情報処理学会論文誌 Vol.61 No.2 474–485 (Feb. 2020)

推薦論文

非負精緻化をともなうPrivelet法の演算効率化手法

本郷 節之[1,a)] 寺田 雅之[2] 鈴木 昭弘[1] 稲垣 潤[1]

受付日 2019年4月1日, 採録日 2019年11月7日

概要：Privelet 法は，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おり，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．この Privelet 法に非負精緻化処理を組み込むと，高い部分和精度を維持しつつ，さらに，「非負制約の逸脱」や「疎データの密度急増」という 2 つの問題への対処も可能となる．この手法の場合，非負精緻化をともなう逆 Wavelet 変換（Top-down 精緻化）部分に枝刈り処理を導入することで，演算を効率化することができる．しかし，その効果の程度，性質などについては，いまだ明らかにされていない．そこで本稿では，演算効率化が期待できる実装法を用いて，枝刈り処理の性能面に対する評価を行い，さらにその特性についての考察を行う．

キーワード：プライバシ保護，差分プライバシ，ウェーブレット変換，非負制約

A Method to Improve the Computational Efficiency of Privelet with Nonnegative Refinement

Sadayuki Hongo[1,a)] Masayuki Terada[2] Akihiro Suzuki[1] Jun Inagaki[1]

Received: April 1, 2019, Accepted: November 7, 2019

Abstract: Privelet is a data publishing technique that ensure ϵ-differential privacy while providing accurate answers for range-count queries. This technique is suitable for scalable utilization of privacy-preserved data. However, it has two problems which are “deviation from the non-negative constraint” and “abruptly increase of data-density”. Privelet with non-negative refinement solves these two problems without losing the accuracy of the partial summation. Introducing the pruning process into the top-down refinement - the inverse wavelet transform with nonnegative refinement - improves the computational efficiency. However, its effects and characteristics have not been clarified yet. In this paper, after showing the evaluation results for performance using an implementation method which can be expected to improve the efficiency of computation, its characteristics are discussed.

Keywords: privacy-preserving data utilization, differential privacy, wavelet transform, non-negative constraint

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出に対する大きな足が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付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，きわ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

[1] 北海道科学大学
Hokkaido University of Science, Sapporo, Hokkaido 006–8585, Japan
[2] 株式会社 NTT ドコモ
NTT DOCOMO, Inc., Yokosuka, Kanagawa 239–8536, Japan
[a)] hongo@hus.ac.jp

本稿の内容は 2018 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2018）シンポジウムにて報告され，同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

り，個人データの国際的な流通を視野に，すべての G7 参加国が加盟する 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 年に出した勧告にともなってガイドラインとして示した 8 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，その名のとおり，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ふ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だけでなく，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ふまえた，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，ビッグデータに基づく大規模かつ高次元な集計データの作成が現実的なものとなりつつある．そしてその普及と活用に，客観的な事実に基づく社会活動の最適化を実現する鍵としての期待が高まっている [2]．またこうした期待のもと，集計データの有効活用とデータに対するプライバシ保護との両立を可能にする技術の研究もさかんに行われている．

さまざまな産業活動で計量・蓄積されている集計データを広く見渡すと，商品の数や生物の個体数，事象の発生件数や人口など，自然数を含む非負値から構成される集計データであることも少なくない．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．そこで本稿では，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，全体的に疎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．これらは統計的開示制御（statistical disclosure control）[3], [4] と呼ばれ，そこではセル秘匿基準や $n-k$%基準などに基づく各種の秘匿方式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きた [5], [6]．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さまざまな研究がさか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 [7]，k-匿名性 [8] や l-多様性 [9]，m-不変性 [10] など，さまざまな基準や手法が提案されている．

しかし，これらの PPDP 技術で前提とするところの，攻撃者の目的や能力，保有知識はそれぞれ異なっており，安全性を統一的に議論することは難しい．そうしたなか，近年，Dwork らが提案した差分プライバシ基準 [11], [12] 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．差分プライバシ基準は，ϵ-差分プライバシ基準とも呼ばれ，データベースへの問合せ（クエリ）を行った際に，「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問合せ結果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．ϵ をある小さな正の定数とし，データベース D に対して確率的問合せ M を適用したときに，問合せ結果として t が得られる確率を $Pr[M(D)=t]$ とする．ここで，任意の隣接する（たかだか 1 レコードしか異ならない）データベース D_1，D_2 に対して，$Pr[M(D_1)=t]/Pr[M(D_2)=t] \leq e^{\epsilon}$ が成立するとき，この問合せ M は ϵ-差分プライバシ基準を満たす．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，データベースへの問合せ結果に対して，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．適用対象が集計データの場合には，単に集計データに含まれる各セル値に対してそれぞれ Laplace ノイズを付加すれば良い．Laplace 分布の確率密度 $\ell(x)$ は，平均 μ とスケール λ を用いて次式で与えられる．

$$\ell(x:\mu,\lambda) = \frac{1}{2\lambda} e^{(-|x-\mu|/\lambda)}$$

ここで，平均 0，スケール λ の Laplace 分布に従って発生させた Laplace ノイズを $Lap(\lambda)$ とし，k 個の互いに独立な $Lap(\lambda)$ からなるスカラベクトルを $Lap(\lambda)^k$ と記すこととする．Laplace メカニズムにおける Laplace ノイズのスケール λ は，パラメータ ϵ と，問合せの種類ごとに決まる大域的感度（global sensitivity，GS）によって与えられる．具体的には，GS_f を問合せ f の感度としたとき，f に対応するランダム化関数は次式で表される．

$$f(X) + Lap(GS_f/\epsilon)^d$$

$$GS_f = \max_{D_1, D_2} \parallel f(D_1) - f(D_2) \parallel_1$$

ここで D_1 および D_2 は任意の隣接したデータベースのペアであり，d はスカラベクトルデータ X の要素数を表すものとする．V が分割表，すなわち，構成する部分集合が互

いに素であるとき，計数間問合せの大域的感度は 1 であることが知られている．したがって，集計データの各セルにスケール $\lambda = 1/\epsilon$ の Laplace ノイズを加えることで，ϵ-差分プライバシを満たすことができる．

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 [13]．「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．データのなかに本来存在し得ない負値が大量に含まれる状況下では，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うえで，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なくなってしまう可能性も懸念される．なお本稿では，演算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」という異なる 2 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「疎データの密度急増」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．

これらの課題に対して，部分的な改善方式がいくつか提案されている [12], [14], [15], [16]．しかし，いずれの方式においても，3 点の課題に対して同時に対処することはできていなかった．そうしたなか，これら 3 点の課題を同時に解消・改善する手法[*1]として，我々は非負精緻化をともなう Privelet 法を提案した [13]．この手法によれば，本来の Laplace メカニズムにおける「負値の発生」や「部分和精度の劣化」，「非 0 データの急増」といった，実用上の困難や問題を解消または改善することができる．これは，Xiao らによって提案された Privelet 法 [15], [16] 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，「疎データの密度急増」の抑制を同時に実現する手法である．一方，近年の研究を見ても，これら 3 つの困難・問題への対処を同時に行う手法は見あたらない．そこで本稿では，この「非負精緻化をともなう Privelet 法」を対象としている．

非負精緻化をともなう Privelet 法は，枝刈り処理と呼ばれる「演算の省略」を行うことで，演算を効率化することができるという性質を備えている．しかしその実装法については明らかにされておらず，またその性能面に対する特性評価も行われていなかった．本稿では，演算効率化が期待できる実装法 [17], [18] を用いて，その性能面に対する特性評価を行う．

まず 2 章では，既存研究ならびに非負精緻化をともなう Privelet 法の概要を解説する．次に 3 章では，本評価において採用したところの，非負精緻化をともなう Privelet 法の演算を効率化する実装法について述べる．続いて 4 章では，枝刈り処理による演算効率化作用の評価方法とその結果を示す．そして 5 章では，1 次元への写像方式と時間短縮率との関係，ならびに，省略された演算数を反映する「重み付き枝刈り発生回数」と時間短縮率との関係についての考察を行う．

2. 非負精緻化をともなう Privelet 法

Privelet 法は，長さ $n = 2^H$ のスカラベクトルデータ $V = (v_1, ..., v_n)$ 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うえで，逆 HWT$\mathcal{H}^{-1}$ 処理を施すことで，差分プライバシ基準を満たすスカラベクトルデータ 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，疎データの密度急増もともなうことになる．

非負精緻化をともなう Privelet 法 [13] では，HWT により得られた Wavelet 係数に対して，Laplace メカニズムの適用，および，非負精緻化をともなう逆 Wavelet 変換を適用して，差分プライバシを満たすスカラベクトルデータ 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，疎データの密度急増を抑制している．なお，文献 [13] 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，集計データであるところのスカラベクトル $V = (v_1, v_2, \ldots v_n)$ を対象に，Haar 分解

[*1] 文献 [13] で提案しているのは「非 0 データの増大」の抑制に基づく「計算量の増大」の抑制である．一方，本稿で対象としているのは，枝刈りによる「演算処理の効率」の向上である．両者は，「計算量の増大」を抑制するためのアプローチが異なっている．

処理を適用する．

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

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得られる．同様に，Haar 分解処理を再帰的に繰り返すことで，最終的に，1 つの近似係数と，$n-1$ 個の要素からなる詳細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 1 つの近似係数および詳細係数ベクトルへの Laplace メカニズム適用，逆 HWT，および，非負精緻化という 3 つの処理を行う．

2.2.1 Laplace メカニズムの適用

Laplace メカニズムの適用では，次式に従って，近似係数 $cA_{H,0}$ と，詳細係数ベクトル cD を構成するすべての要素へ 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近似係数 $cA^*_{H,0}$ および詳細係数ベクトル cD^* を得る [13].

$$cA^*_{H,0} = cA_{H,0} + Lap\left(\frac{1}{2^H\epsilon}\right)$$
$$cD^*_{h,x} = cD_{h,x} + Lap\left(\frac{1}{2^h\epsilon}\right)$$

ここで，$h = 0, 1, \cdots, H$ は階層番号を，$x = 0, 1, \cdots, 2^{H-h}-1$ は階層内ノード番号を，$Lap(\cdot)$ 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．なお，最下層（$h = 0$）にあたる $n = 2^H$ 個の変数群はリーフと呼ばれ，秘匿対象および結果のデータがここに格納される．

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る．これらのうち，HWT の出力 W として保存されたのは，$cA_{H,0}$ および $n-1$ 個の $cD_{h,x}$ である（$1 \le h \le H$）．そしてこの段階では，Laplace メカニズムの適用により $cA_{H,0}$ および $cD_{h,x}$ には Laplace ノイズが付加されており，それぞれ $cA^*_{H,0}$ および $cD^*_{h,x}$ へと値が変化している．

逆 HWT 処理は，HWT 処理と逆のプロセスをたどる．すなわち，逆 HWT 処理は 2 分木の上側からの処理となる．ノード $(H,0)$ には，詳細係数 $cD^*_{H,0}$ と近似係数 $cA^*_{H,0}$ とが割り付けられているが，ここでまず，次式で表す演算により 1 階層下に位置する 2 つの近似係数を求める（$h = H$，$x = 0$）．

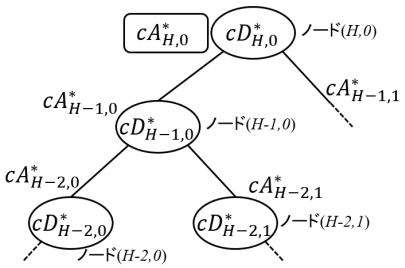

図 1 逆 Wavelet 変換
Fig. 1 The inverse Wavelet transform.

$$cA^*_{h-1,2x} = cA^*_{h,x} + cD^*_{h,x} \quad (1)$$
$$cA^*_{h-1,2x+1} = cA^*_{h,x} - cD^*_{h,x} \quad (2)$$

次に，ノード $(H-1,0)$ に着目する．上記の演算により得られた近似係数 $cA^*_{H-1,0}$ の値と，ノード $(H-1,0)$ に格納されている詳細係数 $cD^*_{H-1,0}$ とを用いて，同様の演算を行い，1 階層下に位置する 2 つの近似係数を求める．以下，同様に，ノード h,x に対応する近似係数 $cA^*_{h,x}$ と詳細係数 $cD^*_{h,x}$ とから，1 階層下に位置する 2 つの近似係数 $cA^*_{h-1,2x}$ および $cA^*_{h-1,2x+1}$ 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（スカラベクトル V^*）が得られる．

Privelet 法では，近似係数および詳細係数ベクトルに対して Laplace メカニズムが適用されているために，それらが HWT によって得られた数値とは異なる数値に変化する．その結果，逆 HWT により得られるスカラベクトル V^* は，入力と異なる値となり，データの秘匿が実現される．しかしその一方で，一連の処理に起因して，非負制約の逸脱や疎データの密度急増といった副作用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 非負精緻化処理の導入

近似係数 $cA_{h-1,2x}$ や $cA_{h-1,2x+1}$ の値は，それぞれ対応するノード $(h-1,2x)$ および $(h-1,2x+1)$ 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には必ず非負値となる．しかし，Laplace メカニズムの適用により，$cA^*_{h-1,2x}$ や $cA^*_{h-1,2x+1}$ へと値が変化した結果，負値となってしまう場合も生じ得る．上述したとおり，これらの値はともに，対応するノードに集約される入力データの平均値であるから，これらの値が負値となることで，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，$cA^*_{h-1,2x}$ や $cA^*_{h-1,2x+1}$ の値に負値が生じないように 1 階層上の $cD^*_{h,x}$ の値を精緻化する．

ノード $(H-1,0)$ に着目する．式 (1)，(2) と同様にして，1 階層下の近似係数を求める（$h = H-1$，$x = 0$）．

$$cA^{*}_{h-1,2x} = cA^{*}_{h,x} + cD^{*}_{h,x}$$
$$cA^{*}_{h-1,2x+1} = cA^{*}_{h,x} - cD^{*}_{h,x}$$

$cA^{*}_{h-1,2x}$ または $cA^{*}_{h-1,2x+1}$ のいずれかが負値となった場合，$cD^{*}_{h,x}$ の値を，符号は変えずに，その絶対値を $cA^{*}_{h,x}$ 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 $cD^{+}_{h,x}$ を得る．

$$cD^{+}_{i} = \begin{cases} cD^{*}_{h,x} & (\text{if } cA^{*}_{h,x} \geq |cD^{*}_{h,x}|) \\ sign(cD^{*}_{h,x}) \cdot cA^{*}_{h,x} & (\text{otherwise}). \end{cases} \tag{3}$$

なお，$sign(\cdot)$ は，入力値の符号を返す関数とし，次式のように定義する．

$$sign(x) = \begin{cases} -1 & (\text{if } x < 0) \\ +1 & (\text{otherwise}) \end{cases} \tag{4}$$

そして，この非負精緻化を施した $cD^{+}_{h,x}$ の値を用いて改めて次式の処理を行うことにより，1 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直す（$h = H-1$, $x = 0$）．

$$cA^{+}_{h-1,2x} = cA^{*}_{h,x} + cD^{+}_{h,x} \tag{5}$$
$$cA^{+}_{h-1,2x+1} = cA^{*}_{h,x} - cD^{+}_{h,x} \tag{6}$$

ここで，$cA^{+}_{h-1,2x}$ および $cA^{+}_{h-1,2x+1}$ は，$cD^{*}_{h,x}$ に対して非負精緻化処理を施した結果である $cD^{+}_{h,x}$ に基づいて求められた近似係数の値を表す．非負精緻化の効果により，$cA^{+}_{h-1,2x}$，$cA^{+}_{h-1,2x+1}$ 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（スカラベクトル V^{+}）の値もすべて非負値となる．

3. 非負精緻化をともなう Privelet 法の演算効率化

上述した非負精緻化をともなう Privelet 法では，いったん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，すべての詳細係数の値が 0 となる．これは，非負精緻化処理が施されたノードでは詳細係数 $cD^{+}_{h,x}$ と近似係数 $cA^{*}_{h,x}$ との絶対値が等しくなることから，式 (5) または式 (6) で求められた近似係数（$cA^{+}_{h-1,2x}$ または $cA^{+}_{h-1,2x+1}$）のうちいずれかが必ず 0 になることに起因する．ここでたとえば，ノード $(h-1, 2x)$ に対応する近似係数 $cA^{+}_{h-1,2x}$ の値が，1 階層上の非負精緻化処理によって 0 となっていた場合，詳細係数 $cD^{*}_{h-1,2x}$ がもしも 0 以外の値をとると $cA^{+}_{h-1,2x} < |cD^{*}_{h-1,2x}|$ となり，またそこで非負精緻化処理が発生することとなる．そしてその結果，詳細係数 $cD^{*}_{h-1,2x}$ の値もまた 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・詳細係数ともにすべて 0 となり，その結果，当該ノードに連結されている出力値（秘匿結果のデータ値）もすべて 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，非負精緻化をともなう 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．

2.2.2 項で述べたとおり，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．**図 2** に示すように，最下層（リーフ）の層番号を $h = 0$，最上位層の層番号を $h = H$ とする．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（$0 \leq x < 2^{H-h} - 1$），このノード番号の順番にのっとって層内各ノードの訪問を行うこととする．

各ノードにおいて枝刈りを行う（演算を省略する）か否かは，近似係数 $cA^{+}_{h,x}$ の値が 0 であるか否かで判定する．上述したとおり，近似係数 $cA^{+}_{h,x}$ の値が 0 であれば，非負精緻化処理の結果，詳細係数 $cD^{+}_{h,x}$ の値は必ず 0 となり，さらに，一層下の近似係数も，ともに 0 となる．したがって，近似係数 $cA^{+}_{h,x}$ の値が 0 の場合には，計算せずとも，ノード (h, x) 配下のすべての近似係数および詳細係数，ひいてはリーフ値がすべて 0 となることが自明であり，これらの演算すべてを省略することができる．

加えて，HWT の場合，2 分木構造を持っていることから，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 h 層で x 番目に位置したノードに対応するノードは，第 $h-1$ 層では $2x$ および $2x+1$ 番目に位置することとなる．

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．1 点鎖線より上側が第 h 層を，下側が第 $h-1$ 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 $Z = (z_{1,0}, z_{1,1}, z_{1,2}, \ldots, z_{H,0})$ の要素とその値（初期値は 0）が記されている．

上述したような層間におけるノードの関係から，いま，ノード (h, x) で $cA^{+}_{h,x} = 0$ が検出され，演算が省略されるとき，そこに連結される 2 つの下位ノード $(h-1, 2x)$ および $(h-1, 2x+1)$ においても，$cA^{+}_{h-1,2x} = 0$ および $cA^{+}_{h-1,2x+1} = 0$ 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 倍になることが分かる．そこで，Top-down 精緻化処理において $cA^{+}_{h,x} = 0$ を検出したノードの演算省略ノード数メモリ $z_{h,x}$ に 1 を代入し，さらに，一層下るごとに層内ノード番号が 2 倍の位置の演算省

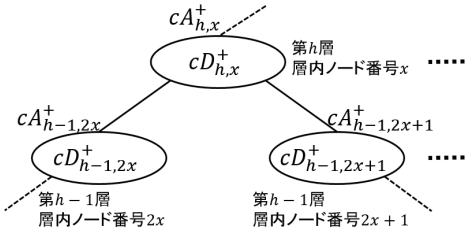

図 2 層番号と層内ノード番号

Fig. 2 The layer number and the node number in a layer.

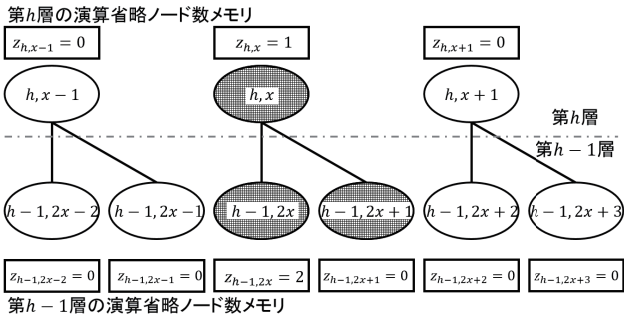

図 3 層間におけるノードの対応と演算省略ノード数メモリ

Fig. 3 The correspondence of nodes between layers and storage mechanism of the number of nodes to be omitted.

略ノード数メモリ $z_{h-1,2x}$ に $z_{h,x}$ の値を 2 倍して代入する処理 $z_{h-1,2x}=2z_{h,x}$ を行い，かつ，各層での処理を行う際に，演算省略ノード数メモリに格納されている値のノード数分だけ水平方向に演算を省略する．こうすることで，各層において $cA^+_{h,x}=0$ となる演算を一気に省略可能となることが期待できる．ただし，最下層（リーフ層；$h=0$）においてのみ，演算を省略する部分に相当するリーフにゼロ値を格納す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 処理部分の演算時間を計測して，演算量抑制効果を評価する．“(α) 枝刈りあり” と “(β) 枝刈りなし” との間の演算時間の差を “(β) 枝刈りなし” の演算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（1 km メッシュ）のデータに対して，1 次元 Haar Wavelwt 型の，非負精緻化をともなう Privelet 法を適用する．差分プライバシを規定するパラメータの値は $\epsilon=0.1$ とする．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（$2^9\times2^9$），(2) 四国（$2^8\times2^8$），(3) 関東（$2^8\times2^8$）の各エリアを切り出して評価用のデータとする．さらに，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（$2^8\times2^8$）も評価用データに加える．

なお，本評価を行うにあたって，2 次元状に配置された地域メッシュ人口データを，1 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 [13]，(d) ランダム方式という 4 方式によって 1 次元化を行い，各々に対して評価を行う．(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 次元化を行う．(b) ソート方式のデータは，(a) ラスター方式で得られる 1 次元データを，その値の大きい順（降順）に並べ替えたものとする．(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 写像を施したものとする．Morton 写像は，多次元空間から 1 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る 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とする．なお，(b) ソート方式は，秘匿前の知識を使用した並べ替えを行っているため差分プライバシ基準を満たさないが，他 3 方式と比較する目的で加えている．

評価には Intel Core i7-875K CPU（2.93 GHz），実装メモリ 4 GB 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，演算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，同一メッシュ数を持つ 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，ランダム方式以外で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[*2]．

各エリアの元データにおける 0 値の比率は，北海道が 95.0%，四国が 78.7%，関東が 61.2%，北海道 1/4 が 90.3%

[*2] 枝刈りによって短縮された非負精緻化をともなう逆 HWT 変換処理の演算時間は，たとえば，Morton 変換方式を採用した四国・関東・北海道 1/4 エリア（すべて $2^8\times2^8$ メッシュ）の場合 5.8〜8.7 [ms] の時間を要している（表 1 より）．同サイズのデータに対して単純な Laplace メカニズムを適用した際の演算時間を計測したところ，約 466.5 [μs] 程度であった．演算時間の観点のみで比較すれば，単純な Laplace メカニズムの演算時間の方が，今回評価したところの，枝刈りによって短縮された演算時間よりも格段に短い時間で処理を行うことができる．しかし，「非負制約の逸脱」や「部分和精度の劣化」，「疎データの密度急増」といった実用上の問題に対処することが求められる非負値データの場合には，単純な Laplace メカニズムでは対処ができず，非負精緻化をともなう Privelet 法が有効である．そして非負精緻化をともなう Privelet 法の演算効率化を行ううえでは，枝刈り処理の導入が有効なのである．

表 1 演算時間の計測結果
Table 1 The computation time and the reduction rate.

1 次元化方式	エリア	演算時間（100 ループの平均）		時間短縮率 [%]
		(α) 枝刈あり [ms]	(β) 枝刈なし [ms]	$(\beta-\alpha)/\beta$
(a) ラスター方式	北海道	23.7	45.1	47.5
	四国	10.2	11.3	9.9
	関東	10.6	11.2	5.8
	北海道 1/4	8.4	10.8	22.3
(b) ソート方式	北海道	4.3	44.5	90.4
	四国	3.2	11.2	71.7
	関東	5.2	11.1	53.3
	北海道 1/4	1.8	10.7	83.1
(c) Morton 方式	北海道	15.2	45.4	66.4
	四国	7.6	11.2	32.1
	関東	8.7	11.1	22.0
	北海道 1/4	5.8	10.7	45.3
(d) ランダム方式	北海道	35.9	44.7	19.8
	四国	14.9	11.3	-31.6
	関東	14.9	11.6	-28.2
	北海道 1/4	12.1	11.1	-9.0

であった．これと照らし合わせると，同一メッシュ数をもつ 3 エリア間での時間短縮率の高さは，0 値比率の高さ，すなわち，データの偏在性が高い順となっていることが分かる（付録 A.1 に補足的考察を示す）．

5. 考察

本章では，枝刈り処理が持つ基本的な性質について考察する．枝刈り処理では，秘匿後データが連続して 0 となる領域の演算を省略して効率化を行う．したがって，非ゼロ値の局在性，すなわち，データのスパース性が効率化の鍵を握る．そこでまず，データを 1 次元へ写像する方式の違いによって生ずるデータの局在性の差異に着目し，演算効率化効果の現れ方の違いについて 5.1 節で考察する．続いて，本手法によりもたらされる演算効率化の効果が，2 分木構造の連鎖に起因していることを検証するとともに，スパースなデータにみられるゼロ値の連続が演算効率化効果を高めるメカニズムについて 5.2 節で論ずる．

5.1 1 次元への写像方式と時間短縮率の関係

今回，2 次元データから 1 次元データへ写像する複数の方式による処理結果を評価している．4 つの写像方式の時間短縮率を見てみると，4.2 節でも述べたとおり，いずれの実装形態でもソート方式において時間短縮率が最も高く，次いで Morton 方式，ラスター方式となっており，ランダム方式において最も低いことが表 1 から分かる．これは，データの局在性が高いほど，本演算効率化手法による効率化の効果が高いことを示唆している．

ソート方式は，人為的にデータの局所性を最大に高めていることから，最大の効果がもたらされたといえる．この方式は，4.1 節でも触れたとおり，差分プライバシを満たさないことから実用に供するものではないものの，本演算効率化手法の適用効果に関する理論的な最大値を示している．

Morton 方式は，ソート方式に次いで効率化の効果が高い結果を示している．これは，元々のデータが「人口分布」という地理的偏在性が高いデータであり，Morton 写像が，写像元の 2 次元空間上で「近い」距離にある点を，写像先の 1 次元空間でもなるべく「近い」距離に配置するように写像する，局所性保存写像の性質を備えているためと考えられる．無論，元の 2 次元データに偏在性がなければ局所性保存写像を用いても効果はないが，地理空間データなどの実世界のデータはロングテール性を持つ，すなわち，偏在性が高い傾向にあるため，局所性保存写像の適用は人口分布に限らず，他の実データへの適用にも広く有効であることが期待される．

ラスター方式は，Morton 方式とランダム方式の間に位置している．これは，ラスター方式の場合，X 軸方向の局所性が保持され，Y 軸方向の局所性は基本的に保持されないことに起因していると考えられる．この結果は．ラスター方式の場合，データの局所性，すなわち，1 次元配列上における隣接データ間で差分値の小さい部分が，ランダム方式ほどではないものの，細分化された状態で大量に分布していることを示唆している．

ランダム方式は，ソート方式とは逆に「最悪」のケースに対する評価となっている．ランダム方式の評価結果の場合，データの局所性を失わせることで，本演算効率化手法による効率化の効果がほとんど得られないばかりか，悪化する場合さえあることを示している．このことは，データが持つ局所性を保存する形で 1 次元データへの写像を行ったうえで本演算効率化手法を適用することが重要であることを裏付けている．

5.2 重み付き枝刈り発生回数と時間短縮率の関係

次に，枝刈り処理の発生回数と時間短縮率の関係について考察を行う．2 章で述べたとおり，本手法は 2 分木構造を基盤としている．そのため，3 章にて説明したとおり，ある階層 h_0 に位置する 1 つのノードにおいて枝刈りが発生した場合，その 1 階層下（階層 h_0-1）で枝刈りが発生するノードは隣接した 2 ノードとなり，さらにその 1 階層下（階層 h_0-2）では，隣接した 4 ノードとなる．そしてこの枝刈りの連鎖は，最下層ノード（$h=1$）まで続くことになる．すなわち，ある階層 h_0 を起点とする 1 つの枝刈りから始まる枝刈り連鎖によって演算が省略されるノード数 ν_{h_0} は，起点の位置する階層によって決定されることとなり，次式で表される．

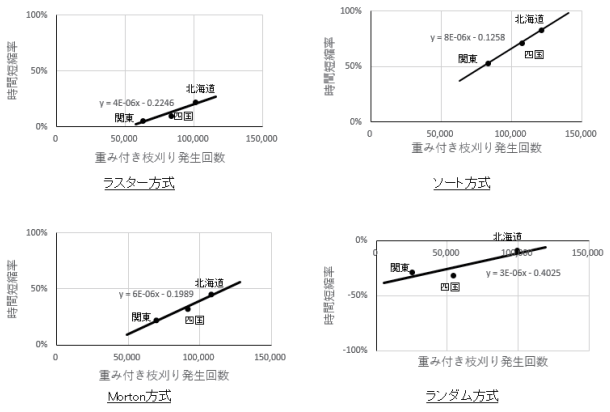

図 4 重み付き枝刈り発生回数と時間短縮率の相関

Fig. 4 Correlation between weighted pruning count and time reduction rate.

$$\nu_{h_0} = \sum_{i=1}^{h_0} 2^{h_0 - i}$$

そこで，任意の階層 h において発生した枝刈りの起点数 s_h を階層ごとに数えておき，階層 h を起点とする 1 つの枝刈り連鎖で演算が省略されるノード数 ν_h との積を階層ごとに求め，さらにその総和を求めることで，演算が省略されたノードの総数を求めることができる．

$$S_w = \sum_{h=1}^{H} (s_h \cdot \nu_h) = \sum_{h=1}^{H} \left(s_h \cdot \sum_{i=1}^{h} 2^{h-i} \right)$$

もしもここで考察したとおり，この重み付き枝刈り発生回数が，演算が省略されたノード数に対応しているのであれば，時間短縮率との間に，正の相関を示すことが期待できる．そこで，両者の関係を実験的に評価し，重み付き枝刈り発生回数の値が，演算が省略されたノード数に対応すると考えることの妥当性を確認する．**図 4** に，重み付き枝刈り発生回数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のあることが分かる．

以上の考察結果をふまえると，より高い階層を起点とする枝刈りが発生するほど，そこから派生する形で演算が省略されるノード数の値が大きくなる，すなわち，演算が省略されるノード数が増えると考えられる．HWT においては，0 値が広い範囲で連続している場合，上位階層まで近似係数 $cA_{h,x}$ の値が 0 値となり，詳細係数 $cD_{h,x}$ への Laplace ノイズの付加によって逆 HWT の過程で $cA^*_{h,x}$ に負値が発生し，その結果非負精緻化が起こり，枝刈りが発生する可能性が高くなる．したがって，0 値が広く分布するような，スパースなデータに本手法を適用した場合に，より大きな時間短縮率が期待できる．

6. おわりに

本稿では，非負精緻化をともなう Privelet 法について，逆 Wavelet 変換（Top-down 精緻化）処理の性質に着目した枝刈り実装法を適用してその性能面に対する評価を行い，性能特性を多角的に明らかにした．評価の結果，1 次元への写像方式と時間短縮率の関係の評価から，データが持つ局所性を保存する形で 1 次元データへの写像を行ったうえで本演算効率化手法を適用することの重要性が見い出された．また，重み付き枝刈り発生回数と時間短縮率との関係から，本手法は，適用されるデータがスパースであるほど，より大きな演算効率化効果が得られることが示唆された．なお，本研究の意義を明確にするうえで，演算の高速化が必要とされる用途に関する考察を付録 A.3 に示しておく．

非負精緻化をともなう Privelet 法の性能面に対する特性評価は，従来研究では行われておらず，これらの知見は，本稿による貢献である．今後は，本枝刈り実装方式の振舞いをさらに詳細に分析する過程を通じて，より効率的な枝刈り手法についての検討を進めて行く．

謝辞 シミュレーションの一部に協力していただいた大加瀬稔氏，飯塚皇太氏に感謝する．なお，本研究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）による助成を受けて行われたものである．

参考文献

[1] OECD 理事会：勧告 8 原則, OECD（オンライン），入手先〈http://www.soumu.go.jp/main_sosiki/gyoukan/kanri/oecd8198009.html〉（参照 2019-03-24）.
[2] Misuraca, G., Mureddu, F. and Osimo, D.: Policy-Making 2.0: Unleashing the Power of Big Data for Public Governance, *Open Government, Public Administration and Information Technology*, Vol.4, pp.171–188, Springer (2014).
[3] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Longhurst, J., Nordholt, E.S., Seri, G. and Wolf, P.-P.: *Handbook on Statistical Disclosure Control*, Statistics Netherlands (2010).
[4] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E.S., Spicer, K. and Wolf, P.-P.: *Statistical Disclosure Control*, John Wiley & Sons (2012).
[5] 統計センター：統計データ開示制御に関する用語集（改訂版），製表関連国際用語集，No.2 (2005).
[6] 瀧 敦弘：集計表におけるセル秘匿問題とその研究動向，統計数理，Vol.51, No.2, pp.337–350 (2003).
[7] Fung, B.C.M., Wang, K., Chen, R. and Yu, P.S.: Privacy-preserving Data Publishing, *ACM Computing Surveys*, Vol.42, No.4, pp.1–53 (2010).
[8] Sweeney, L.: k-anonimity: A model for protecting privacy, *Intl. J. Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol.10, No.5, pp.557–570 (2002).
[9] Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M.: l-diversity: Privacy Beyond k-anonymity, *ACM Trans. Knowledge Discovery from Data* (*TKDD*), Vol.11, No.1 (2007).
[10] 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).
[11] 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).
[12] 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 PODS '07*, pp.273–282 (2007).
[13] 寺田雅之，鈴木亮平，山口高康，本郷節之：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，Vol.56, No.9, pp.1801–1816 (2015).
[14] Cormode, G., Procopiuc, M., Srivastava, D. and Tran, T.: Differentian Private Publication of Sparse Data, *Proc. Intl. Database Theory* (*ICDT 2012*) (2012).
[15] Xiao, X., Wang, G. and Gehrke, J.: Differential Privacy via Wavelet Transforms, *Proc. 26th Intl. Conf. Data Engineering* (*ICDE 2010*), pp.225–236 (2010).
[16] Xiao, X., Wang, G., Gehrke, J. and Jefferson, T.: Differential Privacy via Wavelet Transforms, *IEEE Trans. Knowledge and Data Engineering*, Vol.23, No.8, pp.1200–1214 (2011).
[17] 本郷節之，手塚理貴，寺田雅之，稲垣　潤：Top-down 精緻化を伴う Privelet 法における演算効率化手法の検討，マルチメディア，分散，協調とモバイルシンポジウム（DICOMO2018）講演論文集，pp.460–466 (2018).
[18] 本郷節之，大加瀬稔，手塚理貴，寺田雅之，稲垣　潤，鈴木昭弘：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 III，*2019 Symposium on Cryptography and Information Security*, pp.1–8 (2019).
[19] 総務省統計局：地域メッシュ統計の結果資料と提供方法，入手先 ⟨https://www.stat.go.jp/data/mesh/teikyo.html⟩（参照 2019-07-31）.
[20] 国土交通省：ETC2.0 データを活用した新たな民間サービスの実用化に向けパーク 24 株式会社とデータ配信に関する協定（第 1 号）を締結，入手先 ⟨http://www.mlit.go.jp/report/press/road01_hh_001144.html⟩（参照 2019-07-31）.
[21] 佐治秀剛，田中良寛，鹿野島秀行，鹿野島秀行：ETC2.0 プローブを活用した分析事例，土木技術資料，Vol.57, No.5, pp.22–25 (2015).
[22] 大口　敬：渋滞のメカニズムおよび渋滞対策の全体像，高度情報通信ネットワーク社会推進戦略本部第 2 回 ITS に関するタスクフォース資料，No.2, pp.1–24 (2010).
[23] 寺田雅之，外山敬祐：リアルタイム人口統計と AI 渋滞予知，電子情報通信学会技術報告，Vol.IEICE-118, No.305 (MoNA), pp.67–68 (2018).
[24] 寺田雅之，赤塚裕人，永田智大，仲西哲志：東京湾アクアラインの渋滞を「AI 渋滞予知」で回避する，NTT DOCOMO テクニカル・ジャーナル，Vol.27, No.2, pp.26-33 (2019).
[25] NTT ドコモ：モバイル空間統計に関する情報，入手先 ⟨https://www.nttdocomo.co.jp/corporate/disclosure/mobile_spatial_statistics/⟩（参照 2019-07-31）.

付　　録

A.1 ゼロデータの比率と時間短縮率との関係

データのスパース性を論じるうえで，5.2 節で述べたような疎密の観点のほかに，そもそもゼロ値以外のデータが少ないことに起因するスパース性の視点についても考慮する必要がある．そこで，時間短縮率の変化について，ゼロ値含有比率との関係からも補足的考察を加える．

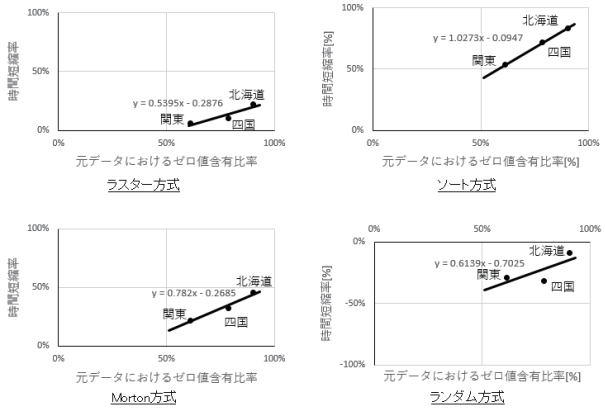

図 **A·1**　時間短縮率対元データにおけるゼロ値含有比率
Fig. A·1　Time reduction rate versus zero value ratio in original data.

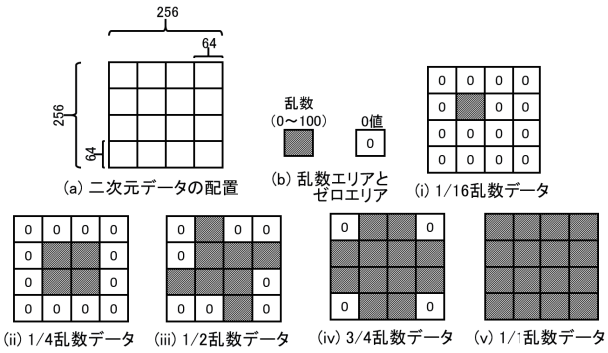

図 **A·2**　人工的に作られた密度の異なる評価用パターン
Fig. A·2　Artificial patterns with different density used for evaluation.

図 **A·1** に元データにおけるゼロ値含有比率と時間短縮率との関係を示す．グラフから，元データにおけるゼロ値含有比率と時間短縮率との間に高い正の相関があることが分かる．このことは，元データにおけるゼロ値含有比率が高いことによってスパース性がもたらされるような場合においても，枝刈り処理によってより大きな演算の効率化が期待できることを示唆している．

A.2 密度の異なるパターンを用いた補足的評価

A.2.1 評価の目的

4 章で実施した評価は，国勢調査に基づく地域メッシュ人口（1 km メッシュ）のみを対象としたものである．しかも，同一エリアサイズのデータは，北海道 1/4，四国，関東の 3 エリアのみとなっている．そこで，これとは異質な人工パターンを用いた追評価を行い，評価結果の妥当性の補強を試みる．

A.2.2 評価方法

図 **A·2** に，追評価に使用した 2 次元パターンを示す．まず，256 × 256 要素からなる正方パターンを 64 × 64 要素からなる 16 のブロックに分け，0 値エリア比率の異なる

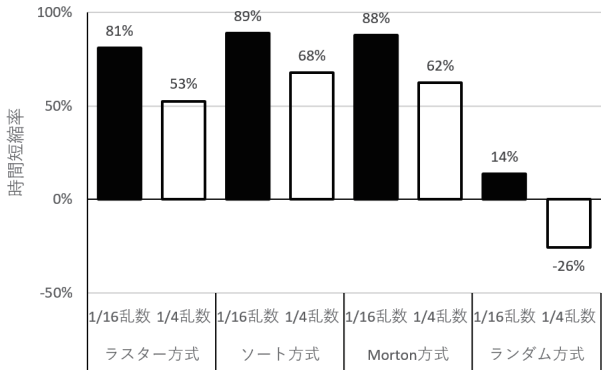

図 A·3 人工的パターンに対する時間短縮率

Fig. A·3 Evaluation results obtained from the artificial patterns.

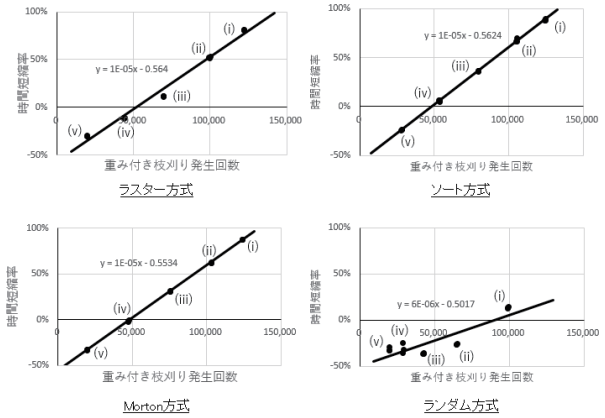

図 A·4 人工的パターンにおける時間短縮率対重み付き枝刈り発生回数特性

Fig. A·4 Evaluation results obtained from the artificial patterns.

(i)〜(v) のような 5 種類のパターンを用意した．そしてそれらを (a) ラスター方式，(b) ソート方式，(c) Morton 方式，(d) ランダム方式により変換し，得られた 1 次元データを追評価用データとした．各変換方式とも，Laplace 摂動値を変えて 3 回ずつ計測を行った（計 15 パターン）．なお，計算機環境は 4.1 節で述べたものと同じものを使用した．

A.2.3 評価結果

図 A·3 に評価結果を示す．図には，4 章で使用したメッシュ人口データと同程度のゼロ値比率を持つパターンの結果を示している．1 次元への写像方式による違いについては，時間短縮率の大きさが，ソート方式，Morton 方式，ラスター方式，ランダム方式の順となっており，メッシュ人口データでの評価結果と同じ傾向を示している．

さらに，**図 A·4** に，人工的パターンにおける時間短縮率対重み付き枝刈り発生回数のグラフを示す．ここでは，ゼロ値比率の高いパターンほど，重み付き枝刈り発生回数が多くなっている．グラフから，本追評価においても，メッシュ人口データでの評価と同様に，重み付き枝刈り発生回数に比例する形で，演算時間が短縮されていることが分かる．

A.3 演算の高速化が必要とされる用途の考察

本稿では，実験的な評価を現実的な時間で行うことができるサイズのデータを使用している．しかし，本技術の適用が期待，または，想定される用途においては，実験に使用したものの何倍ものデータサイズを対象とするケースも考えられる．さらに用途によっては，非常に短時間での処理を行うことが有効なケースも考えられる．

はじめに，データサイズについて考察する．データサイズを左右する要因としては，「①対象領域の広さ」，「②メッシュの細かさ」，「③各セルに対応する属性数」が考えられる．まず，①対象領域の広さについてであるが，今回の演算時間計測には，北海道を除いて $2^8 \times 2^8$ からなるメッシュサイズのエリアを対象に行っているが，日本全国を対象にする場合には，メッシュサイズは $2^{11} \times 2^{11}$ となり，64 倍程度のデータ量となる．当然のことながら，より国土面積の広い地域に適用する際には，データ量もさらに膨らむこととなる．次に，②メッシュの細かさに関して見てみると，今回は 1 km メッシュといわれる「基準地域メッシュ（第 3 次地域区画）」を使用しているが，総務省統計局では，8 分の 1 地域メッシュ（125 m メッシュ）コードまで定義されており [19]，64 倍程度のデータ量となる．こちらの場合にも，より細かい粒度のメッシュが求められる用途に適用する場合には，データ量がさらに膨らむ．一方，③各セルに対応する属性数に関しては，たとえば対象が人の場合には性別や年齢層や居住場所，たとえば対象が自動車の場合には車種や車両区分や登録運輸支局などさまざまな属性が考えられる．そしてこれらの組合せによって，データ量は数十倍から数百倍になることもありうる．

続いて，データの処理に要する時間について考察する．今回の評価で使用した人口統計のような用途であれば，その処理には数日から数十日程度あるいはそれ以上の時間的猶予があると考えられる．しかし，現在注目を集めているビッグデータの活用をふまえると，数時間，場合によっては，数十分の間に何らかの分析や判定の結果を出すことが求められるケースも想定される．

たとえば交通量分析に関して，ETC2.0 プローブ [20] を活用して渋滞検出を行ったり，孤立地域検出を行ったりする取り組みが始まっている [21]．交通渋滞は 15 分から 20 分程度で発生するケースもみられ [22]，渋滞の発生をいち早く検出し，ドライバーに通知したり，実用化が迫りつつある自動運転における経路選択に応用したりするには，この時間よりも十分に短い時間での判定が必要となる．しかもプライバシ保護技術は一連の分析・判定処理の前段階で行われる前処理のさらに一部ともいえ，判定・判断に要する時間よりも十分に短い時間での処理が必要となる．

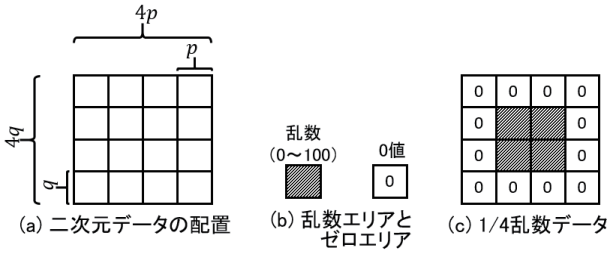

図 A·5 サイズ変化にともなう処理時間変化の計測に使用された評価用人工パターン

Fig. A·5 Artificial pattern used to measure changes in processing time with size changes.

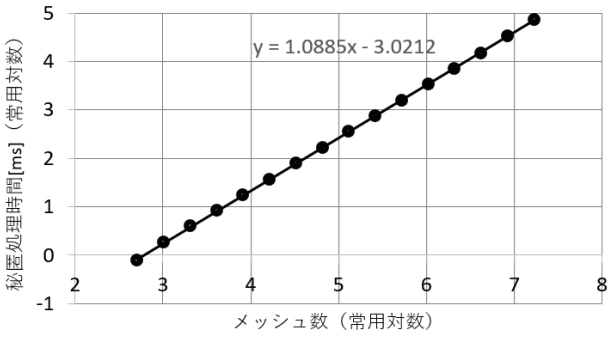

図 A·6 データサイズによる秘匿処理時間の変化

Fig. A·6 Relationship between the size of data and the processing time.

人の移動の分析に関しても，短時間での演算処理が必要となるケースが想定される．たとえば，人気遊戯施設や人気イベント会場，その周辺駅などにおいては，人の移動に合わせて入場ゲートや改札口の人員配置を行ったり，周辺道路での誘導員の配置を行ったりするような用途が考えられる．また，イベントなどの終了後，人の集中している場所に重点的にタクシーを配車するよう調整したり，できるだけ人が多いエリアを通るように選挙カーの経路選択を行ったりするようなケースも想定できる．さらに，携帯電話ネットワークの運用データからほぼリアルタイムに作成される「リアルタイム人口統計」に AI 技術を適用し，交通渋滞の発生やその規模・時間帯などを予測する技術もすでに実証実験の段階に入っている [23], [24]．こうした用途の場合，人や車の移動に合わせた，リアルタイムに近い短時間での処理が求められる．そして，この「リアルタイム人口統計」を提供するサービスは，一部ですでに開始されている [25]．

本稿では，非負精緻化をともなう Privelet 法の逆 Wavelet 変換部分のみを対象に時間計測を行っているが，この秘匿手法を実際に適用するうえでは，順方向の Wavelet 変換や摂動処理など，逆 Wavelet 変換部分以外の処理にも時間を要することになる．そこでこの秘匿処理全体に要する時間とデータサイズとの関係について調べてみた．

時間計測には，**図 A·5** (c) に示す 1/4 乱数データを使用した．図 A·5 (a) 中の p，q はブロックの 1 辺のメッシュ数を表している．ここで，p の値は $(8, 16, 32, 64, 128, 256, 512, 1024)$ なる 8 段階とし，一方，q の値は $(p, p/2)$ として生成した，計 16 サイズのパターンを用いた秘匿処理時間計測を行った．得られたデータを常用対数空間（両対数）にマッピングしたグラフを**図 A·6** に示す．グラフから，データが直線状に分布していることが分かる．このデータ分布に対する近似直線を求めたところ，$y = 1.0885x - 3.0212$ となった．ここで，上述した「①対象領域の広さ」,「②メッシュの細かさ」を参考にすると，全国を対象とした「基準地域メッシュ」のサイズは 2,048 × 2,048 程度，さらに，「8 分の 1 地域メッシュ」を使用するケースを考えると，そのメッシュサイズは，16,384 × 16,384 程度にもなりうる．これを上記近似直線の式にあてはめると，以下の時間を要することが想定される．

$$10^{1.0885\times(\log_{10} 16{,}384^2)-3.0212} = 1{,}424{,}270\,[\mathrm{ms}]$$

これは，約 24 分に相当する時間である．これに加えて，さらに，「③各セルに対応する属性数」の考慮が必要となり，属性ごとの処理が発生する場合も想定される（例：属性を組み合わせた場合の数が N のとき，処理対象となるデータ数も N 倍化）．そしてそれ以上に，この秘匿処理は，さまざまな用途に使用するための，いわば前処理であって，各用途ごとに必要となる処理は，これに加えて別途発生することになる．

本節の前半でで述べたような，リアルタイム，またはそれに準ずるような用途を想定した場合，あるいは，プライバシの安全性が確保された各種データがリアルタイムに近い状態で利用することが可能になったことで創生される新たな用途をも想定した場合，少しでも演算の効率化を図ることには十分な価値があると考えられる．

以上述べたように，非負精緻化をともなう Privelet 法を，今後展開が期待される多様なサービスに適用するうえで，本稿で評価を行った枝刈り処理のような手法を用いて効率化を図ることがきわめて効果的であると考えられる．

推薦文

DICOMO2018 の発表論文の中で特に評価が高かったため．

（マルチメディア，分散，協調とモバイル（DICOMO2018）シンポジウムプログラム委員長　福澤寧子）

本郷 節之 （正会員）

1984 年岩手大学大学院工学研究科修士課程修了．同年日本電信電話公社入社．1987 年国際電気通信基礎技術研究所（ATR）へ出向．1991 年 NTT へ復帰．1999 年 NTT ドコモへ転籍，セキュリティ方式研究室長．2010 年北海道工業大学（現北海道科学大学）教授に着任，現在に至る．モバイルセキュリティならびにプライバシ保護技術の研究開発に従事．博士（工学）．著書『ネットワークセキュリティ』（共著）他，2015 年度論文賞，2017 年度大会優秀賞，DICOMO2018 優秀論文賞受賞，電子情報通信学会，IEEE 各会員．

寺田 雅之 （正会員）

1995 年神戸大学大学院工学研究科修士課程修了．同年日本電信電話（株）入社，2003 年（株）NTT ドコモへ転籍，2008 年電気通信大学大学院博士後期課程修了，2009 年より NTT ドコモ．博士（工学）．情報セキュリティ技術，プライバシ保護技術，大規模データに基づく人口推計技術および交通予測技術の研究開発に従事．DICOMO2014 最優秀論文賞，2015 年度論文賞，山下記念研究賞受賞．電子情報通信学会会員．

鈴木 昭弘 （正会員）

2012 年北海道工業大学大学院工学研究科博士後期課程修了．博士（工学）．同年株式会社ジャパンテクニカルソフトウェア入社．以来システムエンジニアに従事．2018 年北海道科学大学助教，現在に至る．ソフトウェア工学等の研究に従事．電子情報通信学会，プロジェクトマネジメント学会会員．

稲垣 潤

1996 年北海道大学工学部卒．2001 年同大学大学院博士後期課程修了．博士（工学）．同年北海道東海大学講師．2008 年北海道工業大学講師，准教授，北海道科学大学准教授を経て，2018 年より同大教授．現在に至る．ソフトコンピューティングを用いた最適化手法，リハビリテーション支援システム，運動学習支援等の研究に従事．電子情報通信学会，IEEE，臨床歩行分析研究会，日本運動・スポーツ科学学会各会員．