情報処理学会論文誌 Vol.61 No.9 1458–1471 (Sep. 2020)

推薦論文

非負精緻化をともなうPrivelet法における演算効率化手法の性能向上

本郷 節之[1] 寺田 雅之[2] 鈴木 昭弘[1] 稲垣 潤[1]

受付日 2019年10月29日, 採録日 2020年6月1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．筆者らは以前，Top-down 精緻化の性質に着目した枝刈り実装法（水平型）を提案した．本報告では，先の提案とは異なる実装法（垂直型）を新たに提案する．さらに，先に提案した実装法との間での，演算効率化効果の比較評価も試みる．

キーワード：プライバシ保護，差分プライバシ，Wavelet 変換，非負制約

Improvement of the Computational Efficiency for Privelet with Non-negative Refinement

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

Received: October 29, 2019, Accepted: June 1, 2020

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. We have proposed a pruning implementation method - the horizontal type - focused on characteristics of the top-down refinement. In this paper, we propose a new implementation method - the vertical type - different from the previous proposal. Additionally, we try to compare and evaluate the efficiency improvement effect between these two implementation methods.

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

1. はじめに

1.1 プライバシ保護の重要性

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付いた

[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

本論文の内容は 2019 年 3 月の第 84 回 CSEC 研究発表会にて報告され，同研究会主査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．

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]．

1.2 高速処理の必要性と対象データ

近年，たとえば交通量分析において，ETC2.0 プローブ [3] 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 [4]．交通渋滞は 15 分から 20 分程度で発生するケースも見られ [5]，渋滞の発生をいち早く検出し，ドライバに通知したり，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さらにまた十分に短い時間での処理が必要となる．災害による孤立地域検出に至っては，いち早く孤立地域を検出し救援体制を確立することが人命にも関わることから，少しでも短時間で処理を行うことが重要であると考えられる．

また，人々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，群衆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．さらに，イベントなどの終了後，人々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，時々刻々と変化する人の流れに応じて，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．こうした用途の場合，人の移動に合わせた，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，人の移動の統計的情報の提供サービスは，一部ですでに開始されている [6], [7], [8]．本稿に示した，2^{18} 個のデータを対象にした小規模な実験において短縮している演算時間はわずか数十ミリ秒程度ではあるが，提案手法を，より大きなデータセットやより多様な属性を持った対象に適用した場合には，はるかに大きな演算効率化効果が期待できる．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を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．

そして，様々な産業活動で計量・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．

1.3 本稿の構成

我々は以前，差分プライバシ基準を満たすプライバシ保護技術であるところの，非負精緻化をともなう Privelet 法（3 章で詳述）を対象に，その Top-down 精緻化（非負精緻化をともなう逆 Wavelet 変換）部分の性質に着目した枝刈り実装法（水平型）を提案した [9], [10]．この手法は，枝刈りによる演算の省略を行うことで効率化を図るものであり，データへの依存性はあるものの，北海道のメッシュ人口データを使った評価実験において，演算時間を 1/3 程度にまで抑制できることが確認できた [11]．しかし我々は，この枝刈り処理の動作の分析を深めることで，先の実装法とは異なる，新たな実装法（垂直型）の可能性を見い出した．本報告では，この，新たな枝刈り実装法について詳しく述べるとともに，先に提案した実装方法との比較評価を行う．

まず 2 章では，先に提案された水平型実装法も含めた，

本研究の背景となる主要な先行研究について概説する．次に 3 章では，Privelet 法，ならびに，非負精緻化をともなう Privelet 法について詳しく説明する．続いて 4 章では，非負精緻化をともなう Privelet 法の演算効率をより向上させるために今回提案する枝刈り実装法（垂直型実装）について述べる．さらに 5 章では，枝刈り処理による演算効率化効果の評価方法とその結果を示す．そして 6 章では，各種条件と時間短縮率の関係や垂直型実装と水平型実装との時間短縮率の比較，関連する演算効率化手法との差異についての考察を行う．

2. 先行研究

2.1 差分プライバシ基準と Laplace メカニズム

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）[12], [13] 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来た [14], [15]．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 [16]，k–匿名性 [17] や l–多様性 [18]，m–不変性 [19] など，様々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 [20], [21] 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．

いま，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 に対して次式の関係が成立するとき，この問合せ M は ϵ–差分プライバシ基準を満たす．

$$\frac{Pr[M(D_1)=t]}{Pr[M(D_2)=t]} \leq e^{\epsilon}$$

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 $Lap(\lambda)$ から成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 [22], [23]．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．

2.2 非負精緻化をともなう Privelet 法の有用性

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」，「疎データの密度急増」といった問題への対処が必要となる [23]．「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，「疎データの密度急増」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．

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 [21], [24], [25], [26]．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．たとえば Barak らの手法 [21] は，非負制約の逸脱への対処は実現しているものの，Privelet 法のように部分和精度劣化を抑制する直接的なメカニズムは備えておらず，さらに疎データの密度急増への対処も行われていない．また，Cormode らの手法 [24] は，非負制約の逸脱への対処，および，疎データの密度急増の抑制は実現しているものの，部分和精度劣化への対処は実現できていない．一方，Xiao らが提案した Privelet 法 [25], [26] の場合には，部分和により摂動が相殺されるという，精度劣化を原理的・直接的に抑制する性質を有しているものの，非負制約の逸脱への対処と疎データの密度急増には対処できていない．

また，Laplace メカニズムが非負制約を満たさないことを解決する手法としては，確立単体（probability simplex）もしくは正規単体（canonical simplex）への射影 [27], [28] を応用し，Laplace メカニズムの出力に対して，事後処理としてこの計算を行う手法が提案されている [29]．しかし，これらの手法では，部分和精度の問題は解決されていない．一方，ヒストグラム公開を対象とした Xu らの手法 [30] や Acs らの手法 [31] は，いずれも差分プライバシ基準を満たし，かつ，精度に優れた集計データを得る方法を提案している．しかし，ここで比較されている Privelet 法は，非負制約を満たさない（元々の）Privelet 法であり，これらの手法から得られる出力は非負制約を満たしていない．

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 [23]．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，Privelet 法が有する，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，近年の他の研究を見ても，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」を検討対象とする．

[*1] 文献 [23] 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．

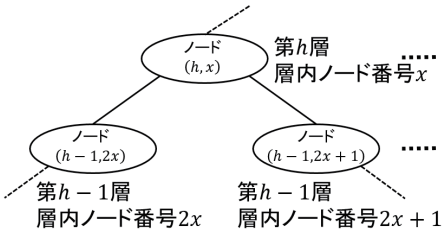

図 1 階層番号と層内ノード番号
Fig. 1 The layer number and the node number in a layer.

2.3 非負精緻化をともなう Privelet 法の演算効率化（水平型実装）

1.2 節において述べたとおり，リアルタイムに準ずる性能が求められるようなビッグデータに基づく大規模集計データを処理するうえでは，演算処理の高速化，演算の効率性がきわめて重要になって来る．そこで我々は，先に，非負精緻化をともなう Privelet 法の演算効率化手法を開発した [9], [10]．これは，非負精緻化処理が隣接したノード間で連続して発生するという性質に着目した演算効率化手法である．

Xiao らが提案した二分木に基づく Privelet 法は，1 次元の入力スカラベクトルに対して，Haar Wavelet 変換（HWT）を施し，そこで得られた Wavelet 係数に対して Laplace 分布に従うノイズを付加したうえで逆 HWT を施すことにより秘匿された出力データを得る手法である（3.1 節で詳述）．その際，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 1 に，HWT における階層番号と層内ノード番号の例を示す．丸印はノードを，直線は木構造を表している．ここで，$h = 0, 1, \cdots, H$ は，木構造における階層番号を，$x = 0, 1, \cdots, 2^{H-h} - 1$ は階層内ノード番号をそれぞれ表す．説明のため，最下層（リーフ）の階層番号を $h = 0$，最上位層の階層番号を $h = H$ 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（3.2.3 項で詳述）を繰り返す．

逆 HWT 処理において非負精緻化を行った場合，非負精緻化が発生したノードの片側子ノードおよびそこに連結されたすべての下層ノードの Wavelet 係数（近似係数 [23]）の値は 0 へと精緻化される．この性質をふまえれば，当該ノードにおける演算（非負精緻化をともなう逆 HWT 処理）を行わずとも，演算結果が得られることから，このノードの演算を省略することができる．しかもこの演算省略が可能なノードは，HWT が持つ木構造の性質から，下層へ行くほど連続（1 階層下るごとにノード数は倍増）して発生することになる．そこで，演算省略ノード数メモリを用いて連続して演算の省略が可能なノード数を管理すれば，各層内の演算処理において，連続する省略可能なノードの演算を一気に削減することができる．

図 2 に，階層間におけるノードの対応と演算省略ノー

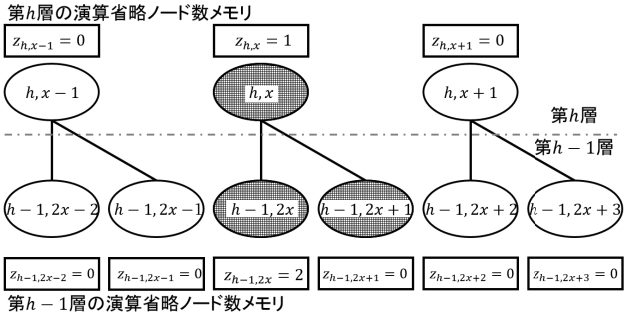

図 2 層間におけるノードの対応と演算省略ノード数メモリ
Fig. 2 The correspondence of nodes between layers and storage mechanism of the number of nodes to be omitted.

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．

層間におけるノードの関係から，いま，ノード (h,x) で 0 値をとる近似係数が検出され，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，0 値をとる近似係数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．そこで，Top-down 精緻化処理において 0 値をとる近似係数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，各層において近似係数が 0 値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．

この手法によれば，出力に近い下位の層に行くほど演算が省略される連続したノード数が増加し，著しい効率化を実現することが期待できる．しかしその反面，演算省略ノード数メモリの操作によって生ずるオーバヘッドが避けられないことから，逆に演算効率を大きく低下させてしまうケースも発生するという問題があった．そこで本稿では，演算省略ノード数メモリの管理を必要としない，新たな演算効率化実装法（垂直型実装：4.2 節で詳述）を提案する．

3. 非負精緻化をともなう Privelet 法

Xiao らが提案した二分木に基づく Privelet 法 [25] は，長さ $n=2^H$ のスカラベクトルデータ $V=(v_1,\ldots,v_n)$ 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 [23] 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 HWT 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．すなわち，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 [23] 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採用する．

3.1 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を得る．同様に，Haar 分解処理を再帰的に繰り返すことで，最終的に，1 つの近似係数と，$n-1$ 個の要素から成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．

3.2 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．

3.2.1 Laplace メカニズムの適用

Laplace メカニズムの適用では，近似係数 $cA_{H,0}$

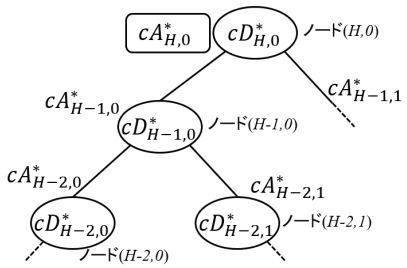

図 3 逆 Haar Wavelet 変換
Fig. 3 The inverse Wavelet transform.

と，詳細係数ベクトル cD を構成するすべての要素 $(cD_{1,0}, \cdots, cD_{H-1,0}, cD_{H-1,1}, cD_{H,0})$ に対して，次式に従って 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 [23].

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

ここで，$h = 1, 2, \cdots, H$ 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．

3.2.2 逆 HWT の適用

逆 HWT の適用については，**図 3** 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 3 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一階層下に位置する 2 つの近似係数を求める（$h = H$, $x = 0$）．

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

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，一階層下に位置する 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，一階層下に位置する 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 Wavelet 変換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．

3.2.3 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，近似係数の値が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一階層上の $cD^*_{h,x}$ の値を精緻化する．

まず，ノード $(H-1, 0)$ 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（$h = H-1$, $x = 0$）．

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

$cA^*_{h-1,2x}$ 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．

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

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

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

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，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直す（$h = H-1$, $x = 0$）．

$$cA^{+}_{h-1,2x} = cA^{*}_{h,x} + cD^{+}_{h,x} \quad (5)$$

$$cA^{+}_{h-1,2x+1} = cA^{*}_{h,x} - cD^{+}_{h,x} \quad (6)$$

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．

4. 提案手法（垂直型実装）

4.1 枝刈り処理の概要

3章で述べたとおり，非負精緻化をともなう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，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(5)または式(6)いずれかの近似係数の値（$cA^{+}_{h-1,2x}$ または $cA^{+}_{h-1,2x+1}$）が必ず0になることに起因する．そして，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．この演算効率化過程が“枝刈り”と呼ばれる処理である．

2.3節で述べたとおり，この枝刈りを実現する従来手法である水平型実装[10]は，演算省略ノード数メモリを用いることで，水平方向に連続する枝刈り対象ノード（近似係数の値が0のノード）の演算を，一気に省略するアルゴリズムとなっている．しかしその反面，このアルゴリズムでは，(i) 演算対象ノードの近似係数の値が0のとき，演算省略ノード数メモリの，対応する位置の値が0であればそこに1を書き込む操作や，(ii) 演算省略ノード数メモリの値に従って演算対象のノード位置を変化させる操作，さらには，(iii) 処理対象の階層が下がるたびに演算省略ノード数メモリの値を2倍する操作といった，枝刈りを導入しなければ本来不要な筈の，演算省略ノード数メモリ操作というオーバヘッドをともなっている．しかもこのオーバヘッドが，むしろ演算効率を大きく低下させてしまうケースも発生するという問題があった．

たとえば局在性が著しく低く，ゼロ値がまばらに点在しているようなデータの場合には，非負精緻化処理，そして，枝刈り処理が，低位の階層で散発的に発生することとなり，その結果，演算省略ノード数メモリの書き換え回数が大きく増え，むしろ演算量が増大してしまうような事態も生じうる．このことは，先行研究[10]における，水平型枝刈り処理の導入による時間短縮効果の評価において，ランダム一次元化方式の場合，枝刈り処理の導入によって，逆に演算時間が増大してしまっている例（時間短縮率が負値）からも分かる．

そこで本稿では，演算省略ノード数メモリの管理を必要としない，新たなアルゴリズムを提案する．具体的には，演算を，同一階層内（水平方向）を優先して行うのではなく，階層間方向（垂直方向）を優先して行うアルゴリズムである．これは，単に演算の手順を入れ替えたものではなく，1つのノードで枝刈りが発生するとその配下の演算がすべて省略可能になるとなるという非負精緻化の性質を，木構造それ自体が有する構造上の特質を活かして効率的に実現する新たなアルゴリズムである．しかもこのアルゴリズムの場合，演算省略ノード数メモリが必要ないことから，演算省略ノード数メモリに起因するオーバヘッドを排除することができるという利点も期待できる．次節では，この垂直型の処理について，詳しく説明する．

4.2 垂直型実装

3.2節で説明したとおり，非負精緻化をともなう逆HWT処理は，各層の各ノードにおいて，自ノードの近似係数 $cA_{h,x}$ の値と詳細係数 $cD_{h,x}$ の値から，直下ノードの近似係数 $cA_{h-1,2x}$ および $cA_{h-1,2x+1}$ の値を求めるものである．これらの処理は，同一層内では他のノードとの相互作用が存在せず，また，下層側から見て層間相互作用を有する上位ノードは1つ（すなわち複数の上層側ノードとの相互作用は存在しない）という構造を持っている．この性質に着目すると，この処理は階層ごとに行う必要はなく（すなわち同一層内のすべてのノードの処理を終えてから次の階層内のノードの処理に移行する必要はなく），層間結合の範囲のみを対象に深さ方向（層間方向）に処理を進めて行って構わないことになる．深さ方向に処理を進めることで，水平方向の演算省略ノード数を管理する必要がなくなり，演算省略ノード数メモリは不要となる．これにともない，前節で述べたオーバヘッドを解消することができ，枝刈りによる演算効率化性能のさらなる向上が期待できる．そこで，この処理を，最上位層から深さ方向に進めて行くことを考える．

このような処理は，再帰呼出し方式での実装に適している．再帰呼出しは，手続き型プログラミングにおいて，あるまとまった処理ブロックをモジュール化（例：関数化）

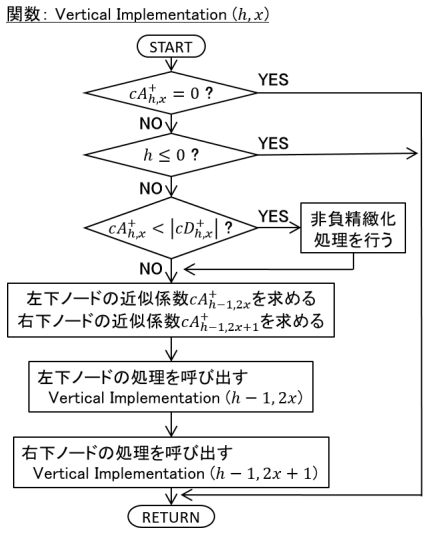

図 4 垂直型実装用関数の説明図
Fig. 4 An example of the recursive implementation.

し，そのモジュール内で自分自身を呼び出す方式である．

本実装では，1 つのノード（例：ノード (h,x)）における処理をモジュール化する．**図 4** に垂直型実装用関数の説明図を示す．この処理モジュールでの基本的な演算処理は，図の中央部分にあるところの，自ノードの近似係数 $cA^+_{h,x}$ と詳細係数 $cD^+_{h,x}$ を用いて一階層下の近似係数 $cA^+_{h-1,2x}$ および $cA^+_{h-1,2x+1}$ を求める処理である．概念的には，上記演算を行った結果，$cA^+_{h-1,2x}$ または $cA^+_{h-1,2x+1}$ のいずれかに負値が発生した場合には，3.2.3 項で説明した非負精緻化処理を行ったうえで，改めて式 (5) および式 (6) に従って，2 つの近似係数の値を求め直す処理を行うといった処理の流れとなる．しかし，$cA^+_{h,x}<|cD^+_{h,x}|$ を満たせば左下右下いずれかの演算結果が必ず負値となることから，実装にあたっては，$cA^+_{h-1,2x}$ および $cA^+_{h-1,2x+1}$ を求める演算に先んじてこの条件判定を行い，その結果に基づき，非負精緻化処理の要否を判定している．

続いてこれらの処理を終えたところで，左下のノード $(h-1,2x)$ および右下のノード $(h-1,2x+1)$ に対するこの処理モジュール自身の呼出しを行う．こうすることで，上位層の処理を終えた直後に，直下の 2 ノードの処理へと降りて行く．

本再帰呼出しでは 2 つの終了条件を設定する．1 つ目の終了条件は，枝刈り適用に対する判定条件である．2.3 節でも述べたとおり，枝刈りの適用条件は，近似係数 $cA^+_{h,x}$ の値が 0 であるか否かである．もう 1 つの終了条件は，最下層ノードの判定条件である．階層番号を参照して，最下層ノードを超えた場合 $(h\le 0)$ には処理を行わないように条件を設定する．なお，これらの終了条件の判定は，このモジュールにおける一連の処理の最初に設けておく．

従来手法である水平型実装 [10] の場合，演算省略ノード数メモリの操作によって生ずるオーバヘッドが避けられないことから，枝刈り処理の導入により，逆に演算効率を大きく低下させてしまうケースさえも発生するという問題があった．しかし上述したように，非負精緻化をともなう枝刈り型 Privelet 法における枝刈り処理を再帰呼出しによって実装することにより，演算効率を高める目的で水平型実装に導入されている演算省略ノード数メモリの管理がそもそも不要となる．すなわち，垂直型実装の狙いは，単に演算の手順を入れ替えることではなく，水平型実装における演算省略ノード数メモリの管理を排除したうえで高い演算効率化を図ることにある．このことは，たとえばランダム一次元化方式により変換されたデータに代表される，非ゼロ値が多数点在しているようなデータの場合も含めて，演算効率化効果の向上が期待できることを意味する．

5. 評価

5.1 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非負精緻化をともなう Privelet 法による秘匿処理を適用し，提案手法（垂直型実装）による逆 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（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 法を適用し，差分プライバシを規定するパラメータの値は $\epsilon=0.1$ とした．また，考察において条件をそろえた形で水平型実装と垂直型実装との実行時間の比較を行うため，あらかじめ摂動値データを作成しておき，各ノードには同じ摂動値を加えるようにした．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，2 次元上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 [23]，(d) ランダム方式という 4 方式によって 1 次元化を行ったうえで，その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，その値の大きい順（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

表 1 垂直型実装の処理時間
Table 1 Processing time of vertical implementation.

1 次元化方式	エリア	処理時間（100 ループの平均）		処理時間比率 [%]	時間短縮率 [%]
		(α) 枝刈あり [ms]	(β) 枝刈なし [ms]	α/β	$(\beta-\alpha)/\beta$
(a) ラスター方式	北海道	5.5	36.9	15.0	85.0
	四国	3.5	9.1	37.9	62.1
	関東	4.6	9.0	51.5	48.5
	北海道 1/4	2.3	9.2	24.4	75.6
(b) ソート方式	北海道	1.3	36.9	3.6	96.4
	四国	1.6	9.1	17.4	82.7
	関東	3.1	8.8	35.1	64.9
	北海道 1/4	0.6	9.1	6.8	93.2
(c) Morton 方式	北海道	3.7	37.0	10.1	89.9
	四国	2.6	9.0	28.2	71.8
	関東	4.2	8.8	47.2	52.8
	北海道 1/4	1.7	9.2	18.2	81.8
(d) ランダム方式	北海道	9.0	37.0	24.3	75.7
	四国	5.6	9.1	61.5	38.5
	関東	7.5	9.0	83.8	16.2
	北海道 1/4	3.6	9.2	38.8	61.2

のである．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基準を満たさないが，ほか 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．

5.2 評価結果

表 1 に，提案手法（垂直型実装）による処理時間の計測結果を示す．データ 1 次元化 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，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．

6. 考察

本章では，前章で行った評価から得られた結果をもとに，今回提案した垂直型実装手法についての考察を行う，提案手法は，5.2 節に示したように，高い演算効率化の効果が期待できるものの，対象とするデータによってその効果が変わって来る．そこで本章では，まず時間短縮率に影響を与える諸条件についての考察を行う．続いて，今回提案を行った垂直型実装手法が，従来手法であるところの水平型実装手法と比べてどの程度の演算効率の改善効果をもたらすことができたのかについて考察する．さらに，今回提案した演算効率化手法に関連する，主要ないくつかの演算効率化手法との差異についての考察も行う．

6.1 諸条件と時間短縮率の関係

表 1 から分かるように，今回提案を行った垂直型実装手法では，入力となる 1 次元スカラベクトルの値とその並びによって，演算効率化効果に差異が見られる．そしてこの入力ベクトルの並びは，2 次元空間に配置されたメッシュ人口データを 1 次元スカラベクトルへと写像する方法によって異なって来る．そこで本節では，はじめに 1 次元への写像方式と時間短縮率の関係について考察する．ここでの考察は，入力ベクトルの値におけるスパース性と密接な関係を持つ，疎密の現れ方に対する考察という側面をも有している．

一方，入力ベクトルの値については，そのスパース性に影響を与えるもう 1 つの要因であるところの，元データにおけるゼロ値含有比率の観点から，演算効率化効果に与える影響についての考察を行う．さらにスパース性とは異なる視点に立ち，枝刈りにより省略される演算の数と演算効率化の効果，および，入力ベクトルが持つ値との関係についての考察も試みる．

6.1.1 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，提案手法（垂直型実装）の場合，表 1 に示したとおり，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．

これらの写像方式によってもたらされるところの，写像後データの偏在性の高さについて考察してみると，ソート方式の場合にはデータが降順で並ぶことから偏在性が最も高く，次いで，「人口分布」という地理的偏在性の高いデータの局所性が保存される Morton 方式，そして，X 軸方向の局所性のみが保存されるラスター方式となり，データの偏在性を破壊するランダム方式が最も低いこととなる [9], [10]．したがって上記の結果は，本演算効率化手法による効率向上の効果が，データの偏在性が高いほど大きいことを示唆している．

なお，この性質は，**表 2** に示す水平型実装による処理時間の計測結果（文献 [9], [10] より引用）にも現れているとおり，水平型実装においても見られる性質である．表を見ると，データ 1 次元化 4 方式間での時間短縮率の傾向は垂

表 2 水平型実装の処理時間 [9], [10]
Table 2 Processing time of horizontal implementation [9], [10].

1 次元化方式	エリア	処理時間（100 ループの平均）		処理時間比率 [%]	時間短縮率 [%]
		(α) 枝刈あり [ms]	(β) 枝刈なし [ms]	α/β	$(\beta-\alpha)/\beta$
(a) ラスター方式	北海道	23.7	45.1	52.5	47.5
	四国	10.2	11.3	90.2	9.9
	関東	10.6	11.2	94.2	5.8
	北海道 1/4	8.4	10.8	77.7	22.3
(b) ソート方式	北海道	4.3	44.5	9.6	90.4
	四国	3.2	11.2	28.4	71.7
	関東	5.2	11.1	46.7	53.3
	北海道 1/4	1.8	10.7	16.9	83.1
(c) Morton 方式	北海道	15.2	45.4	33.6	66.4
	四国	7.6	11.2	67.9	32.1
	関東	8.7	11.1	78.0	22.0
	北海道 1/4	5.8	10.7	54.7	45.3
(d) ランダム方式	北海道	35.9	44.7	80.2	19.8
	四国	14.9	11.3	131.6	−31.6
	関東	14.9	11.6	128.8	−28.2
	北海道 1/4	12.1	11.1	109.0	−9.0

直型実装による処理時間と同様，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．このように，1 次元への写像方式と時間短縮率の関係については，垂直型実装においても水平型実装においても処理結果が同様の傾向を示すことが確認でき，この傾向が実装手法による効果ではなく，枝刈り処理自体がもたらす効果であることが示唆される．

6.1.2 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，6.1.1 項で述べたような疎密の現れ方という観点のほかに，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という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．

提案手法（垂直型実装）における，元データのゼロ値含有比率と時間短縮率との関係を**図 5** に示す．比較のため，水平型実装における元データのゼロ値含有比率と時間短縮率との関係もあわせて示している．グラフから，提案手法においても水平型実装と同様に，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．

これらの結果は，実装手法に関わらず，データの偏在性に加え，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．

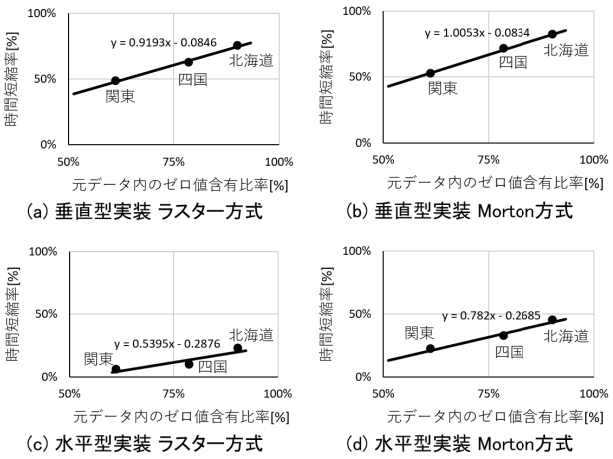

図 5 時間短縮率対元データにおけるゼロ値含有比率
Fig. 5 Time reduction rate versus zero value ratio in original data.

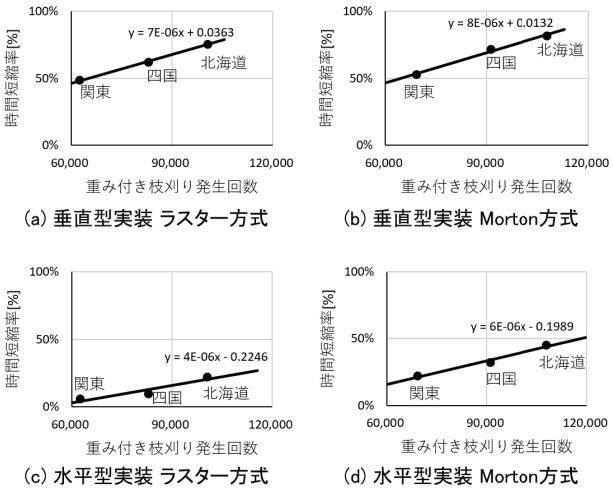

図 6 重み付き枝刈り発生回数と時間短縮率の相関
Fig. 6 Correlation between weighted pruning count and time reduction rate.

6.1.3 省略された演算の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相関について考察を行う．3 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．すなわち，1 つの最下層ノードで枝刈りが発生した場合，省略される演算数は 2（2 分木で接続された左下右下各 1 つずつの最終結果であるリーフ値を求める演算の数）であるが，その一階層上で枝刈りが発生した場合に演算が省略されるノード数はその 2 倍の数が加算された値となる．したがって，階層 h を起点とする枝刈りの発生回数を p_h とするとき，枝刈りにより省略される近似係数演算の総数（ここでは “重み付き枝刈り発生回数” と呼ぶ）の値 WP は，次式によって求めることができる．

$$WP = \sum_{h=1}^{H-1} \left(p_h \cdot \sum_{i=1}^{h} 2^i \right)$$

図 6 に，提案手法（垂直型実装）における重み付き枝刈り発生回数と時間短縮率の関係を示す．なお，比較のため，

表 3 垂直型実装と水平型実装の時間短縮率比較
Table 3 Time reduction rate of Vertical and horizontal implementation.

1 次元化方式	エリア	元データのゼロ値含有比率 [%]	垂直型実装の時間短縮率 [%]	水平型実装の時間短縮率 [%]
(a) ラスター方式	北海道	95.0	85.0	47.5
	四国	78.7	62.1	9.9
	関東	61.2	48.5	5.8
	北海道 1/4	90.3	75.6	22.3
(b) ソート方式	北海道	95.0	96.4	90.4
	四国	78.7	82.7	71.7
	関東	61.2	64.9	53.3
	北海道 1/4	90.3	93.2	83.1
(c) Morton 方式	北海道	95.0	89.9	66.4
	四国	78.7	71.8	32.1
	関東	61.2	52.8	22.0
	北海道 1/4	90.3	81.8	45.3
(d) ランダム方式	北海道	95.0	75.7	19.8
	四国	78.7	38.5	−31.6
	関東	61.2	16.2	−28.8
	北海道 1/4	90.3	61.2	−9.0

水平型実装における重み付き枝刈り発生回数と時間短縮率の関係もあわせて示している．グラフから，提案手法においても水平型実装と同様に，重み付き枝刈り発生回数と時間短縮率の間に高い正の相関関係のあることが分かる．これらの結果は，より高い階層を起点とする枝刈りが発生するほど，演算が省略されるノード数の値が大きくなること，すなわち，演算が省略されるノード数が増えることを示唆している．

一般に，入力データの広い範囲でゼロ値が連続している場合，上位階層まで近似係数 $cA_{h,x}$ および詳細係数 $cD_{h,x}$ の値がともにセロ値となり，$cD_{h,x}$ へのノイズ付加によって負値が発生し，その結果として非負精緻化が起こり，上位階層において枝刈りが発生する可能性が高くなる [9], [10]．したがって，どちらの実装方式においても，入力データの広い範囲にゼロ値が分布するようなデータに対して本手法を適用した場合に，枝刈りによる大きな時間短縮率が期待できることが分かる．

6.2 提案手法と水平型実装の演算効率化効果の比較

ここでは，提案手法（垂直型実装）と水平型実装の演算効率化の効果を比較する．**表 3** に，元データのゼロ値含有比率と，垂直型・水平型実装の時間短縮率の値を示す．

表 3 から，すべての条件において，提案手法の方が，水平型実装よりも時間短縮率が高いことが分かる．特に，ランダム方式において顕著な差異が見られる．これは，ランダム方式の場合にはまとまった省略が発生しにくいことから，水平型実装に必要となる，省略するノード数の算出によるオーバヘッドを上回る効率化が難しいことに起因しているものと考えられる．

本研究で行った評価実験の結果を示した表 1 と表 2 との間で「(β) 枝刈りなし」の値を見比べると，おおむね 20%ほどの差異が見られる．これは，水平型実装と垂直型実装における処理プロセスの違いによってもたらされたものであるが，「(α) 枝刈りあり」の演算時間の値にも，この差異が含まれてしまっている．このため，両実装方式の演算時間自体を直接比較することは，枝刈り導入効果に対する正当な評価とはいえない．しかし，時間短縮率という，各々の実装方式内に閉じて求めた演算効率化効果の違いを比較することはできる．すなわち，本稿で提案した垂直型実装の方が，水平型実装よりも演算効率化効果が高いことを具体的に示した点は本研究の貢献であると考える．

6.3 関連する演算効率化手法との差異

枝刈りによる演算の効率化は，データのスパース性によってもたらされる演算過程のスパース性に基づいて達成されている．演算過程のスパース性に基づく差分プライバシにおける演算効率化手法としては，Cormode らの手法 [24] が先行研究としてあげられる．この手法は，Laplace メカニズムの適用の結果，ある閾値を超える値を持つセルの値だけが出力されればよい（出力はスパースになる）という前提で，出力されない値（閾値以下となる値）の演算処理を抑制することにより，演算過程のスパース性を達成している．ここで，閾値を 0 と設定すれば，出力の非負制約を満たすことに相当するが，Laplace ノイズの対称性より，非負制約を満たしているデータに対して Laplace メカニズムを適用した出力において，非負となるセル数の期待値は入力データ数の半分を上回ることになる．すなわち，この手法で非負制約を達成しても，原理的に演算処理の削減率が 50%を超えることはない．また，演算処理を抑制しているという点を除けば Laplace メカニズムの適用後に閾値以下のデータ（負の値）を単に削除するという処理と同等の出力となることから，出力データに含まれる値の総和は入力データに含まれる値の総和から「上振れ」してしまうとともに，部分和精度の改善も達成されないなど，出力データの精度に関して実用上無視できない課題を持つ．

一方，匿名化個票開示を対象とした我々の提案手法 [32] では，等価な個票データを持つ集計データを得ることを目的として，Laplace メカニズムの適用の後に，その出力値を多次元ベクトル空間上の点と見なし，非負制約と総数制約（入力データと出力データで，全セルの値の総和が等しいという制約），および整数制約（出力データに含まれる各セルの値は必ず整数となるという制約）の 3 種類の制約を満たす多次元ベクトル空間における超平面上の格子点を最近傍探索することにより，スパースな入力データに対する出力のスパース性を達成している．そこでは，最近傍探索にお

いて演算上の工夫をすることにより，探索に要する演算量を大幅に削減している．この手法では，出力のスパース性を確保する過程で総数制約を保持することから，Cormode らの手法 [24] のような出力データの値の「上振れ」の問題はない．しかし，この手法では（目的の相違から）部分和精度の改善については特に扱われておらず，さらに Laplace ノイズの発生自体を抑制するわけではないことから，その演算量は必ず入力の次元数を上回ることになる．

また，出力データの部分和精度を向上させる先行研究としては，Barak らの手法 [21] があげられる．Privelet 法が部分和精度を向上させるために HWT を用いるのに対し，この手法では Fourier 変換を用いている．しかし，HWT においては，その変換および逆変換において部分和（をスケールさせた値）が陽に演算過程に現れるのに対し，Fourier 変換はそのような性質を持たない．したがって，提案手法で用いたような，部分和の非負制約に基づく「枝刈り」を Barak らの手法に対して適用することは困難であり，彼らの論文でも非負制約を満たすために（「演算コストが高い」という注釈つきで）出力データに対して線形計画法を適用することを提案している．

なお，原理的には Haar Wavelet 以外の Wavelet 関数（Debussy Wavelet など）による離散 Wavelet 変換に基づいて，差分プライバシを満たす出力を得る方式を構成することも可能と考えられる．ただし，Haar Wavelet 以外の Wavelet 変換において，その演算過程で部分和が演算過程に現れるとは限らない．その場合，部分和の非負制約に相当する，Wavelet 係数間の制約条件を発見することができれば，本方式と同様の構成による効率化が可能と考えられはするものの，そのような制約条件の有無や条件式は適用する Wavelet 関数に依存する．

以上のような状況のもと，提案手法による演算の効率化は，非負制約の導入をふまえ，入力データのスパース性を失うことなく出力データの生成を行う処理をその演算過程のスパース性を保持することにより達成している．しかも，単一のセル値の非負制約だけでなく，複数のセル値の和である部分和もまた非負制約を満たすことに着目し，部分和に基づく「枝刈り」を適用することによって高い演算効率化効果の実現に成功している．

7. おわりに

本報告では，非負精緻化をともなう Privelet 法に対する新たな演算効率化手法として，垂直型枝刈り実装法の提案を行った．1 次元への写像方式については，提案手法においても水平型実装と同様に，比較した 4 方式のうち差分プライバシ基準を満たさないソート方式を除くと Morton 方式が最も大きな時間短縮率が得られることが示された．続いて，水平型実装と同様，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，高い相関がみられる傾向も確認された．さらに，省略された演算の数と時間短縮率の関係についても，水平型と同様に，高い相関がみられることが明らかとなった．そして，提案手法が水平型実装よりも大きな時間短縮率を実現することが示された．最後に，関連する演算効率化手法を紹介し，それらの手法との間の差異についても考察を加えた．今後は，たとえば入力データの次元変換を必要としない手法なども含めた，さらなる処理効率化の可能性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 2019-09-24）．
[2] Misuraca, G., Mureddu, F. and Osimo, D.: Policy-Making 2.0: Unleashing the Power of Big Data for Public Governance, *Open Government* (*Ed.*) *Public Administration and Information Technology*, Vol.4, pp.171–188, Springer (2014).
[3] 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，国土交通省（オンライン），入手先 ⟨http://www.mlit.go.jp/report/press/road01_hh_001144.html⟩（参照 2019-09-24）．
[4] 佐治秀剛，田中良寛，鹿野島秀行，牧野浩志：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).
[5] 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).
[6] NTT ドコモ：モバイル空間統計に関する情報，NTT ドコモ（オンライン），入手先 ⟨https://www.nttdocomo.co.jp/corporate/disclosure/mobile_spatial_statistics/⟩（参照 2019-09-24）．
[7] 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).
[8] 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).
[9] 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).
[10] 本郷節之，寺田雅之，鈴木昭弘，稲垣 潤：非負精緻化をともなう Privelet 法の演算効率化手法，情報処理学会論文誌，Vol.61, No.2, pp.474–485 (2020).
[11] 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，*2019 Symposium on Cryptography and Information Security*, pp.1–8 (2019).
[12] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Longhurst, J., Nordholt, E., Seri, G. and Wolf, P.-P.: *Handbook on Statistical Disclosure Control*, Statistics Netherlands (2010).
[13] Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E., Spicer, K. and Wolf, P.-P.:

Statistical Disclosure Control, John Wiley & Sons (2012).
[14] 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).
[15] 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).
[16] Fung, B., Wang, K., Chen, R. and Yu, P.: Privacy-preserving Data Publishing, *ACM Computing Surveys*, Vol.42, No.4, pp.1–53 (2010).
[17] 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).
[18] 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).
[19] 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).
[20] 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).
[21] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K. and Berkeley, U.: 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, ACM Press (2007).
[22] Li, C., Hay, M., Rastogi, V., Miklau, G. and McGregor, A.: Optimizing linear counting queries under differential privacy, *Proc. 29th ACM SIGMOD-SIGACT-SIGART Symp., Principles of Database Systems* (*PODS '10*), pp.123–134, ACM Press (2010).
[23] 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).
[24] Cormode, G., Procopiuc, M., Srivastava, D. and Tran, T.: Differentian Private Publication of Sparse Data, *Proc. Intl. Database Theory* (*ICDT 2012*) (2012).
[25] Xiao, X., Wang, G. and Gehrke, J.: Differential Privacy via Wavelet Transforms, *Proc. 26th Intl. Conf. Data Engineering* (*ICDE 2010*), pp.225–236 (2010).
[26] 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).
[27] Chen, Y. and Ye, X.: Projection Onto A Simplex, arXiv:1101.6081 (2011).
[28] Wang, W. and Carreira-Perpiñán, M.Á.: Projection onto the probability simplex: A efficient algorithm with a simple proof, and an application, arXiv:1309.1541 (2013).
[29] 寺田雅之，山口高康，本郷節之：高次元大規模データへの差分プライバシー適用のための最適精緻化法，暗号と情報セキュリティシンポジウム（SCIS），No.3B3-5, pp.1–8 (2017).
[30] Xu, J., Zhang, Z., Xiao, X., Yang, Y. and Winslett, M.: Differentially Private Histogram Publication, *The International Journal on Verry Large Data Bases* (*VLDB*), Vol.22, No.6, pp.1–10 (2013).
[31] Acs, G., Castelluccia, C. and Chen, R.: Differentially Private Histogram Publishing through Lossy Compression, *IEEE International Conference on Data Mining* (*ICDM*), pp.1–10 (2012).
[32] 寺田雅之，山口高康，本郷節之：匿名化個票開示への差分プライバシーの適用，情報処理学会論文誌，Vol.58, No.9, pp.1483–1500 (2017).

推薦文

Wavelet 変換，差分プライバシ加工，逆 Wavelet 変換のデータ加工工程に対し，演算効率化の工夫を試みており，国勢調査の地域メッシュ人口データを用いた実証的評価において，提案手法の計算時間の効率化が有効である範囲を，データの前処理方法，データのスパース性を変数として包括的に実証評価した点が評価できる．また，分割表のような非負値条件，集計値との整合性といった拘束条件を満たす差分プライバシの実施手法に関する既存研究と本研究の手法の関係が明確に説明されており，その点でも今後この分野の研究を行う研究者に有益な情報が提供されている．以上のことから，推薦論文に値するため，推薦いたします．

（コンピュータセキュリティ研究会主査　山内 利宏）

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ー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）ほか，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 年より現職．博士（工学）．情報セキュリティ技術，プライバシー保護技術，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各会員．

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．