

誤差上限付き近似問合せ処理におけるシノプシス構築の高速化

堀崎 祥[†] 倪 天嘉^{††} 杉浦 健人^{††} 石川 佳治^{††} 陸 可鏡^{††}[†] 名古屋大学情報学部コンピュータ科学科 ^{††} 名古屋大学大学院情報学研究科

1 はじめに

近年、データ量の増大にともない、データベースにおける問合せ処理が重要視されている。大量データに対する問合せを効率的に実行するための手法として近似問合せ処理 (Approximate Query Processing, AQP) が注目されており、近年では誤差上限を保証可能な BAQ (Bounded AQP) [1] も提案された。BAQ には誤差保障に関し不足があり、それらを改善した BAQ± [2] も提案されている。

一方で、数値属性を含む場合に元テーブルに対する複数回のソート処理が必要となるなど、BAQ におけるシノプシス構築には未だ課題が残されている。そこで、本研究では効率的なシノプシス構築を前提として BAQ におけるシノプシスを再定義し、シノプシスの高速な構築アルゴリズムを提案する。

2 BAQ± の概要

本章では BAQ± の概要について述べる。BAQ± では、元データを事前に要約することで生成したシノプシスに対し近似的問合せ処理を実行する。シノプシスの生成にはユーザから目的のワークロードとして SQL の集合を受け取る必要がある。ここで、シノプシスの構築手順を図 1 に示す。以下、BAQ± における相対誤差の定義、およびシノプシス構築の手続きについて順に述べる。

2.1 相対誤差

BAQ± では、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。

2.2 問合せ単位のシノプシス構築

BAQ± では、ユーザが用意したワークロード中の問合せ単位でシノプシスを構築する。各問合せ q_i が問合せの列集合 (QCS) π_i に変換され、QCS を基にシノプシス S_i を構築する。各シノプシス S_i は π_i と要約レコード数 (SF) を保持する。表 2、表 3 にシノプシスの構築例を示す。

データの要約手法は QCS に含まれる列の属性により処理が

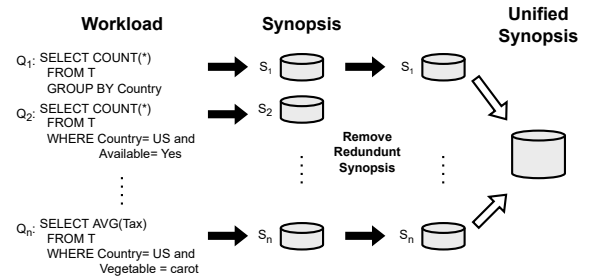


図 1 シノプシス構築手順

表 1 テーブル T

	Country	Vegetable	Available	Revenue	Tax
r_1	JP	tomato	Yes	1,200	120
r_2	JP	tomato	No	1,400	140
r_3	JP	carot	Yes	1,450	145
r_4	CN	onion	Yes	1,520	152
r_5	JP	carot	Yes	1,610	161
r_6	AU	onion	Yes	1,440	144
r_7	JP	carot	No	1,750	175
r_8	AU	onion	Yes	1,250	125

表 2 図 1 の Q_2 に対応するシノプシス例

Country	Available	SF
JP	Yes	3
JP	No	2
CN	Yes	1
AU	Yes	2

表 3 図 1 の Q_5 に対応するシノプシス例

Country	Vegetable	Tax	SF
JP	tomato	120	2
JP	carot	145	1
JP	carot	161	2
CN	onion	152	1
AU	onion	125	2

異なる。

QCS がカテゴリ属性のみの場合、元テーブルのレコードをカテゴリ属性でグループ化することによりシノプシスを構築する。カテゴリ属性とは、数が限られ全ての値が区別可能であるような属性を指す。

QCS に数値属性を含む場合、BAQ± ではバケット $b \subseteq R$ について、ユーザの指定した許容誤差 δ を用いて次式を満たすよ

Improving Synopsis Construction in Bounded Approximate Query Processing
 Sho Horisaki[†], Ni Tianjia^{††}, Sugiura Kento^{††}, Ishikawa Yoshiharu^{††}, and
 Lu Kejing^{††}

[†]Department of Computer Science, School of Informatics, Nagoya University

^{††}Graduate School of Informatics, Nagoya University

表4 表2, 表3の統合シノプシス

Country	Vegetable	Available	Revenue	Tax	SF ₂	SF ₅
JP	tomato	Yes	1,200	120	3	2
JP	tomato	No	1,400	140	2	0
JP	carot	Yes	1,450	145	0	1
CN	onion	Yes	1,520	152	1	1
JP	carot	Yes	1,610	161	0	2
AU	onion	Yes	1,440	144	2	2

うに数値属性をバケットに分割する.

$$err(\min b, \bar{b}) \leq \delta \wedge err(\max b, \bar{b}) \leq \delta \quad (2)$$

ここで, $\bar{b}$ をバケット内の平均値である. QCS にカテゴリ属性を含む場合は, まずカテゴリ属性についてグループ化する. 次に, 各グループの数値属性を 0, 正, 負の場合でグループを分け, 数値の絶対値が昇順になるようにレコードをソートする. ソートしたデータを先頭から順にバケットに挿入し式 (2) を満たすようバケットを分割する.

2.3 冗長なシノプシスの削除

問合せ単位でシノプシスを構築すると, ワークロード全体を考慮した際に冗長となるシノプシスが存在しうる. BAQ± では, あるシノプシスの QCS が他シノプシスの QCS の部分集合となる場合, 冗長と見なしシノプシスを削除する. 図 1 において Q_1 の QCS が {Country} であり, Q_2 の QCS が {Country, Vegetable} であることから, S_1 を冗長なシノプシスと見なし削除している.

2.4 統合シノプシスの構築

ここまでで構築したシノプシスで問合せ結果を返すことは可能だが, まだ冗長な情報が残されている. BAQ± では, シノプシス間で QCS が重なるものは冗長でありシノプシスを統合することで容量を小さくする. まず, 元テーブルのレコードをスキャンしそれらが問合せ単位のシノプシスの要素として使用できることを確かめる. 使用できる場合, レコードと対応する SF を統合シノプシスに挿入する. シノプシスの要素として使用できるレコードが全て見つかるまでスキャンを続ける.

表2, 表3を例に統合シノプシスの構築を説明する. まず, 元テーブル表1の先頭レコード (r_1) は表2において1つ目のタプル, 表3において1つ目のタプルを要素として使用できるため $SF_2 = 3, SF_5 = 2$ および r_1 を統合シノプシスに挿入する. 次に, r_2 は表2において2つ目のタプル, 表3において1つ目のタプルを要素として使用できる. ここで, 表3の1つ目のタプルは既に統合シノプシスに挿入済みのため, ここでは無視し $SF_2 = 3, SF_5 = 0$ および r_2 を統合シノプシスに挿入する. 以下, 同様の処理を繰り返すと最終的に表4に示す統合シノプシスを得る.

3 シノプシス構築の高速化

本章ではシノプシス構築の高速化のためにバケットを再定義し, それに合わせた構築アルゴリズムを提案する. 前述した BAQ± の既存のシノプシス構築アルゴリズムでは, 問合せ単位のシノプシス構築時にソート処理が必要となるため, ワーク

ロードの問合せが増えると処理時間が増大し, シノプシス構築処理のボトルネックとなる. このボトルネックを改善することで処理速度の大幅な改善が期待できる.

3.1 バケットの再定義

提案するバケット定義では, 各数値属性のバケットを全て事前に定めることによりソート処理を排除する. まず, 各数値属性 A_i に対し最小単位 u_i を指定し, 入力値 a_i が $|a_i| < u_i$ を満たすとき a_i を 0 と見なす. 0 と見なした値は ID=0 のバケットにまとめる. 最小単位を仮定することで, 全てのバケットが事前に定まる. ここで, 正の値を含む ID = j (0) のバケットを次式に示す.

$$[u_i(1+\delta)^{j-1}, u_i(1+\delta)^j] \quad (3)$$

逆に正の値が与えられたとき, その値を含むバケットの ID は次式で計算可能となる.

$$j = \left\lceil \log_{1+\delta} \frac{|a_i|}{u_i} \right\rceil \quad (4)$$

負の値を含むバケットも同様に計算可能である.

3.2 シノプシス構築アルゴリズム

提案する構築アルゴリズムでは, 元テーブルのレコード数に対し線形な時間計算量でシノプシスを構築できる. まず, 元テーブルの各レコードに対し, 数値属性を式 (4) で ID へ変換する. 変換した ID でレコードをシノプシスへ挿入ないし更新する. 挿入時は SF を 1 に設定し, 更新時は SF を 1 ずつ加算する.

これにより, BAQ± の既存アルゴリズムよりも構築時間を削減することが可能となる. 元テーブルのレコード数を n , シノプシスのタプル数を m とし, シノプシス側に索引を前提とすれば $O(n \log m)$ で構築可能である. 更に, BAQ± の既存アルゴリズムでは元テーブルに変更があるとシノプシスの再構築が必要なのに対し, 提案手法では差分レコードを挿入ないし更新するだけで再構築せず元テーブルの変更に追従可能である.

4 おわりに

本稿では, BAQ± におけるシノプシス構築の高速化手法を提案した. 今後は, 既存手法および提案手法の性能評価のために実験していく予定である. 具体的には, シノプシスの構築時間, 構築したシノプシスのサイズを主に比較する.

5 謝辞

本研究の一部は科研費 (JP20K19804, JP21H03555, JP22H03594) による.

参考文献

- [1] 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.
- [2] 倪 天嘉, 杉浦 健人, 石川 佳治, 陸 可鏡, "シノプシスの最適化に基づく近似問合せ処理の高速化," in 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, 2022.