Top-down精緻化を伴うPrivelet法における演算効率化手法の検討

本郷 節之[1]　手塚 理貴[†1]　寺田 雅之[2]　稲垣 潤[1]

概要： 近年，大規模な集計データの有効活用を目指す技術者の間で，差分プライバシ基準が有する高い安全性が注目を集めており，さまざまな実現手法が提案されている．なかでも，Privelet 法は，この差分プライバシ基準に準拠しつつ，部分和精度にも優れているという特徴をもつことから，プライバシが保護されたデータのスケーラブルな活用を可能にする．

Privelet 法を集計データに適用する場合，「非負制約の逸脱」や「スパースデータのデンス化」という問題が発生する．これらふたつの問題に対処するため，我々は Privelet 法に Top-down 精緻化手法の導入を提案した．この手法は，Privelet 法が本来もっているところの部分和精度が高いという特長を維持しつつも，上記ふたつの問題への対処を可能としている．この手法の場合，枝刈りと呼ばれる処理をその演算過程に導入することで，演算の効率化が可能である．しかし，その実装方法や効果の程度，性質などについては，未だ明らかにされていない．

そこで本報告では，Top-down 精緻化を伴う Privelet 法に対して，枝刈り処理を実装する方法の提案を行う．さらに，今回実現した枝刈り処理によってもたらされる演算効率化の程度と，その性質についても考察を行う．

Improving Computation Efficiency for Privelet with Top-down Refinement

SADAYUKI HONGO[1]　YOSHIKI TEZUKA[†1]　MASAYUKI TERADA[2]　JUN INAGAKI[1]

1. はじめに

デジタルデータの蓄積が加速される今日，蓄積されたいわゆるビッグデータから抽出・加工された大規模集計データの有効活用を図ることは，新たな産業分野の創生に対する大きな足掛かりとなる可能性がある．しかし，あらゆる日常シーンが情報通信ネットワークと融合している現代においては，多方面から収集，抽出，蓄積された集計データが，人々の消費活動や，日常的な生活行動などと結びついたものであることも少なくない．そこで，大規模集計データの有効活用を考えるとき，プライバシ保護の観点に立った高い安全性を確保することが，極めて重要となってくる．

プライバシ保護の重要性は国際的にも広く認識されており，個人データの国際的な流通を視野に，G7 参加国がすべて参加している OECD（経済協力開発機構）が中心となって，その取り組みを続けている．OECD は，1980 年に出した勧告に伴ってガイドラインとして示した八つの原則（OECD8 原則）[1] を，OECD 加盟国が国内法化することを勧告している．わが国では，これを受ける形で，1988 年に公的部門について，また，2003 年に民間部門について，個人情報の保護に関する法律（個人情報保護法）が制定された．1988 年の個人情報保護法は，その名の通り，行政機関の保有する個人情報の保護を目的としたものであったが，2003 年の保護法制定においては，個人情報の利活用に対する社会的要請を踏まえたものへと移行し，産業活動への有効活用の門戸が開かれた．さらにその後，OECD が 2013 年に発表したガイドラインの改定を受け，わが国

1 北海道科学大学
Hokkaido University of Science
2 株式会社 NTT ドコモ
NTT DOCOMO, Inc.
†1 現在，システムデザイン開発株式会社
Presently with System Design Development co., ltd

© 2018 Information Processing Society of Japan

でも，2015 年に個人情報保護法の大改正が行われた．この大改正は，個人情報関連技術への対応のみならず，マイナンバーの導入やビッグデータの利活用といった社会情勢の変化を踏まえた，経済活動において個人データを合法的に利用するための仕組みとしての位置づけも備えている．

このような社会的背景にも後押しされて，ビッグデータの有効活用とデータに対するプライバシ保護の両立を可能にする技術の研究も盛んに行われている．様々な産業活動で計量・蓄積されているビッグデータを広く見渡すと，商品の数や生物の個体数，事象の発生数や人口など，自然数，すなわち，非負値から構成される集計データであることが少なくない．また，データが，人口密集地や商業地などのような固有の特性を有するエリアのみに集中するような，全体的には疎（スパース）な分布をとる傾向もしばしば見られ，そしてこの傾向は，特に，集計データの規模が大きくなるほど生ずる可能性が高まる．そこで本研究では，元のデータベースに含まれる個々のデータの集合体（個票）から，何らかの条件を満たすデータの個数を数えた数値データの集合体であり，さらに，全体的にスパースな分布をとるような集計データを対象とする．

集計データに対するプライバシ保護に関しては，古くから検討が行われて来ている．そこではセル秘匿基準や $n-k\%$ 基準などに基づく基準が専門家によって注意深く適用されており，長年にわたって安全性が確保されて来た．しかし，ビッグデータに基づく大規模集計データでは，値の小さな大量のセル値に対しても，切り捨てるのではなく，活用することが望まれる．そこで近年，プライバシを保護しつつも有用なデータを有効に活用するための，新たな基準や手法に対する様々な研究が盛んに進められている．こうした技術はプライバシ保護データ公開（PPDP）と呼ばれ，$k-$ 匿名性基準 [2] や $l-$ 多様性 [3]，$m-$ 不変性 [4] など，さまざまな技術が提案されている．

これらの PPDP 技術では，前提とするところの，攻撃者の目的や能力，保有知識が異なっており，安全性を統一的に議論することは難しい．そうした中，近年，Dwork らが提案した差分プライバシ基準 [5,6] が，高い安全性を実現するための基準として注目を集めている．これは，データベースから集計データを作成した際に，「ある特定のデータがデータベースに含まれているか否かを集計データから判別することが困難である」ことを安全性の根拠とするプライバシ保護基準である．この基準を満たす処理手法は，従来手法と異なり，背景知識や攻撃手法に依存しない数学的な安全性を備えることが保証されている．

この差分プライバシ基準を満たす代表的な手法に Laplace メカニズムがある．この手法は，集計データの各セル値に対して，平均値が 0 の Laplace ノイズ（Laplace 分布に従う独立な乱数）を付加するものである．しかしこの Laplace メカニズムを，大規模集計データに適用すると，「非負制約の逸脱」「部分和精度の劣化」「計算量の増大」といった問題への対処が必要となる [8]．「非負制約の逸脱」とは，Laplace メカニズムが適用されたデータに，本来の集計データには存在し得ない負値が多く含まれることである．本来存在し得ない負値が大量に含まれることで，プライバシが保護された集計データを利用する際の利便性が損なわれる事態が懸念される．一方，「部分和精度の劣化」とは，複数セルの部分和をとった際に，元データの値に対する誤差値が大きくなる現象をさす．これは，Laplace ノイズが付加されたセル値の部分和をとる際に，付加されたノイズが多重に作用することにより誤差値が増大してしまい，集計データの利用価値が低下するような事態をさす．部分和処理は，プライバシが保護された集計データのスケーラブルな利活用を行う上で，重要な特性といえる．また，「計算量の増大」とは，Laplace ノイズの付加により，集計データにおける非 0 値の割合（密度）が増大してしまう現象である．特に大規模な集計データにおいてはその影響が顕著であり，計算量やデータ量が現実的では無くなってしまう可能性も懸念される．なお本稿では，処理効率の改善手法とその効果を検討対象としていることから，「演算処理の効率」と「非 0 データの増大の問題」という異なるふたつの問題に対する混同が発生することによる混乱を避けるため，この現象（計算量の増大）については「スパースデータのデンス化」という，現象の本質部分に着目した呼称を用いることとする．

これら 3 点の課題を同時に解決する手法として，我々は Top-down 精緻化を伴う Privelet 法を提案した [8]．これは，Xiao らによって提案された，Privelet 法 [7] が有する部分和精度が高いという性質を維持しつつも，非負制約の逸脱に対する回避と，スパースデータのデンス化の抑制を同時に実現する手法である．本報告では，この，Top-down 精緻化を伴う Privelet 法の演算を効率化する「枝刈り」処理の実装方法を提案する．さらに，今回実装した枝刈り処理によってもたらされる演算効率化の程度と，その性質についても考察を行う．

2. Topdown 精緻化を伴う Privelet 法

Privelet 法は，データ（長さ $n = 2^k$ のベクトル列 $V = \{v_1, v_2, \ldots, v_n\}$）に対して Haar 基底に基づく離散 Wavelet 変換 (HWT)$\mathcal{H}$ を導入し，その Wavelet 係数に対して Laplace メカニズムを適用した上で，逆 Wavelet 変換 $\mathcal{H}^{-1}$ を施すことで，差分プライバシ基準を満たすデータ V^+ を得る手法である．しかし，この方法で得られた集計データは，ノイズの影響により，非負制約を逸脱する．さらに，本来ゼロ値であった大量のデータが非ゼロ値となるため，スパースデータのデンス化も伴うことになる．

Top-down 精緻化を伴う Privelet 法では，HWT により

© 2018 Information Processing Society of Japan

得られた Wavelet 係数に対して，Laplace メカニズムの適用，および，逆 Harr Wavelet 変換，非負精緻化を適用して，差分プライバシを満たすデータ V^+ を得ている．オリジナルの Privelet 法に，非負精緻化処理を加えることで，非負制約の逸脱を回避するとともに，スパースデータのデンス化を抑制している．なお，文献 [8] では，Top-down 精緻化の構成法として，直列構成法と並列構成法という 2 つの構成法を提案しているが，ここでは，枝刈り処理による演算効率化が期待できる並列構成法を使用する．

2.1 Haar Wavelet 変換

はじめに，Haar Wavelet 変換部分の処理について概説する．基本となる処理は，n 個の入力データに Haar Wavelet 変換 $\mathcal{H}$ を適用して，$n/2$ 個の近似係数ベクトル cA，および，同じく $n/2$ 個の詳細係数ベクトル cD を得る，Haar 分解と呼ばれる処理である．まず，集計データベクトル $V = \{v_1, v_2, \ldots v_n\}$ を対象に，Haar 分解処理を適用する．

$$cA = \left(\frac{v_1+v_2}{2}, \frac{v_3+v_4}{2}, \ldots \frac{v_{n-1}+v_n}{2}\right),$$
$$cD = \left(\frac{v_1-v_2}{2}, \frac{v_3-v_4}{2}, \ldots \frac{v_{n-1}-v_n}{2}\right),$$

続いて，Haar 分解によって生成された $n/2$ 個の近似係数ベクトル cA に対して，再び Haar 分解を施すことで，$n/4$ 個の近似係数ベクトルと，同じく $n/4$ 個の詳細係数ベクトルがられ得る．同様に，Haar 分解処理を再帰的に繰り返すことで，最終的に，k 個の詳細係数ベクトルと，1 個の近似係数ベクトルが得られ，これらが HWT の出力 W となる．

2.2 Top-down 精緻化

次に，Top-down 精緻化を行って，差分プライバシ基準を満たす集計データを生成する．Top-down 精緻化では，HWT により得られた詳細係数ベクトル cD への Laplace メカニズムの適用，逆 HWT の適用，および，非負精緻化処理という 3 つの処理を行う．

2.2.1 Laplace メカニズムの適用

詳細係数ベクトル $cD = \{cD_1, cD_2, \ldots, cD_{n-1}\}$ への Laplace メカニズムの適用では，次式に従って cD を構成する全ての要素へ Laplace ノイズを加算することにより，差分プライバシを満たす詳細ベクトル $cD^* = \{cD_1^*, cD_2,^* \ldots, cD_{n-1}^*\}$ を得る [8]．

$$cD_i^* = cD_i + Lap\left(\frac{1}{\epsilon}\right)$$

ここで，$Lap(\cdot)$ は，Laplace 分布に従う乱数を，また，ϵ は，Laplace 分布のスケールを規定するパラメータを表す．

2.2.2 逆 HWT の適用

逆 HWT の適用については，図 1 を用いて説明する．HWT により得られた係数ベクトルの各要素は，図 1 のように 2 分木の各ノードに対応させて配置することができ

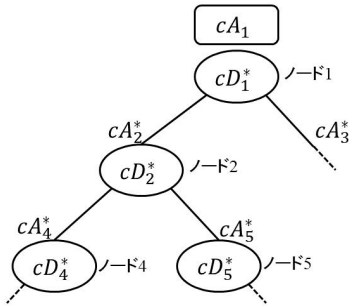

図 1 Top-down 精緻化

る．これらのうち，HWT の出力 W として保存されたのは，cA_1 および $n-1$ 個の cD_i である．そしてこの段階では，Laplace メカニズムの適用により cD_i には Laplace ノイズが付加されて，cD_i の値は cD_i^* へと変化している．

逆 HWT 処理は，HWT 処理と逆のプロセスをたどる．すなわち，2 分木の上側からの処理となる．ノード 1 には，詳細係数 cD_1^* と近似係数 cA_1 とが割り付けられているが，次式で表す演算により一階層下に位置する 2 つの近似係数を求める $(i = 1, j = 2)$．

$$cA_j^* = cA_i + cD_i^* \tag{1}$$
$$cA_{j+1}^* = cA_i - cD_i^* \tag{2}$$

次に，ノード 2 に着目する．上記の演算により得られた近似係数 cA_2^* の値と，ノード 2 に格納されている詳細係数 cD_2^* とを用いて，同様の演算を行い，一階層下に位置する 2 つの近似係数を求める．同様に，ノード i に対応する近似係数 cA_i^* と詳細係数 cD_i^* とから，一階層下に位置する 2 つの近似係数 cA_j^* および cA_{j+1}^* を求める演算を再帰的に実行して行く．こうすることで，最終的に差分プライバシを満たす集計データベクトル V^* が得られる．

Privelet 法では cD に対して Laplace メカニズムが適用されているために，各ノードに対応する詳細係数 cD_i が異なる値に変化してしまっていることから，逆 HWT の結果得られるデータは，入力と異なる値となり，このことに起因して，非負制約の逸脱やスパースデータのデンス化といった不都合が発生することになる．そこで，逆 HWT の過程に，非負精緻化処理を導入する．

2.2.3 非負精緻化の導入

近似係数 cA_j の値は，対応するノード j に集約される入力データの平均値であるから，HWT への入力データが非負値であった場合，必ず非負値となる．しかし，Laplace メカニズムの適用により $cA_j (1 < j \leq n-1)$ を算出する際に使用される cD_i の値がもとの値と異なっていることから，逆 HWT 処理の過程で，負の値をもった cA_j が生ずる場合が発生する．上述した通り，cA_i の値は対応するノード i に集約される入力データの平均値であるから，cA_j の値が負値であることは，出力データに多数の負値が発生する事態を招く可能性がある．そこで，cA_j の値に負値が生

© 2018 Information Processing Society of Japan

じないように一階層上の cD_i の値を精緻化する．

ノード 2 に着目する．式 (1, 2) と同様にして，一階層下の近似係数を求める $(i = 2, j = 4)$.

$$cA_j^* = cA_i^* + cD_i^*$$
$$cA_{j+1}^* = cA_i^* - cD_i^*$$

ここで，cA_j^* または cA_{j+1}^* のいずれかが負値となった場合，cD_i^* の値を，符号は変えずに，その絶対値を $|cA_i^*|$ へと置き換えることにより，非負精緻化が施された詳細係数 cD_i^+ を得る．

$$cD_i^+ = sign(cD_i^*) \cdot |cA_i^*| \tag{3}$$

ここで，$sign(\cdot)$ は，以下の値をとる符号関数である．

$$sign(x) = \begin{cases} -1 & \text{if } x < 0 \\ +1 & \text{otherwise} \end{cases}$$

そして，この非負精緻化を施した cD_i^+ の値を用いて改めて次式の処理を改めて行うことにより，一階層下に位置する非負精緻化が施された 2 つの近似係数を求める $(i = 2, j = 4)$.

$$cA_j^+ = cA_i^* + cD_i^* \tag{4}$$
$$cA_{j+1}^+ = cA_i^* - cD_i^* \tag{5}$$

上記に示す非負精緻化が施された cD_i^+ の効果により，$cD_j, cDj + 1$ の値は，ともに非負値となる．そしてその結果，最終的に出力される差分プライバシ基準を満たす集計データ V^+ の値も全て非負値となる．

3. Top-down 精緻化を伴う Privelet 法の演算効率化

上述した Top-down 精緻化を伴う Privelet 法では，一旦非負精緻化処理が発生すると，その片側子ノードから下は，全ての詳細係数の値が 0 となる．これは，非負精緻化処理を施したノードでは詳細係数と近似係数との絶対値が等しくなることから，式 (4) または式 (5) いずれかの近似係数の値 (cA_j^+ または cA_{j+1}^+) が必ず 0 になることに起因する．もしも近似係数の値が 0 のノードに対応する詳細係数が 0 以外の値をとると，またそこで非負精緻化処理が発生することとなり，そのノードの詳細係数の値は 0 へと精緻化される．こうした非負精緻化処理が繰り返されることにより，詳細係数の値が 0 となったノードに連結される下層のノードでは，近似係数・詳細係数ともに全て 0 となり，その結果，当該ノードに連結されている出力値も全て 0 となる．

この性質に着目すると，近似係数 (および詳細係数) の値が 0 となるノードの演算を省略し，Top-down 精緻化を伴う Privelet 法の処理を効率化することができる．ここではこの効率化処理を“枝刈り”と呼ぶことにする．第 2.2.2 項で述べた通り，逆 HWT 処理は最上位層（層内のノード数が 1 の層）から，一層ずつ下りながら，順次演算を行って行く．説明のため，図 2 に示すように，最下層（入力層）の層番号を $i = 0$，最上位層の層番号を $i = k$ とする層番号 i を付与する．一方，各層では層内に含まれるノードを順次訪問しながら，非負精緻化が組み込まれた逆 HWT 処理を繰り返す．同一層内のノードの訪問順序に特に制約はないので，ここでは，図 2 に示すように，左側から連続したノード番号 x を付与し $(1 \leq x \leq 2^{k-i})$，このノード番号の順番に則って層内各ノードの訪問を行うこととする．

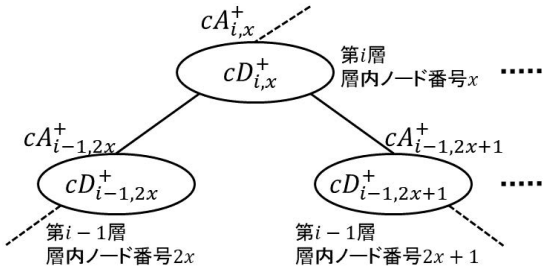

図 2 層番号と層内ノード番号

各ノードにおいて演算を省略するか否かは，近似係数 $cA_{i,x}^+$ の値が 0 であるか否かで判定する．上述した通り，近似係数 $cA_{i,x}^+$ の値が 0 であれば，非負精緻化処理の結果，詳細係数 $cD_{i,x}^+$ の値は必ず 0 となる．すなわち，式 (4) および式 (5) の値は，ともに 0 となる．したがって，近似係数 $cA_{i,x}^+$ の値が 0 の場合には，計算せずとも結果が 0 となることが自明であり，この演算を省略することができる．さらに，Haar Wavelet の場合，2 分木構造をもっていることから，階層間での対応が容易である．処理が一層ずつ下るにつれて，同一層内に含まれるノード数は 2 倍となり，また，第 i 層で x 番目に位置したノードに対応するノードは，第 i-1 層では $2x$ および $2x + 1$ 番目に位置することとなる．

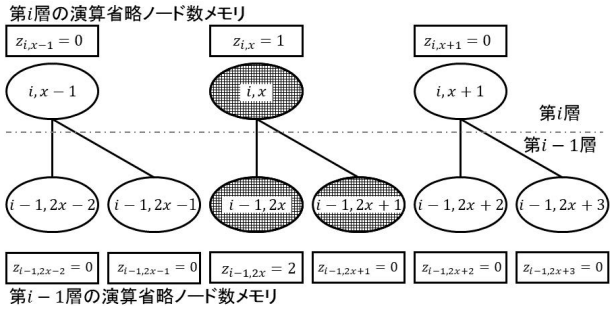

図 3 層間におけるノードの対応と演算省略ノード数メモリ

図 3 に，階層間におけるノードの対応と演算省略ノード数メモリの様子を示す．一点鎖線より上側が第 i 層を，下側が第 $i - 1$ 層をそれぞれ表している．楕円がノードを，長方形が演算省略ノード数メモリをそれぞれ表している．楕円内には層番号と層内ノード番号の組が，長方

© 2018 Information Processing Society of Japan

形内には演算省略ノード数メモリを表す変数ベクトル $Z = (z_{k,1}, z_{k-1,1}, z_{k-1,2}, \ldots, z_{0,2^k,})$ の要素とその値が記されている．

上述したような層間におけるノード関係から，いま，ノード (i,x) で $cA^{+}_{i,x} = 0$ が検出され，演算が省略されるとき，第 $i-1$ 層ではそこに連結される 2 つの下位ノード $(i-1,2x)$ および $(i-1,2x+1)$ においても，$cA^{+}_{i-1,2x} = 0$ および $cA^{+}_{i-1,2x+1} = 0$ が検出されることとなり，一層下るごとに，省略できるノード数が 2 倍になることがわかる．そこで，Top-down 精緻化処理において $cA^{+}_{i,x} = 0$ を検出した層内ノード番号 x と演算省略ノード数 $z_{i,x}$（初期値は 1）を記録しておき，一層下るごとに，層内ノード番号 x ならびに演算省略ノード数 $z_{i,x}$ を 2 倍して行けば，非負精緻化を組み込んだ逆 HWT 演算の過程で記録された層内ノード番号 x の位置から $z_{i,x}$ ノード分の演算を一気に省略可能であることが期待できる．ただし，最下層（出力層）においてのみ，演算を省略する部分に相当するノードにゼロ値を格納する必要がある点には注意が必要である．

4. 演算効率化作用の評価

4.1 評価方法

本評価では，異なる複数のエリアにおける人口分布データに対して Privelet 法による秘匿処理を適用し，非負精緻化を組み込んだ逆 HWT 処理部分の処理時間を計測する．“(α) 枝刈りあり”と“(β) 枝刈りなし”との間の処理時間の差を“(β) 枝刈りなし”の処理時間で正規化した値 $\{(\beta)-(\alpha)\}/\beta$ をここでは“時間短縮率”と呼ぶこととし，演算量抑制効果の指標とする．

本評価では，平成 22 年度国勢調査に基づく地域メッシュ人口（1km メッシュ）のデータに対して，一次元 Haar Wavelwt 型の，Top-down 精緻化を伴う Privelet 法を適用した．日本全国（2^{11}メッシュ $\times$ 2^{11}メッシュ）のデータから,(1) 北海道（$2^9 \times 2^9$）,(2) 四国（$2^8 \times 2^8$）, (3) 関東（$2^8 \times 2^8$）の各エリアを切り出して評価用のデータとした．また，他エリアとの比較用に，隣接する縦横 2×2 メッシュの人口をひとまとめにした (4) 北海道 1/4（$2^8 \times 2^8$）も評価用データに加えた．

なお，本評価を行うにあたって，二次元上に配置された地域メッシュ人口データを，一次元化する方式による差異についても確認するべく，(a) ラスター方式，(b) ソート方式（降順），(c)Morton 方式 [8]，(d) ランダム方式という 4 方式によって一次元化を行い，各々に対して評価を行った．(a) ラスター方式では，2 次元に配置されているメッシュ人口データを，X 軸方向に取り出す操作を，Y 軸方向に繰り返すことでデータの 1 次元化を行っている．(b) ソート方式のデータは，(a) ラスター方式で得られた 1 次元データを，その値の大きい準（降順）に並べ替えたものである．(c)Morton 方式のデータは，2 次元に配置されているメッシュ人口データに Morton 写像を施したものである．Morton 写像は，多次元空間から一次元空間への全単射を行う写像であり，元の空間上における距離の遠近が写像先の空間における距離の遠近に反映される性質を持つ，局所性保存写像の一種である．(d) ランダム方式のデータは，(a) ラスター方式で得られた 1 次元データに対して一様乱数を用いた並べ替え処理を施したものである．なお，(b) ソート方式は差分プライバシを満たさないが，他 3 方式と比較する目的で加えている．

評価には Intel Core i7 CPU (2.93GHz), 実装メモリ 4GB のデスクトップ PC を使用した．また，同一処理を 100 回繰り返した時間を計測して 1/100 し，計測時間の精度向上を図った．

4.2 評価結果

表 1 に処理時間の計測結果を示す．4 方式間での時間短縮率の傾向を見ると，ソート方式において最も時間短縮率が高く，ランダム方式において最も低い．また，同一メッシュ数をもつ 3 エリア間での時間短縮率の傾向については，北海道 1/4 の値が最も高く，関東の値が最も低い．さらに，メッシュ数の多い北海道エリアは，著しく高い時間短縮率を示している．

表 1 処理時間の計測結果

一次元化方式	エリア	処理時間 (100 ループの平均) (α) 枝刈あり [ms]	処理時間 (100 ループの平均) (β) 枝刈なし [ms]	処理時間比率 [%] α/β	時間短縮率 [%] $(\beta-\alpha)/\beta$
ラスター方式	北海道	22.8	48.1	47.4	52.6
	四国	10.2	12.3	82.6	17.4
	関東	11.2	12.0	93.7	6.3
	北海道 1/4	8.4	12.0	69.7	30.3
ソート方式	北海道	4.3	47.9	9.0	91.0
	四国	3.4	11.9	28.3	71.7
	関東	5.4	11.7	46.0	54.0
	北海道 1/4	1.8	11.9	15.3	84.7
Morton 方式	北海道	10.5	47.2	22.2	77.8
	四国	7.9	12.0	66.0	34.0
	関東	9.3	11.9	77.9	22.1
	北海道 1/4	5.7	11.7	48.3	51.7
ランダム方式	北海道	32.6	48.5	67.2	32.8
	四国	15.1	12.1	125.2	-25.2
	関東	16.0	11.9	134.3	-34.3
	北海道 1/4	12.2	11.9	102.5	-2.5

表 2 には非負精緻化発生回数とゼロ値含有比率の値を示す．4 方式間での非負精緻化発生回数の傾向を見ると，ランダム方式で最も多く，ソート方式が最も少ない．また，同一メッシュ数をもつ 3 エリア間での非負精緻化発生回数を比較すると，ソート方式を除くと四国が最も多く，ラス

© 2018 Information Processing Society of Japan

ター方式以外で北海道 1/4 が最も少ない．一方，処理結果のゼロ値含有比率は，ソート方式の値が最も高く，ランダム方式で最も低い．そして，同一メッシュ数をもつ 3 エリア間での処理結果のゼロ値含有比率を比較すると，北海道 1/4 の値が最も高くて関東の値が最も低く，元データのゼロ値含有比率の傾向を反映している．さらに，メッシュ数の多い北海道エリアは，著しく高い非負精緻化発生比率，ならびに，処理結果のゼロ値含有比率を示している．

表 2　非負精緻化発生回数とゼロ値含有比率

一次元化方式	エリア	元データのゼロ値含有比率 [%]	時間短縮率 [%] $(\beta-\alpha)/\beta$	非負精緻化発生回数	処理結果のゼロ値含有比率
ラスター方式	北海道	95.0	52.6	57,370	90.8
	四国	78.7	17.4	23,867	68.2
	関東	61.2	6.3	20,649	50.9
	北海道 1/4	90.3	30.3	20,805	81.8
ソート方式	北海道	95.0	91.0	6,261	96.9
	四国	78.7	71.7	4,835	85.3
	関東	61.2	54.0	5,640	64.9
	北海道 1/4	90.3	84.7	3,033	92.8
Morton方式	北海道	95.0	77.8	23,841	95.3
	四国	78.7	34.0	16,622	73.9
	関東	61.2	22.1	15,322	55.6
	北海道 1/4	90.3	51.7	13,273	85.9
ランダム方式	北海道	95.0	32.8	82,485	84.9
	四国	78.7	-25.2	35,793	45.9
	関東	61.2	-34.3	31,154	18.3
	北海道 1/4	90.3	-2.5	30,154	70.6

5. 考察

図 4 に非負精緻化の発生回数と時間短縮率の関係を示す．グラフから，非負精緻化の回数と処理時間との間に単純な相関は見られないことがわかる．これは，非負精緻化が発生する階層によって，1 回の非負精緻化が演算時間短縮に及ぼす効果が異なるためであると考えられる．すなわち，Wavelet 木の上位の層で非負精緻化が起こると省略が起こる演算数が多くなり効率化の効果が大きく現れるが，その一方で，下位の層で非負精緻化が起こっても省略が起こる演算数はそれほど大きくもなく，大きな時間短縮率は現れないことに起因している現象と考えられる．

次に，処理結果のゼロ値含有比率と時間短縮率の関係について評価を行った．前述の通り，非負精緻化が起こると，その片側子ノードから下の値は全て 0 となる．この特性に着目すると，枝刈りによる時間短縮の効果（時間短縮率）は，処理結果のゼロ値含有比率との間に正の相関を持つことが期待できる．図 5 に，今回行った評価における，処理結果のゼロ値含有比率と時間短縮率の関係を示す．グラフから，両者の間に高い正の相関関係が見て取れる．

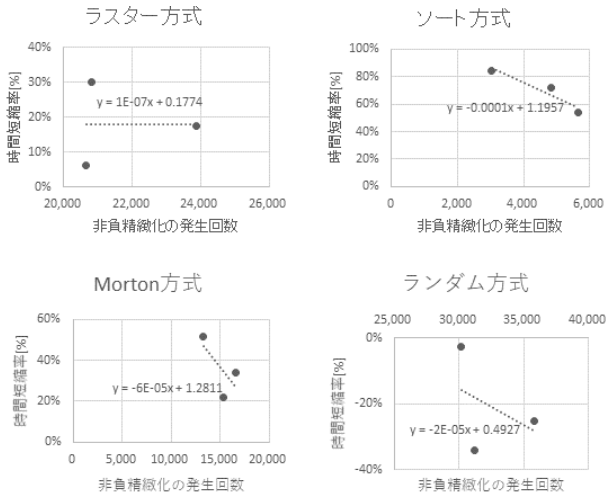

図 4　非負精緻化発生回数と時間短縮率の相関

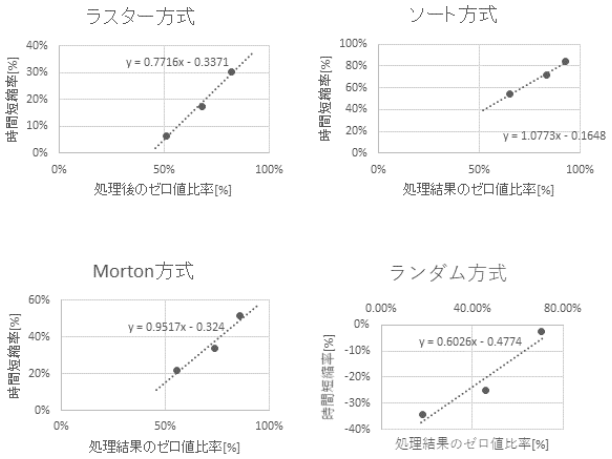

図 5　ゼロ値発生比率と時間短縮率の相関

6. おわりに

Top-down 精緻化を伴う Privelet 法の演算量を抑制する手法を提案し，効果の評価を行った．評価結果から，処理結果のゼロ値含有比率が高いほど時間短縮率が高く，さらに，メッシュ数の多い北海道エリアについては，著しく高い時間短縮率を示すことがわかった．処理結果のゼロ値含有比率は元データのゼロ値含有比率とも相関が高い．すなわち，提案手法がもたらす演算時間の短縮効果は，スパースなデータほど有効性が期待できることを明らかにすることができた．今後は，非負精緻化の発生位置と演算効率化の関係についての，より詳細な評価を進めて行くことが課題である．

謝辞

本研究の一部は日本学術振興会科学研究費補助金基盤研究 (C)（課題番号：15K00190）による補助を受けて行なわれた．

© 2018 Information Processing Society of Japan

参考文献

[1] OECD 理事会勧告 8 原則, 入手先 ⟨http://www.soumu.go.jp/main_sosiki/gyoukan/kanri/oecd8198009.html⟩. （参照 2018-05-01）.

[2] Sweeney, L.: *k-anonimity: A model for protecting privacy*, Intl. J. Uncertainty, Fuzziness and KnowledgeBased Systems, Vol.10, No.5, pp.557-570 (2002).

[3] Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M.: *l-diversity: Privacy Beyond k-anonymity*, ACM Trans. Knowledge Discovery from Data (TKDD), Vol.1, No.1 (2007).

[4] Xiao, X. and Tao, Y.: *m-lnvariance: Towards Privacy Preserving Re-publication of Dynamic Datasets*, Proc. 2007 ACM SIGMOD Intl. Conf. Management of Data, pp.689-700, ACM (2007).

[5] Dwork, C.: *Differential Privacy*, Proc. 33rd Intl. Conf. Automata, Languages and Programming - Volume Part II, Bugliesi, M., Preneel, B., Sassone, V. and Wegener, I.(Eds.), Lecture Notes in Computer Science, Vol.4052, pp.1-12, Springer (2006).

[6] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K. and Berkeley, U. C.: *Privacy, accuracy, and consistency too: a holistic solution to contingency table release*, Proc. 26th ACM SIGMOD-SIGACT-SIGART symp. Principles of database systems - POD'S07, ACM Press, 273282 (2007).

[7] Xiao, X., Wang, G., Gehrke, J. and Jefferson, T.: *Differential Privacy via Wavelet Transforms*, IEEE Trans. Knowledge and Data Engineering, 23, 8, 1200-1214 (2011).

[8] 寺田，鈴木，山口，本郷: 大規模集計データへの差分プライバシの適用, 情報処理学会論文誌, 56, 9,1801-1816 (2015).

[9] 本郷，手塚，寺田，稲垣: 集計データへの差分プライバシ適用における特性の一考察 II, 第 80 回情報処理学会全国大会，情報処理学会第 80 回全国大会講演論文集, 3, 457-458 (2018).

© 2018 Information Processing Society of Japan