワークロードの最適化

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、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ワークロードの最小化を操作する。

具体的には、問合せ i について関連する属性 (ここでは「関連列」で表す) のセット 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 があるとすると、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、以下の関連列セットの集合 Q_1, Q_2, Q_3 を例として説明する。

- Q_1 : name, price
- Q_2 : name, category
- Q_3 : name, price, revenue

Q_3 が Q_1 を含むことから、 Q_3 を含むシノプシス 3 で Q_1 を含むシノプシス 1 を処理できる。これにより、ユーザから Q_1, Q_2, Q_3 に関するワークロードが与えられる場合、問合せ 2 と 3 のみ保持して、メモリに最適なシノプシスを生成できる。

5 実験と評価

本章では、提案手法 BAQ± に関する評価する。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 点で比較する。

5.1 準備

5.1.1 データセット

本研究では 2 つのデータセットを使用した。1 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 280 万レコードがあるデータセットには、321 個都市の 36 年間の気温の統計が含まれている。2 つ目はよく知られている TPC-H ベンチマークからの 900 万のレコードを含むデータセットであり、5.2 節と 5.3 節の実験では 4 列 (3 つのカテゴリ列と 1 つの数値列) の orders テーブルを使用する。加えて、複雑な問合せへの対応を考えて、TPC-H ベンチマークの全体データと [12] の問合せを使用して BAQ± と BAQ を比較する。5.4 節で詳しく説明する。

5.1.2 問合せ

実験のため、まず、各データセットについて 310 個の問合せを生成した。これには、選択条件がない 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する。

次いで, オンラインに問合せの処理時間と相対誤差を比較するため, 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 BAQ と比較した。

複雑な問合せの対応問題を考えて, ここで [12] の 1, 3, 5 と 6 番目の定番問合せを利用して BAQ_± を評価した。具体的に, 以下のような特定の処理状態にすべての品目の合計価格, 平均合計価格および平均割引の合計をリストして, 1 つ以上の数値属性に対する SUM と AVG 集約計算を含む問合せに関して評価実験を行った。

```
SELECT l_returnflag,
       l_linestatus,
       sum(l_extendedprice) as sum_base_price,
       avg(l_extendedprice) as avg_price,
       avg(l_discount) as avg_disc,
       count(*) as count_order
FROM lineitem
WHERE l_linestatus = 'URGENT'
```

GROUP BY l_returnflag, l_linestatus
実験では, 複数のテーブル間の結合操作について, 事前に 1 つのテーブルにまとめて実験する。BAQ は最新の SAQP (Blink, Seek), Sketch, Wavelet よりも優れていることが実証されているので [6], SAQP 系の手法とは比較しなかった。

5.1.3 評価項目

BAQ_± と BAQ を 4 つの角度から比較する: (1) 推定の結果と真値間の誤差 (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 (ミリ秒) (4) オフラインのシノプシスの生成時間 (秒)。評価実験では, BAQ_± 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, BAQ_± 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 BAQ_± と BAQ の誤差計算の結果は図 1 の上図のようになる。

BAQ_± の相対誤差ははるかに小さく, orders データを利用するとき BAQ_± は 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 BAQ_± 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, BAQ_± 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 は最悪の推定を対応するために実際の推定誤差は指定された誤差限界よりも小さくなる。

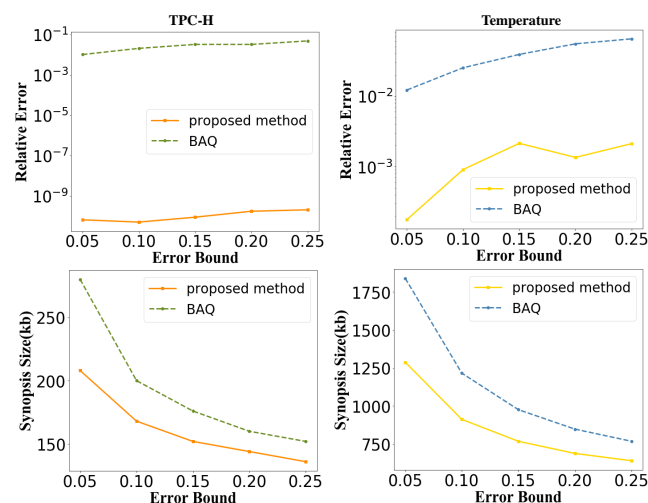


図 1 全体問合せの誤差とシノプシスのサイズ

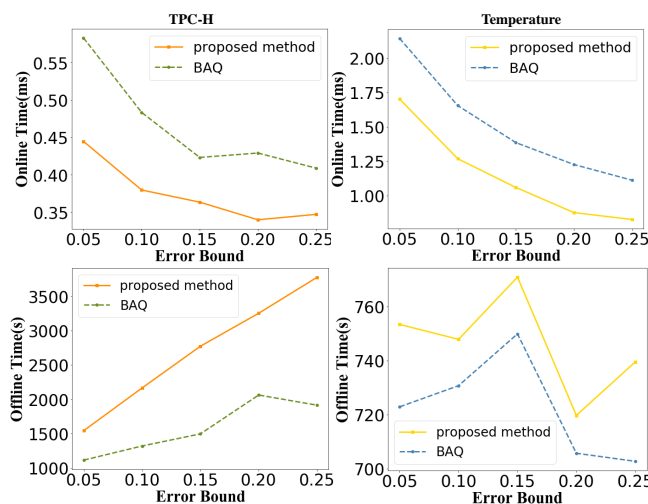


図 2 オンライン問合せの処理時間とオフラインの生成時間

5.2.2 シノプシスのサイズ

BAQ_± と BAQ で生成したシノプシスのサイズを図 1 の下図に示す。

orders データについて BAQ_± 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する。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。BAQ_± は正およ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求める。各バケットのレコード数は, BAQ_± 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、オンラインで問合せを処理する時間を計測した。BAQ± と既存研究 BAQ の結果を図 2 の上図に示す。

BAQ± のシノプシスのサイズは BAQ よりも小さかったため、BAQ± は BAQ よりも高速であり、orders の場合が 0.5 ミリ秒以内、temperature の場合が 1.8 ミリ秒以内で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ため、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 2 の下図に示す。

orders データについて、BAQ± は BAQ より最大 2 倍の時間を要した。 δ が 0.25 の場合、BAQ± は 3800 秒があるのに対して、BAQ は 2000 秒である。しかし、temperature については BAQ± と BAQ 間の差は小さくなり、いずれも 800 秒以内で処理できる。

BAQ± は各バケットの平均値を計算する必要があるため、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ため、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を深く分析する。

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 BAQ± 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、2 つのデータセットに対し COUNT 関数を考えたときの、BAQ± と BAQ の推定誤差は図 3 のようになる。

以下の観察が行える。BAQ± 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、BAQ± 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。

次に、MIN/MAX 集約計算をする場合の推定誤差の評価結果を図 4 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、BAQ± の方が BAQ に劣っている。ただし、BAQ± は誤差限界以内には収まっている。

最後に、SUM と AVG 関数を含む問合せに対する結果を図 5 に示す。BAQ± では条件がない場合誤差 0 を保証できて WHERE 句の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 BAQ± より大きくなった。選択条件に数値属性を含む場合には、BAQ± は BAQ より小さい誤差となる。

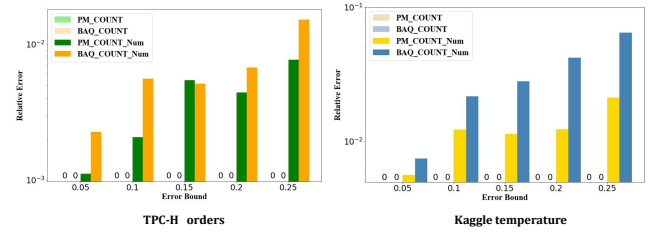


図 3 選択問合せの誤差 (COUNT 関数の場合)

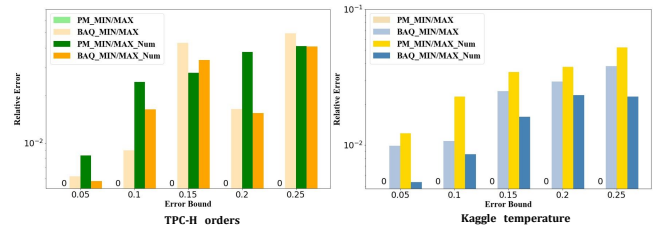


図 4 選択問合せの誤差 (MIN/MAX 関数の場合)

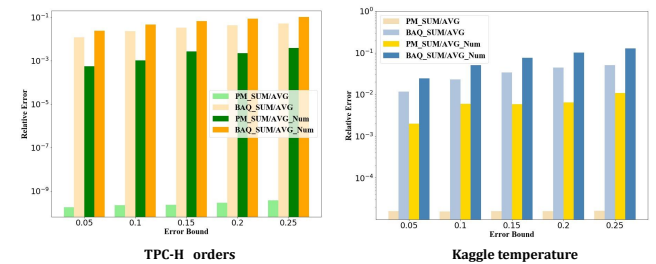


図 5 選択問合せの誤差 (SUM/AVG 関数の場合)

以上より、BAQ± 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。

5.4 複雑な問合せについて

問合せについて、複数の集約関数を含む問合せの分析の必要があるため、本節では、1 つ以上の数値属性に対する SUM と AVG 集約関数を計算する TPC-H の 1, 3, 5 と 6 番目の定番問合せ [12] について深く分析する。TPC-H のデータセットに対し集約関数を考えたときの、BAQ± と BAQ の評価結果は図 6 のようになる。

以下の観察が行える。まず、BAQ± の相対誤差は、BAQ の 30%まで小さくなった。誤差限界の増加に伴い、推定誤差も増加するが、限界 δ は最悪の推定を対応するために実際の推定誤差が指定された誤差限界よりも小さくなる。実際にははるかに小さくなっている。

次に、シノプシスのサイズについて、BAQ± の結果は BAQ に対して 60–80%となっている。5.2.2 節と同じように、BAQ± は各数値属性のバケットの平均値を代表値として、BAQ より大きなバケットを取得できる。つまり、小さいシノプシスを生成

できる．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．

また，オンラインに問合せを処理する場合，BAQ \pm のシノプシスのサイズは BAQ よりも小さかったため，BAQ \pm は BAQ よりも高速であって，900 万のレコードを含む TPC-H からのデータセットに対して 80 ミリ秒以内で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ため，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．

最後に，シノプシスの生成する場合，BAQ \pm は BAQ より最大 1.05 倍の時間を要した． δ が 0.1 の場合，BAQ \pm は 7190 秒であるのに対して，BAQ は 6810 秒である．BAQ \pm は複数の数値属性に対してバケットの平均値を計算する必要があるため，既存研究より時間を要した．ただし，シノプシスの生成はオフラインで実行するため，ユーザに対する影響は小さいと考えられる．

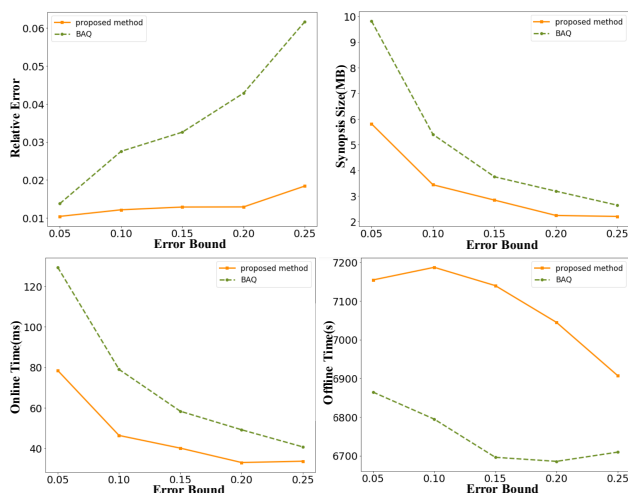


図 6 複雑な問合せの場合

以上より，BAQ \pm は同時に複数の集約関数に対する問合せする場合，0.05 から 0.25 までの誤差限界に推定誤差，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 まとめと今後の課題

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て近似処理システム BAQ \pm を提案した．また，上記のシノプシスを最適化できる近似的問合せ処理フレームワークを実装し，2 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 BAQ \pm 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．今後の課題としては，今回議論した BAQ \pm に基づく結合操作を含む問合せの近似処理について検討する予定である．

Acknowledgement

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

文 献

- [1] S. Chaudhuri, B. Ding, and S. Kandula, “Approximate query processing: No silver bullet,” in *Proc. SIGMOD*, pp. 511–519, 2017.
- [2] K. Li and G. Li, “Approximate query processing: What is new and where to go? A survey on approximate query processing,” *Data Science and Engineering*, vol. 3, pp. 379–397, 2018.
- [3] B. Mozafari and N. Niu, “A handbook for building an approximate query engine,” *IEEE Data Eng. Bull.*, vol. 38, no. 3, pp. 3–29, 2015.
- [4] A. Pol and C. Jermaine, “Relational confidence bounds are easy with the bootstrap,” in *Proc. SIGMOD*, pp. 587–598, 2005.
- [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] K. Li, Y. Zhang, G. Li, W. Tao, and Y. Yan, “Bounded approximate query processing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2262–2276, 2019.
- [7] S. Agarwal, A. Panda, B. Mozafari, S. Madden, and I. Stoica, “BlinkDB: Queries with bounded errors and bounded response times on very large data,” in *Proc. EuroSys*, pp. 29–42, 2013.
- [8] Y. Park, B. Mozafari, J. Sorenson, and J. Wang, “VerdictDB: Universalizing approximate query processing,” in *Proc. SIGMOD*, pp. 1461–1476, 2018.
- [9] 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.
- [10] Q. Ma and P. Triantafillou, “DBEst: Revisiting approximate query processing engines with machine learning models,” in *Proc. SIGMOD*, pp. 1553–1570, 2019.
- [11] 倪天嘉, 杉浦健人, 石川佳治, “誤差を保証する近似的問合せについて,” 第 13 回データ工学と情報マネジメントに関するフォーラム, 2021.
- [12] T. P. P. Council, “TPC Benchmark H (decision support) standard specification.” http://tpc.org/tpc_documents_current_versions/pdf/tpc-h_v3.0.0.pdf. Accessed January 4, 2022.