2022 年度 修士論文

完全準同型暗号上での差分プライベート パーティショニングアルゴリズムの高速化

提出日： 2023 年 01 月 23 日
指導： 山名 早人 教授
研究指導名：並列・分散アーキテクチャ研究

早稲田大学大学院 基幹理工学研究科 情報理工・情報通信専攻
学籍番号：5121F018-7

牛山 翔二郎

概　要

本研究では，複数のデータ所有者から収集した生データに対して，クラウド上でレンジクエリ処理を行う際の，差分プライバシと完全準同型暗号を用いたプライバシ保護に取り組む．具体的には，データ所有者，クラウドサーバ，クエリ応答システムを使用するデータ解析者の 3 パーティからなるモデルを想定し，生データのプライバシをクラウドサーバ，データ解析者から保護する．プライバシ保護を実現する手法として，2020 年の Chowdhury らによる提案と同様に，生データから差分プライバシ適用済インデックスの生成までを準同型暗号上で行うモデルを採用する．レンジクエリを対象とした場合，個々のカウント値に差分プライバシを適用する代わりに，ヒストグラムをパーティショニングした後で差分プライバシを適用することにより，差分プライバシ適用後の誤差を小さくすることができることが知られている．しかし，同処理を準同型暗号上で行う場合，データ数に対し指数的に計算量が増加し，実用的ではない．この問題に対し，本研究では，計算量を線形増加に抑えたプライバシ保護パーティショニングアルゴリズムを提案する．提案手法は，隣接するデータ間の差分と閾値との比較のみによりデータ分割点を決定する．Nettrace をはじめとする 7 種類のデータセットを用いた評価実験の結果，従来のデータ依存型パーティショニングアルゴリズムと同等の精度を保つことができること，および，データ数に比例した実行時間で処理できることを確認した．

目次

第1章　　はじめに

1.1 研究背景とモチベーション

近年，クエリ応答システムとして，クラウドコンピューティングの活用が注目される．データ解析において，解析対象のデータ（生データ）が機密性の高い情報（例：金融データ）を含む場合，データのプライバシ漏洩は深刻な問題となる．具体的には，データを保存するクラウドは容易に生データを知ることができる．また，クエリ応答を受け取るエンティティは複数のクエリ応答から生データを推定することができる．本研究では，クラウドコンピューティングを用いたクエリ処理を対象に，複数のデータ所有者から提供される生データの保護に取り組む．また，クエリとして，データベースの特性を理解するために頻繁に利用されるレンジクエリ処理を対象とする．

（1）データ所有者，（2）クラウドサーバ，（3）データ解析者の 3 つのエンティティから構成されるレンジクエリ応答システムを想定する．ここで，データ所有者は解析対象の生データを提供し，クラウドサーバは生データを保存及び操作する．また，データ解析者はレンジクエリを送信し，クラウドサーバからクエリ応答を受信する．上述のようなシステムにおいて，生データはクラウドサーバとデータ解析者のいずれか，または両者に対して漏洩する危険がある．本研究のセキュリティ目標は，クラウドサーバとデータ解析者の両者に対して生データを保護することである．

クラウドサーバとデータ解析者の両者に対するプライバシ保護手法として，(1) 準同型暗号と差分プライバシの組み合わせ[1, 2, 3, 4, 5, 6, 7, 8]，（2）秘匿マルチパーティ計算と差分プライバシの組み合わせ[9, 10, 11]，（3）ローカル差分プライバシ[12, 13]の 3 つの手法が提案されている．

準同型暗号と差分プライバシの組み合わせ[1, 2, 3, 4, 5, 6, 7, 8]は，準同型暗号上で差分プライバシ[14, 15]を適用することで，クラウドサーバとデータ解析者の両者に対して生データのプライバシを保護する．準同型暗号は復号を行わずに暗号文上で加算や乗算を行うことを可能にする暗号方式である．また，差分プライバシはデータの真値を秘匿するために，データに適切な量のノイズを加算するプライバシ保護手法である．上記の組み合わせはクラウドサーバとデータ解析者から生データを保護する有望な手法であるが，準同型演算による長い実行時間が必要となる．また，取り扱うデータが暗号化されており，準同型演算では条件分岐を実装することがでないため，柔軟なプログラム実装が困難となる．

秘匿マルチパーティ計算と差分プライバシの組み合わせ[9, 10, 11]は，秘匿マルチパーティ計算下で差分プライバシを適用することで，クラウドサーバとデータ解析者に対するプライバシ保護を達成するクエリ処理を実現する．ここで，秘匿マルチパーティ計算とは，非

共謀である複数のパーティを使用することで，信頼できる第三者を必要としない秘匿計算プロトコルの総称である．しかし，上記の組み合わせを使用する場合は，秘匿マルチパーティ計算が対話的なプロトコルに基づいており，計算中は複数のパーティがオンラインでなければならず，また，パーティ間での通信が必要となるため，通信コストが大きくなる．

ローカル差分プライバシ[12, 13]は，クラウドサーバが生データを収集する前にデータ所有者が生データにノイズを加算することで，クラウドサーバとデータ解析者に対して生データを保護する差分プライバシ手法の一つである．しかし，ローカル差分プライバシでは，収集前の個々のデータに対してノイズを加算するため，データ収集後にクラウドサーバがクエリ応答にノイズを加算するセントラル差分プライバシと比較してクエリ応答の精度が劣化する．

本研究では，クラウドサーバとデータ解析者の両者に対して生データを保護する中央集権型のレンジクエリ応答処理システムを実現するために，準同型暗号と差分プライバシの組み合わせ手法を採用する．具体的には，2020 年に Chowdhury らが提案した DP-index[7]と同様に，暗号文上で収集した生データから差分プライバシ適用済データの生成までを準同型暗号上で行う．本研究の主要なチャレンジは，差分プライバシに起因する誤差を低減し，クエリ応答の高い精度を保ちながら，準同型暗号上での差分プライバシ適用処理を高速化することである．

レンジクエリを対象とした高精度な差分プライバシ手法として，データ依存パーティショニング手法[16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]とワークロード依存最適化手法[25, 26, 27, 28]がある．データ依存パーティショニング手法は，ヒストグラム内の連続する値が近いデータを統合することで，差分プライバシ下での精度を向上させる．データ依存パーティショニング手法は，入力クエリとは独立した処理であり，入力ヒストグラムのみを用いて差分プライベートなヒストグラムを構築することができる．一方，ワークロード依存最適化手法は，想定される入力クエリの傾向が既知であるという前提のもと，与えられたワークロードに最適化したノイズを加算することで，差分プライバシ下での精度を向上させる処理である．

本研究では，入力クエリの傾向を想定せず，あらゆるレンジクエリに対応するため，データ依存パーティショニング手法を採用する．そして，差分プライバシ適用済みデータ（差分プライベートヒストグラムと呼ぶ）を事前に生成する．データ依存パーティショニング手法では，事前に構築した差分プライベートヒストグラムを用いて全てのクエリに応答するため，クエリごとにプライバシ予算を消費しない．つまり，クエリ応答回数によらずデータ解析者は生データを推定することができない．これは，「対話的に差分プライバシを適用し，クエリ応答ごとにプライバシ予算を消費するクエリ応答システム[29, 30, 31]では，クエリ応答回数が増加すると生データを統計的に推定されてしまう」という問題を解決する．

データ依存パーティショニングは，差分プライバシを保証しながら小さい応答誤差を達成することができるが，準同型暗号上で実装する場合は，計算コストが大きくなる．この計算コスト問題を解決するため，本研究では準同型暗号に適したプライバシ保護パーティショニングアルゴリズムを提案する．提案モデルの概要を図 1.1 に示す．

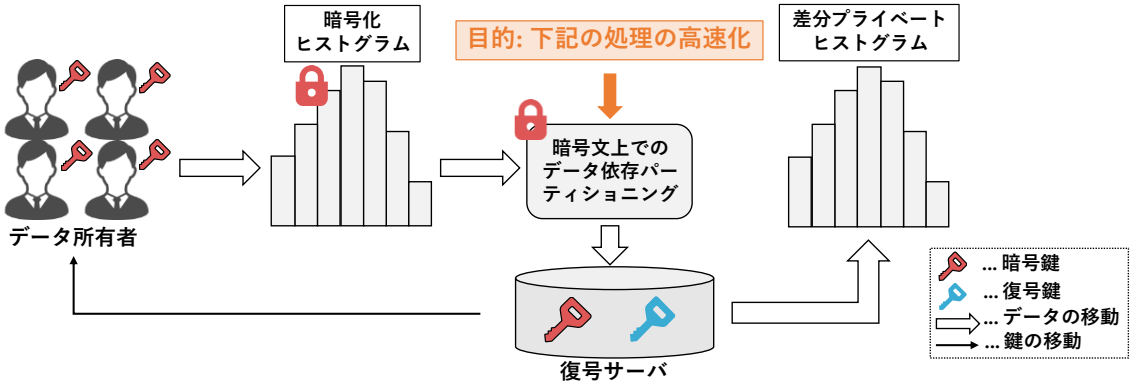


図 1.1　提案モデルの概要

1.2 提案アルゴリズムのアプローチ

提案アルゴリズムは，一次元ヒストグラムを対象に，暗号文上でのデータ依存パーティショニングの計算コストを削減し，高速化することを目的としている．準同型暗号を使用した演算の実行においては，平文上での実行時間と比較して約 10^9 倍の実行時間を要する[32]ため，データ依存パーティショニングの計算量を低減する必要がある．

上記の目的を達成するために，データ依存パーティショニングの処理を簡略化することを考える．データ依存パーティショニングは，データを統合する際とノイズを加算する際に発生する二種類の誤差の合計で推定されるエラーコストが最小になるような最適パーティションを探索する．パーティショニングの際に，連続するデータ間の小さな差分による誤差は，推定されるエラーコストに大きな影響を与えない．すなわち，連続するデータ間の大きな差分にのみ注目し，パーティショニングを簡略化することで，最適に近いパーティショニングを実現できる可能性がある．この考えに基づき，提案アルゴリズムでは，隣接するデータ間の差分が小さい場合に，それらのデータを統合する．すなわち，ヒストグラム内の隣接するデータ間の差分によって，パーティションの各境界を決定する．既存のデータ依存パーティショニング[25]を暗号文上で厳密に再現した実装は，計算量が$O(n^2)$となる[8]のに対して，提案手法の計算量は$O(n)$を達成する（nは入力ヒストグラム内の要素数を指す）．

準同型暗号に関しては，ビットワイズな完全準同型暗号[33]スキームである TFHE (torus fully homomorphic encryption) [34]を採用する．TFHE は，暗号文上での任意の論理ゲー

ト（例：AND, OR）の演算を可能にするため，パーティショニングを実装する際に不可欠となる絶対値や二値比較を取得する関数の実装を可能にする．

1.3 貢献

本研究の貢献は以下の通りである．

- 暗号文上でのデータ依存パーティショニングを高速化することを目的に，準同型暗号に適したプライバシ保護パーティショニングアルゴリズムを提案する．提案アルゴリズムは，既存研究[8]の計算量$O(n^2)$から，計算量を$O(n)$に低減する（nは入力ヒストグラム内の要素数を指す）．評価実験では，$n = 4{,}096$において，提案手法の実行時間が実用的であることを示した．

- 7 つの実データセットを使用した評価実験から，提案手法の精度が，Li らが提案した最先端のデータ依存パーティショニングアルゴリズム[25]と比較して同等であることを確認した．また，提案手法は 3 つの代表的な競合手法[17, 18, 19]と比較しても，優位な精度結果を示した．

なお，本論文は著者らの[35]に対して，評価実験で用いる競合手法及びワークロードを増やすことにより，提案手法の精度及び実行時間に関する貢献を詳細化したものである．

本論文は以下の構成をとる．第 2 章では本論文の主要な背景知識である差分プライバシと完全準同型暗号について説明する．第 3 章では本研究の関連研究を紹介する．第 4 章では提案手法の詳細を説明する．第 5 章では提案手法の性能を実験により評価する．最後に第 6 章で結論を示す．

第2章　　背景知識

本章では，本研究の主要な背景知識である差分プライバシと完全準同型暗号について説明する．さらに，本研究で対象としているレンジクエリについて説明する．

2.1 差分プライバシ

差分プライバシ[14, 15]は，ランダムノイズを加算することで生データの真値を秘匿する匿名化手法であり，2006 年に Dwork らによって提案された．差分プライバシは，数学的根拠に基づいた厳格なプライバシ保護手法であり，情報理論的安全性を提供する．また，任意の背景知識を持つ攻撃者に対して有効であるとされている．

一般的に，差分プライバシにおけるプライバシ保護強度と有用性（すなわち，差分プライバシ適用後のデータの精度）はトレードオフの関係にある．具体的には，大きなノイズは強いプライバシ保護を提供する一方で，データの有用性を劣化させる．

差分プライバシの定義を，以下の**定義 1** に示す．

定義 1：ϵ-差分プライバシ[14, 15]

DとD'を隣接データベース，Sを出力の集合，ϵをプライバシパラメータとした時，以下の不等式を満たす場合に，ランダムメカニズムmはϵ-差分プライバシであると言う．

$$\frac{\Pr\left(m(D) \in S\right)}{\Pr\left(m(D') \in S\right)} \leq \exp(\epsilon) \qquad (2.1)$$

ここで，隣接データベースとは，1 つのレコードを除いて，残りの全レコードが同一ある 2 つのデータベースである．また，プライバシパラメータは，0 より大きい実数であり，プライバシ保護強度と有用性のトレードオフを調整する．

ランダムメカニズムとは，差分プライバシを満たすために，入力値にランダムノイズを加算する関数である．最も代表的なランダムメカニズムであるラプラスメカニズム[14]を，**定義 2** に示す．

定義 2：ラプラスメカニズム[14]

ラプラスメカニズムm_{Lap}は，平均 0 のラプラス分布からサンプリングされるノイズzを，クエリqの出力に加算する．

$$m_{Lap}(D) = q(D) + z \qquad (2.2)$$

ラプラスノイズzの平均が 0 であることから，ラプラスメカニズム適用後のデータを多数集めた攻撃者は，生データを統計的に復元することができる．差分プライバシを満たす全てのメカニズムは，上記のような性質を有する．このような攻撃を防ぐため，差分プライバシを適用することができる上限回数を，プライバシ予算を設けることで制限する必要がある．

一方で，ラプラスノイズzのスケールは，プライバシパラメータϵとクエリの敏感度Δ_qの 2 つのパラメータによって決定される．スケールはΔ_q/ϵで与えられるため，zの大きさにΔ_qが大きく影響する．敏感度Δ_qは，隣接データベースに対して，単一レコードがクエリqの出力に与える影響の最大値で与えられる．敏感度の定義を，**定義 3** に示す．

定義 3：敏感度[14]

DとD'を隣接データベースとした時，クエリqの敏感度Δ_qは以下のように定義される．

$$\Delta_q = \max\left|\left|q(D) - q(D')\right|\right|_1 \qquad (2.3)$$

ここで，$||\cdot||_1$は L1 ノルムを評価する．

複数の差分プライバシアルゴリズムを合成した時，その合成アルゴリズムもまた差分プライバシを満たすという性質（直列合成定理[36]）がある．これを**定理 1** に示す．

定理 1：直列合成定理[36]

それぞれϵ_i-差分プライバシを満たすn個のアルゴリズムA_i $(i \in [1,n])$が与えられた時，$(A_1, A_2, \ldots, A_n)$の直列合成は$(\sum_{i=1}^{n} \epsilon_i)$-差分プライバシを満たす．

2.2 完全準同型暗号

完全準同型暗号[33]は，暗号文上での任意回数の加算と乗算を復号することなく行う暗号方式であり，2009 年に Gentry らにより提案された．一方，「完全」の付かない準同型暗号は，暗号文上での加算または乗算，もしくは加算と限られた回数（または数回）の乗算を可能にする．完全準同型暗号では，暗号文上で演算を行うことができるが，復号しない限り演算結果を知ることができない．すなわち，条件分岐では条件式の演算はできても，その結果である真理値を知ることができないため，真理値に応じて制御フローを変更することができない．また，平方根や三角関数などの複雑な関数は，演算に条件分岐が必要なため，完全準同型暗号上に制御フローの変更を伴った方法で実装することができない．上記の関数を実装する方法として，テイラー展開を用いた多項式近似がある．しかし，近似計算を行うために乗算が複数回必要となり実行時間が長くなると共に，近似誤差が生じる．

一方で，ビットワイズな完全準同型暗号を用いることで，論理回路で構成される任意の関数を暗号文上で実装することができる．そこで，本研究では暗号文上での絶対値及び二値比較結果の取得を目的に，ビットワイズな完全準同型暗号の一つである TFHE[34]を採用する．

2.3 レンジクエリ

本研究では，データベースに対する基本的な操作であるレンジクエリを対象としている．レンジクエリは，データサイエンスや IoT（Internet-of-Things）アプリケーションなどでデータベースの特性を把握するために頻繁に用いられる．

d次元のデータベースMに存在する列ベクトルを$\boldsymbol{x_i} = (x_{i,1}, x_{i,2}, \dots, x_{i,|\boldsymbol{x_i}|})$とする$(1 \leq i \leq d)$．すなわち，データベース$M$は列ベクトル$\boldsymbol{x_i}$ $(1 \leq i \leq d)$ の集合であり，列ベクトル$\boldsymbol{x_i}$は点$x_{i,j}$ $(1 \leq j \leq |\boldsymbol{x_i}|)$ の集合である．d次元における閉領域Sを，i番目の次元の閉範囲$[x_{i,min}, x_{i,max}]$とする（minは閉領域S内の列ベクトル$\boldsymbol{x_i}$のインデックスの最小値を，maxは最大値を表す）．レンジクエリは，閉領域Sで囲まれたレコードの総和を応答するクエリであり，以下の式(2.4)ように計算される．

$$RangeQuery\_d(x_{i,min}, x_{i,max}) = Sum(x_{i,max}) - Sum(x_{i,min-1}) \quad (2.4)$$[1]

ここで，$Sum(x_{i,k})$は，$x_{i,1}$から$x_{i,k}$までの合計を求める関数である．

また，本研究の提案手法では，対象データを一次元としているため，想定されるレンジクエリも一次元を対象としている．具体的に，一次元データ$\boldsymbol{x} = (x_1, x_2, \dots, x_{|x|})$が与えられた時，レンジクエリは閉範囲$[x_{min}, x_{max}]$$(1 \leq min \leq max \leq |\boldsymbol{x}|)$で囲まれた範囲の合計を応答し，以下の式(2.5)を計算する．

$$RangeQuery\_1(x_{min}, x_{max}) = Sum(x_{max}) - Sum(x_{min-1}) \quad (2.5)$$

一次元データに対するレンジクエリは実社会運用においても有用である．具体的な事例としては，時系列に気温を収集する IoT センサデータに対するデータ分析への適用が挙げられる．この場合，時系列データのインデックスは日時または時刻に対応していることが想定される．1 時間ごとの気温データといった細かく並べられた一次元時系列データに対して，1 週間や 1 か月ごとの統計的な傾向を分析したい場合において，レンジクエリは効率的な分析手段の 1 つとなる．

1 本計算方法に限らず，レンジクエリの計算効率化では様々な方法が提案されている．

第3章　関連研究

本章では，(1) データ依存パーティショニング手法と (2) 差分プライバシと準同型暗号を組み合わせた手法を紹介する．データ依存パーティショニング手法は，暗号文上でパーティショニングを高速化するための重要なアイディアの基盤となる．差分プライバシと準同型暗号を組み合わせる手法は，提案手法のセキュリティ前提の基盤を提供する．

3.1 データ依存パーティショニング手法

差分プライバシ[14, 15]は，レンジクエリ応答システムにおいて，データ解析者に対して生データのプライバシを保護する有望な手法である．データ依存パーティショニング手法は，差分プライバシ下においてレンジクエリ応答の精度を向上させることで知られる．データ依存パーティショニング手法の関連研究を表 3.1 にまとめる．

データ依存パーティショニング手法は，データにノイズを加算する前に，入力ヒストグラムをいくつかの連続する値が近いデータのグループごとに分割する．そして，各グループの合計にノイズを加算し，グループ内の全てのデータの値をノイズが加算された合計の平均で与える．したがって，(1) 平均化に起因する集計誤差と (2) ノイズに起因する摂動誤差の 2 種類の誤差が発生する．データ依存パーティショニング手法は，上記の 2 種類の誤差の合計が最小になるような最適パーティションを探索する．

既存のデータ依存パーティショニング手法[16, 17, 18, 19, 20, 21, 22, 23, 24, 26]は，アルゴリズム内で条件分岐または平方根を使用している．しかし，このような演算は完全準同型暗号（または準同型暗号）上では，2.2 節で説明したように，実装が困難となる．したがって，著者の知る限り，準同型暗号上の実装に適したデータ依存パーティショニング手法は，これまでに存在していない．

一方，従来提案されているデータ依存パーティショニング手法においては，2014 年に Li らが提案した DAWA[25]が最も高い精度を達成する．これは，既存の差分プライバシアルゴリズムを包括的に評価した DPBENCH[37]の実験結果でも示されている．DAWA はヒストグラムに対するレンジクエリを対象にしたアルゴリズムであり，(1) データ依存パーティショニング部と (2) ワークロード依存最適化部から構成される．近年，[28]のように，DAWA の精度を凌駕するワークロード依存最適化手法が提案されているが，データ依存パーティショニングの手法においては，DAWA が最高精度である[37]．

表 3.1　データ依存パーティショニング手法の関連研究

著者	会議・論文誌	出版年	対象データ	次元数
Acs, Gergely ら [17]	ICDM	2012	ヒストグラム	1 次元
Xu, Jia ら[18]	ICDE	2012	ヒストグラム	1 次元
Qardaji, Wahbeh ら[16]	ICDE	2013	地理空間データ	2 次元
Zhang, Xiaojian ら[19]	SDM	2014	ヒストグラム	1 次元
Xiao, Yonghui ら[20]	Transactions on Data Privacy	2014	ヒストグラム	多次元
Li, Chao ら[25]	VLDB	2014	ヒストグラム	1 及び 2 次元
Zhang, Jun ら [21]	SIGMOD	2016	空間データ	多次元
Kotsogiannis, Ios ら[22]	SIGMOD	2017	ヒストグラム	1 及び 2 次元
Kotsogiannis, Ios ら[26]	VLDB	2018	関係データベース	2 次元
Li, Hui ら[24]	Distributed and Parallel Databases	2019	ヒストグラム	多次元
Kato, Fumiyuki ら [23]	VLDB	2022	関係データベース	多次元

3.2 差分プライバシと準同型暗号を組み合わせた手法

近年，クラウドサーバとデータ解析者の両者から生データを保護することを目的に，差分プライバシと準同型暗号を組み合わせる手法[1, 2, 3, 4, 5, 6, 7, 8]が提案されている．クエリ応答システムを対象とした差分プライバシと準同型暗号を組み合わせる手法の関連研究を表 3.2 にまとめる．

先行研究[1, 2, 3, 4, 5, 6]では，準同型暗号を用いて生データを集約し，その後にクラウド上で暗号化データに差分プライバシ適用するシステムが提案されている．しかし，上記のシステムはデータ解析者が一人であることを想定している．一方で，複数のデータ解析者を対象とした場合，データ解析者ごとに異なる暗号鍵を用いてデータの集約及び差分プライバシ適用を行う必要があり，データ解析者の人数分の生データをクラウド上に保存することが要求されるため，非現実的である．また，データ依存パーティショニングも採用されていない．

2020 年に Chowdhury らによって Cryptϵ[7]論文内で提案された DP-index[7]は，線形準同型暗号の拡張版であるラベル準同型暗号[38]を採用し，暗号文上で差分プライバシを適用している．一方で，2021 年に著者らによって提案された DP-summary[8]は，TFHE 上で差分プライバシを適用する．DP-index と DP-summary は共に，プライバシ保護レンジクエリ処理を対象としており，(1) 演算を行うサーバと (2) 鍵管理および復号を行うサーバの 2 つのクラウドサーバで構成される．このような 2 パーティモデルは，準同型暗号を対象とするシステムでしばしば用いられる[39, 40, 41, 42]．DP-index と DP-summary は，(1) データ解析者からのクエリに対する応答速度の高速化と (2) プライバシ予算から解放されたクエリ応答回数の無制限化という 2 つの共通した利点を持つ．しかし，DP-index は入力データの分布に依存しない等間隔なパーティショニングを採用しているため，クエリ応答の精度が劣化するという欠点がある．一方で，DP-summary は，クエリ応答の精度向上を目的に，TFHE を用いて，暗号文上でデータ依存パーティショニングを適用している．しかし，考えられる全ての組み合わせの中から最適なパーティションを暗号文上で探索するため，計算量が入力データの大きさに対して指数増加する．このため，DP-summary は実行時間の観点から実用的ではない．

表 3.2 差分プライバシと準同型暗号を組み合わせた手法の関連研究

著者	会議・論文誌	出版年	モデル	データ解析者の数
Komawar, Saket ら[1]	ICCUBEA	2018	中央集権型	1 人
Raisaro, Jean Louis ら[2]	IEEE/ACM Transactions on Computational Biology and Bioinformatics	2018	分散型	1 人
Raisaro, Jean Louis ら[3]	IEEE/ACM Transactions on Computational Biology and Bioinformatics	2019	分散型	1 人
Froelicher, David ら[4]	Studies in Health Technology and Informatics	2020	分散型	1 人
Froelicher, David ら[5]	IEEE Transactions on Information Forensics and Security	2020	分散型	1 人
Roy Chowdhury, Amrita ら[7]	SIGMOD	2020	中央集権型	複数
Roth, Edo ら[6]	SOSP	2021	分散型	1 人
著者ら[8]	DEXA	2021	中央集権型	複数

第4章　準同型暗号に適したプライバシ保護パーティショニングアルゴリズムの提案

本章では，一次元ヒストグラムに対するレンジクエリ処理を対象とした，新たなプライバシ保護パーティショニングアルゴリズムを提案する．提案手法の貢献は，TFHE 上でのデータ依存パーティショニングの計算量を，線形増加に低減することである．また，提案手法はDAWA[25]のアイディアに基づき，高精度なレンジクエリ応答を達成することを目指す．本章で使用される記号を表 4.1 に示す．

表 4.1　記号の定義

記号	定義
$\boldsymbol{x}$	$\boldsymbol{x} = (x_1, x_2, \dots, x_{\lvert x\rvert})$のように表現される，入力ヒストグラムのベクトル
x_i	入力ヒストグラムのi番目の要素の値 $(1 \le i \le \lvert\boldsymbol{x}\rvert)$
ϵ_1	データ依存パーティショニングステップで消費されるプライバシパラメータ
ϵ_2	ラプラスメカニズムステップで消費されるプライバシパラメータ
$\boldsymbol{B}$	$\boldsymbol{B} = (b_1, b_2, \dots, b_m)$のように表現される，バケットの集合から構成されるパーティション $(1 \le m \le \lvert\boldsymbol{x}\rvert)$
b_j	$b_j = \{k, k+1, \dots, k+h\}$のように表現される，$j$番目のバケット内の$x_i$のインデックスの集合 $(1 \le j \le m)$（この時，b_jはx_kからx_{k+h}までのインデックスの集合である）
$\boldsymbol{S}$	$\boldsymbol{S} = (s_1, s_2, \dots, s_m)$のように表現される，各バケットの内の合計値の集合 $(1 \le m \le \lvert\boldsymbol{x}\rvert)$
s_j	$s_j = \sum_{i \in b_j}(x_i)$で与えられる，$j$番目のバケット内の合計値 $(1 \le j \le m)$
$[\cdot]$	暗号化された値（$[y]$はyの暗号文を意味する）

4.1 概要

提案手法は，レンジクエリ処理を対象とした，ϵ-差分プライバシを満足するパーティショニングアルゴリズムである．著者の知る限り，提案手法は，暗号文上での実行における計算量が入力データの大きさに対して線形増加となる最初のデータ依存パーティショニング手法である．提案手法の入力は，生データを暗号文上で集約することで構築される暗号化ヒストグラムである．暗号文上で差分プライバシを適用した後に復号することで，出力として平文で表現される差分プライベートヒストグラムを得る．

クラウドサーバは，(1) 計算サーバと (2) 復号サーバの 2 サーバモデルを用いる．両サーバはセミオネストかつ互いに非共謀であることを前提にしている．ここで，セミオネストとは，プロトコルの規則に従うが生データを盗み見ようとするエンティティであることを意味する．すなわち，提案手法は信頼できるサーバを必要としない．このような 2 サーバのセキュリティモデルは，DP-index[7]や DP-summary[8]と同様である．全ての生データは計算サーバ内でのみ取り扱われ，復号サーバは差分プライバシ適用済みデータのみを取り扱う．

提案手法は，DAWA[25]のデータ依存パーティショニング部の考えに基づき，レンジクエリ応答の精度の向上を目指す．提案手法は，(1) データ依存パーティショニングステップ (4.2 節) と (2) ラプラスメカニズムステップ (4.3 節) の 2 つのステップから構成される．それぞれのステップで消費されるプライバシパラメータをϵ_1，ϵ_2とすると，提案手法は，$\epsilon = \epsilon_1 + \epsilon_2$として，$\epsilon$-差分プライバシを満たす．上記のようなプライバシ前提は，DAWA[25]と同様である．データ依存パーティショニングステップでは，ヒストグラム内のデータを，連続する値が近いデータが同一のバケット内に格納されるように分割する．ラプラスメカニズムステップでは，それぞれのバケットごとにラプラスノイズを加算する．上記の 2 つのステップでは，バケット内の平均化による集計誤差とノイズ加算による摂動誤差がそれぞれ発生する．提案手法では，これら 2 つの誤差の和を小さくするような，準最適なパーティションを探索する．

提案手法の革新的な部分は，データ依存パーティショニングステップである．DAWA のデータ依存パーティショニング部では，一次元に限らない多次元のヒストグラムを対象に，集計誤差と摂動誤差の和で与えられる誤差推定値を最小化するような最適なパーティションを，考えられるパーティションの全パターンから探索している．平文上の実装では，動的計画法を適用することで，計算量を入力データの大きさに対する二乗増加に低減することができる．しかし，暗号文上の実装では，動的計画法を採用することができないため，計算コストが大きくなる．したがって，パーティショニングの計算コスト低減が必須となる．

提案手法のアイディアは，ヒストグラムの次元を一次元に限定し，パーティションの全パ

ターンではなく，考えられる全境界を確認することで，低計算コストでの近似的なパーティショニングを達成するというものである．上記の考えに基づき，隣接するデータの値が近い場合に，隣接するデータを統合する．暗号文上で実装される DAWA のデータ依存パーティショニング部の計算量が指数増加である[8]のに対し，提案手法の計算量は線形増加に低減される．提案手法の制限は，一次元で表されるヒストグラム以外に対応していないことである．多次元ヒストグラムを対象にした場合は，確認すべき境界の数が次元数に対して指数的に増加するため，提案手法の計算量もまた次元数に対して指数的に増加する．

4.2 データ依存パーティショニングステップ

本節では，提案手法のデータ依存パーティショニングステップ（**Algorithm 1**）を詳細に説明する．**Algorithm 1** では，暗号化されている値を$[\cdot]$で示す．本ステップは，入力ヒストグラム$[\boldsymbol{x}]$に基づき，ϵ_1-差分プライベートなパーティション$\boldsymbol{B}$を算出する．復号サーバで差分プライバシ適用済みデータを復号する時を除き，本ステップは計算サーバで暗号化データに対して動作する．本ステップの詳細な処理を以下に示す．

Algorithm 1 Data-aware Partitioning Step

function DATA-AWARE_PARTITIONING_STEP($[\boldsymbol{x}], \epsilon_1, \epsilon_2$)
  $t \leftarrow 1/\epsilon_2$
  $\boldsymbol{B} \leftarrow \{\}$
  $b_1 \leftarrow \{1\}$
  $i \leftarrow 1$
  // determine to merge or not
  for $k = 1\ to\ |\boldsymbol{x}| - 1$ **do**
    // z_k is sampled from $Laplace(\Delta diff_k/(\epsilon_1/2))$
    if $[diff_k] + z_k < t$ **then**
      $b_i \leftarrow b_i \cup \{k+1\}$
    else
      $\boldsymbol{B}$.append(b_i)
      $i \leftarrow i + 1$
      $b_i \leftarrow \{k+1\}$
    end if
  end for
  $\boldsymbol{B}$.append(b_i)
  return $\boldsymbol{B}$
end function

1. **閾値の設定**

与えられるプライバシパラメータϵ_2に基づき，閾値tを式(4.1)により計算する．

$$t = \frac{1}{\epsilon_2} \qquad (4.1)$$

tの値は，隣接するデータ間の差分がパーティショニング後に各バケットに加算されるラプラノイズの推定値より小さい場合に，隣接するデータを統合するという考えに基づく．

2. **パーティションとバケットの初期値の設定 p**

パーティション$\boldsymbol{B}$とバケットb_1の初期値を以下のように設定する．

$$\boldsymbol{B} = \{\} \qquad (4.2)$$

$$b_1 = \{1\} \qquad (4.3)$$

3. **隣接データ間の差分の計算**

以下の式(4.4)に基づき，隣接するデータ間の差分$diff_k$を計算する（$1 \le k \le |\boldsymbol{x}| - 1$）．

$$[diff_k] = |[x_{k+1}] - [x_k]| \ \ (1 \le k \le |\boldsymbol{x}| - 1) \qquad (4.4)$$

$[diff_k]$算出後，差分プライバシを満たすために，ラプラスノイズz_kを$[diff_k]$に加える．この時，ラプラスノイズのプライバシパラメータを，ϵ_1に設定する．$diff_k$の敏感度を$\Delta diff_k$とする．ラプラスメカズムのスケールを$(\Delta diff_k/(\epsilon_1/2))$で与えることで，出力パーティションは$\epsilon_1$-差分プライベートとなる．差分プライバシの証明は，本節の「**セキュリティ分析**」に示す．また，$\Delta diff_k$の設定については，「**命題 2**」で後述する．

4. **統合**

$([diff_k] + z_k \le t)$が真である場合，kと$k+1$を同一のバケットに格納する．すなわち，b_iが$b_i \cup k+1$に更新される．$([diff_k] + z_k \le t)$が偽である場合は，b_iを新たに$\boldsymbol{B}$に追加する．ここで，$([diff_k] + z_k \le t)$の比較結果は暗号文上で取得される．すなわち，$([diff_k] + z_k \le t)$が真であれば[1]を，偽であれば[0]を格納する回路を TFHE 上で構成する．比較結果（[1]または[0]）は復号サーバによって復号される．この時，復号されるデータは[1]または[0]の比較結果のみであり，かつ，$[diff_k] + z_k$は差分プライベートであるため，復号サーバに対して$\boldsymbol{x}$のプライバシは漏洩しない．

5. **比較と統合の繰り返し**

「**3. 隣接するデータ間の差分の計算**」と「**4. 統合**」を，$[\boldsymbol{x}]$内の全ての隣接データ間で繰り返すことで，ϵ_1-差分プライベートなパーティション$\boldsymbol{B}$を得る．

例： 入力ヒストグラムを$[\boldsymbol{x}] = ([1], [1], [6], [7], [7], [2], [3])$，プライバシパラメータを$\epsilon = 0.5$ $(\epsilon_1 = 0.25\epsilon, \epsilon_2 = 0.75\epsilon)$とした時の，データ依存パーティショニングステップの例を考える．ここで，閾値は$t = 1/\epsilon_2 = 2.67$と与える．$k = 1$において，TFHE 上で式(4.4)を適用することで，$[diff_1] = [0]$を得る．ここでは，説明の簡略化のため，ラプラスノイズz_1を 0 で

あるとする．$[diff_1]+z_1\ (=[0])$と$t\ (=2.67)$を比較することで，x_1とx_2を統合する．次に，$[diff_2]=[5]$を計算し，z_2を 0 であるとすると，$[diff_2]+z_2\ (=[5])$と$t\ (=2.67)$の比較結果から，x_2とx_3を分割する．上記より、$b_1=\{1,2\}$を得る．上記の処理を繰り返すことで，パーティション$\boldsymbol{B}=\{\{1,2\},\{3,4,5\},\{6,7\}\}$を出力する．

セキュリティ分析： データ依存パーティショニングステップは，計算サーバ上での演算を TFHE で暗号化すると共に，復号サーバに送信するデータに差分プライバシを適用することで，計算サーバと復号サーバから生データを保護する．出力パーティション$\boldsymbol{B}$は両サーバに公開されるが，各パーティションを構成するヒストグラムは暗号化されたままであるため，生データのプライバシは保護される．また，復号サーバに$([diff_k]+z_k\le t)$の比較計算時に用いるtが漏洩したとしても，$[diff_k]+z_k$は差分プライベートであるため，$\boldsymbol{x}$のプライバシは漏洩しない．

プライバシ保護を保証するための残りの課題は，データ依存パーティショニングステップが差分プライバシを満たすことの証明である．本ステップは，ラプラスメカニズム[14]を使用しており，プライバシパラメータが入力として与えられることを考えると，本ステップの敏感度を定義することで，差分プライバシの証明を達成することができる．しかし，$\boldsymbol{B}$が入力ヒストグラム$\boldsymbol{x}$の値を含まない（$\boldsymbol{B}$は$\boldsymbol{x}$のインデックスのみ保持する）ため，本ステップの敏感度を定義することは困難である．そこで，$\Delta diff_k$に注目することで，本ステップの差分プライバシを証明する．$\Delta diff_k$及び差分プライバシの証明を下記に示す．

命題 1： $diff_k$の敏感度は 1 以下である．

命題 1 の証明： $\boldsymbol{x},\boldsymbol{x}'$を隣接ヒストグラム（隣接データベースから構築される 2 つのヒストグラム）とする．$\boldsymbol{x},\boldsymbol{x}'$は隣接データベース内のデータをカウントすることで構築するため，隣接ヒストグラム内の 1 つのペア$x_i,x'_i\ (1\le i\le|\boldsymbol{x}|)$で 1 異なっている．ここで，$i$は隣接ヒストグラム間で値が異なっているデータのインデックスを指す$(1\le i\le|\boldsymbol{x}|)$.

$i=k$または$i=k+1$の時, $diff_k$は隣接ヒストグラム間で 1 のみ異なる. したがって, $diff_k$の敏感度である$\Delta diff_k$は以下のように定義される.

$$\Delta diff_k=1\ \ (1\le k\le|\boldsymbol{x}|-1, i=k\ or\ k+1) \quad (4.5)$$

一方で，$i\ne k$かつ$i\ne k+1$の時，$diff_k$は隣接ヒストグラム間で同一になるため，$\Delta diff_k$は以下のように定義される．

$$\Delta diff_k=0\ \ (1\le k\le|\boldsymbol{x}|-1, i\ne k\ or\ k+1) \quad (4.6)$$

式(4.5), (4.6)より，$diff_k$の敏感度は以下のように定義される．

$$\Delta diff_k\le 1\ \ (1\le k\le|\boldsymbol{x}|-1) \quad (4.7)$$

命題 2： 提案手法のデータ依存パーティショニングステップはϵ_1-差分プライバシである．

命題 2 の証明： $diff_k$を算出する演算は，$[\boldsymbol{x}]$内の全ての隣接するデータ間で繰り返される．

$i = 1$または$i = |\boldsymbol{x}|$である時，$\Delta diff_k = 1$を満たすkは，隣接するデータ間の全ての差分を算出する処理を通して，ただ一つのみ存在する．具体的に，$i = 1$である時は，$\Delta diff_1$のみ 1 となり，他の$\Delta diff_k$ $(2 \leq k \leq |\boldsymbol{x}| - 1)$は全て 0 となる（図 4.1(a)）．同様に，$i = |\boldsymbol{x}|$である時は，$\Delta diff_{|\boldsymbol{x}|-1}$のみ 1 となり，他の$\Delta diff_k$ $(1 \leq k \leq |\boldsymbol{x}| - 2)$は全て 0 となる．一方で，$i = l$ $(2 \leq l \leq |\boldsymbol{x}| - 2)$の時は，$\Delta diff_{l-1}$と$\Delta diff_l$の 2 つの敏感度が 1 となり他の$\Delta diff_k$ $(1 \leq k \leq |\boldsymbol{x}| - 1, k \neq l - 1 \; or \; l)$ は全て 0 になる（図 4.1(b)）．

上記より，隣接するデータ間の全ての差分を算出する処理内で，$\Delta diff_k$は多くとも 2 つの点で 1 となる．したがって，差分プライバシの直列合成定理[36]より，ラプラスメカニズムの敏感度を$\epsilon_1/2$に設定することで，全ての差分を算出する処理の出力はϵ_1-差分プライバシを満たす．一度全ての$diff_k$ $(1 \leq k \leq |\boldsymbol{x}| - 1)$を算出すると，$[\boldsymbol{x}]$は$\boldsymbol{B}$に一切影響を与えないため，全ての差分を算出する処理が差分プライバシを満たすことは，データ依存パーティショニングステップが差分プライバシを満たすことと同義である．したがって，$\Delta diff_k = 1$として，スケール$\Delta diff_k/(\epsilon_1/2)$のラプラスメカニズムを適用することで，データ依存パーティショニングステップはϵ_1-差分プライバシを満たす．

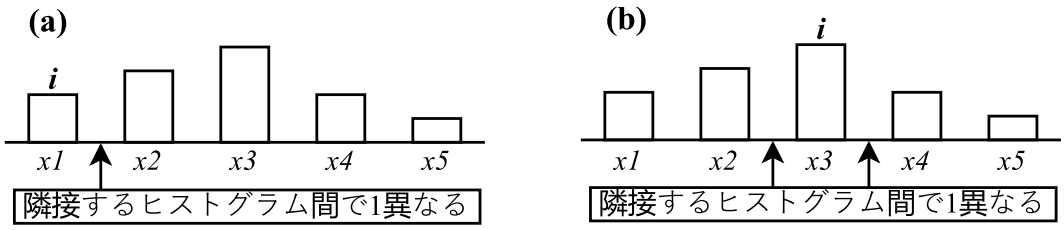


図 4.1　要素数 5 のヒストグラムにおける（a）$i = 1$ の時と（b）$i = 3$ の時の例
（i は隣接ヒストグラム間で値が異なっているデータのインデックスを指す）

4.3 ラプラスメカニズムステップ

本節では，提案手法のラプラスメカニズムステップを詳細に説明する．本ステップは，入力ヒストグラム$[\boldsymbol{x}]$とパーティション$\boldsymbol{B}$から，ϵ-差分プライベートなヒストグラムを出力する．本ステップの詳細な処理を以下に示す．

1. **各バケット内の合計値の算出**
   入力ヒストグラム$[\boldsymbol{x}]$及びパーティション$\boldsymbol{B}$が与えられると，計算サーバは各バケット内のデータの合計値を暗号文上で算出し，$[\boldsymbol{S}]$を取得する．

2. **ラプラスメカニズム**
   計算サーバは，ラプラスノイズを$[\boldsymbol{S}]$に加算し，$[\boldsymbol{S}']$を得る．ここで$\boldsymbol{S}'$は，各バケット内の合計値にラプラスメカニズムを用いてノイズを加算した後の値を表すベクトル

である．この時，ラプラスメカニズムのパラメータとして，プライバシパラメータϵ_2を使用する．また，敏感度に関しては，ヒストグラムの敏感度である 2 を使用する．$\boldsymbol{S}'$は，差分プライバシの直列合成定理[36]より，$\epsilon = \epsilon_1 + \epsilon_2$として，$\epsilon$-差分プライバシを満たす．ラプラスメカニズム適用後に，$[\boldsymbol{S}']$を復号サーバに送信する．

3. **復号**

復号サーバは，$[\boldsymbol{S}']$を復号し，$\boldsymbol{S}'$を取得する．この時，$\boldsymbol{S}'$には差分プライバシが適用されているため，復号サーバに$\boldsymbol{x}$のプライバシは漏洩しない．復号サーバは，計算サーバに$\boldsymbol{S}'$を送信する．

4. **平均化**

計算サーバは，各バケット内の要素数に応じて，$\boldsymbol{S}'$を平均化し，ϵ-差分プライベートなヒストグラム$\boldsymbol{x}'$を構築する．$\boldsymbol{x}'$は，$\boldsymbol{x}' = (x'_1, x'_2, \ldots, x'_{|x'|})$のように表され，差分プライベートヒストグラムとして，計算サーバ内に保存される．

例： ラプラスメカニズムステップの例を図 4.2 に示す．例では，入力ヒストグラムを$[\boldsymbol{x}] = ([1], [1], [6], [7], [7], [2], [3])$，パーティションを$\boldsymbol{B} = \{\{1, 2\}, \{3, 4, 5\}, \{6, 7\}\}$と想定する．各バケット内の合計値を暗号文上で算出することで，$[\boldsymbol{S}] = ([2], [20], [5])$を得る．ラプラスメカニズム適用後の$[\boldsymbol{S}]$が$[\boldsymbol{S}'] = ([2.4], [20.7], [4.8])$であるとすると，復号サーバと協力することで，$\boldsymbol{S}' = ([2.4], [20.7], [4.8])$を取得する．最後に，各バケット内の要素数で平均化することで，$\boldsymbol{x}' = (1.2, 1.2, 6.9, 6.9, 6.9, 2.4, 2.4)$を得る．

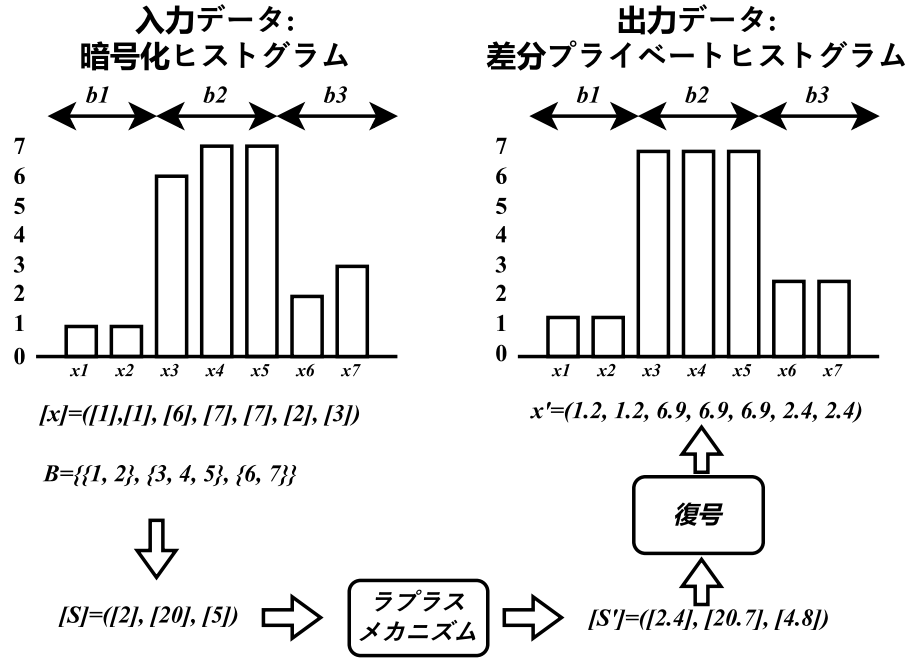


図 4.2　提案手法のラプラスメカニズムの例

4.4 脅威モデル

提案手法を適用したレンジクエリ応答システムを図 4.3 に示す．本システムは，(1) データ所有者，(2) 計算サーバ，(3) 復号サーバ，(4) データ解析者の 4 つのエンティティから構成される．本システムは，計算サーバ，復号サーバ，データ解析者に対して，データ所有者が提供する生データの保護を達成する．本システムの脅威モデルを以下に示す．

- 計算サーバと復号サーバはセミオネストである（すなわち，プロトコルの規則に従うが生データを盗み見ようとするエンティティである）と仮定する．
- データ解析者は信頼できないと仮定する．
- 計算サーバと復号サーバは，互いに及び他のエンティティと非共謀である．

本システムでは，計算サーバ，復号サーバ，データ解析者に対して，差分プライバシが適用された後のデータが漏洩することを許容している．具体的には，システムの動作過程で計算サーバ及び復号サーバは，パーティション$\boldsymbol{B}$及び差分プライベートヒストグラム$\boldsymbol{x}'$を保持する．また，データ解析者は複数のレンジクエリを実行することで，$\boldsymbol{B}$及び$\boldsymbol{x}'$を推定することができる．一方で，差分プライバシが適用される前のヒストグラム$\boldsymbol{x}$は，TFHE により暗号化されているため，計算サーバ，復号サーバ，データ解析者には漏洩しない．また，$\boldsymbol{x}$のプライバシは差分プライバシにより$\boldsymbol{B}$及び$\boldsymbol{x}'$から漏洩しない．

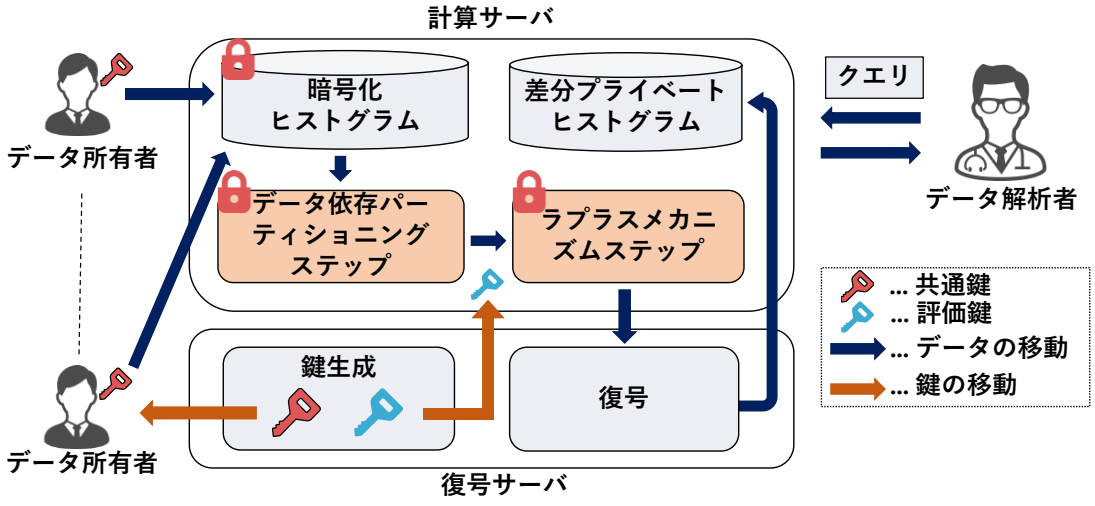


図 4.3　提案手法を適用したレンジクエリ応答システム

4.5 提案手法を適用したレンジクエリ応答システム

図 4.3 のレンジクエリ応答システムの動作を以下に示す.

1. **鍵生成**
   復号サーバは，TFHE の共通鍵を生成し，データ所有者に共通鍵を送信する.

2. **暗号化**
   データ所有者は，自身の生データを共通鍵で暗号化し，計算サーバに暗号化データを送信する．データ送信を終えたデータ所有者は，その後システムに関与しない.

3. **暗号化ヒストグラムの構築**
   計算サーバは，TFHE を用いて暗号化データを集約し，暗号文ヒストグラムを構築する.

4. **データ依存パーティショニングステップ**
   計算サーバは，暗号化ヒストグラムに対して，データ依存パーティショニングステップを実行し，差分プライベートなパーティションを算出する.

5. **ラプラスメカニズムステップ**
   計算サーバは，ラプラスメカニズムを用いて暗号化ヒストグラムにノイズを加算する．そして，差分プライバシ適用済み暗号化ヒストグラムを復号サーバに送信する.

6. **復号**
   復号サーバは，差分プライバシ適用済み暗号化ヒストグラムを復号し，差分プライベートヒストグラムを取得する．次に，復号サーバは差分プライベートヒストグラムを計算サーバに送信し，計算サーバは差分プライベートヒストグラムを保存する．なお，復号サーバによって復号される全てのデータには，差分プライバシが適用済みであるため，復号サーバを含めた全てのエンティティに対して，生データのプライバシは漏洩しない.

7. **クエリ処理**
   データ解析者は計算サーバにクエリを送信し，計算サーバは事前構築した差分プライベートヒストグラムを用いて応答する.

本システムでは，クエリ入力後の処理は全て平文上で行うため，入力クエリに対する応答時間が短縮される．さらに，一度差分プライバシを適用したヒストグラムから全てのクエリに応答するため，プライバシ予算に起因するクエリ応答の回数制限から解放される．これら 2 つの利点は，先行研究の DP-index[**エラー! ブックマークが定義されていません。**]及び DP-summary[**エラー! ブックマークが定義されていません。**]と同様である.

第5章　　評価実験

本章では，提案手法の性能を評価する．提案手法の貢献は，精度を劣化させることなく，暗号文上でのデータ依存パーティショニングを高速化させることである．上記の貢献を確認するために，本評価実験では提案手法の（1）実行時間及び（2）精度を評価する．

5.1 実験条件

本評価実験で使用されたプログラムは，C++を用いて実装されており，TFHE のバージョンは 1.1 を用いた．本評価実験の実行環境を表 5.1 に示す．本評価実験で TFHE を使用する場合は，浮動小数点方式と比較した実装の簡易性を考慮して，固定小数点方式を採用している．データ依存パーティショニングを採用するアルゴリズムにおける 2 つのプライバシパラメータϵ_1とϵ_2の比は，DAWA での評価実験[25]と同様に，$\epsilon_1 : \epsilon_2 = 1:3$とする．

表 5.1　実行環境

項目	値
CPU モデル	Intel(R) Xeon(R) Platinum 8280 × 2
コア数	56
メモリサイズ	1.5 TB
OS	CentOS Linux release 7.6.1810 (Core)
Linux バージョン	3.10.0-957.21.3
g++バージョン	7.3.1

データセット： 精度評価実験で使用するデータセットを表 5.2 に示す．本データセットは DAWA での精度評価実験[25]で用いられたデータセットと同じであり，GitHub[2]上で公開さ

2 https://github.com/dpcomp-org/dpcomp_core

れている．7 つのデータセットは全て，ドメインサイズ（要素数）が 4,096 である一次元ヒストグラムで表現される．

ワークロードと損失関数： 精度評価実験では，(1) プレフィックスワークロード，(2) 単一ワークロード，(3) 一様ランダムワークロードの 3 種類のワークロードを使用する．プレフィックスワークロードは，それぞれクエリが$[1, i]$ $(i \in [1, n])$であるn個のレンジクエリから構成される（nはドメインサイズを指す）．単一ワークロードは，それぞれクエリが$[i, i]$ $(i \in [1, n])$であるn個のレンジクエリから構成される．一様ランダムワークロードは，$[1, n]$間から二点をランダムに取り，それら二点を始点及び終点とするレンジクエリをn個（すなわち，ドメインサイズと同じ個数）生成することで構築される．プレフィックスワークロードは，DPBENCH の精度評価実験[37]で使用され，単一ワークロード及び一様ランダムワークロードは DAWA の精度評価実験[25]で使用されている．

損失関数として，精度評価実験では L2 ノルム損失関数を使用する．上記の損失関数は，DPBENCH の精度評価実験[37]で使用された損失関数と同じである．

表 5.2　データセットの概要

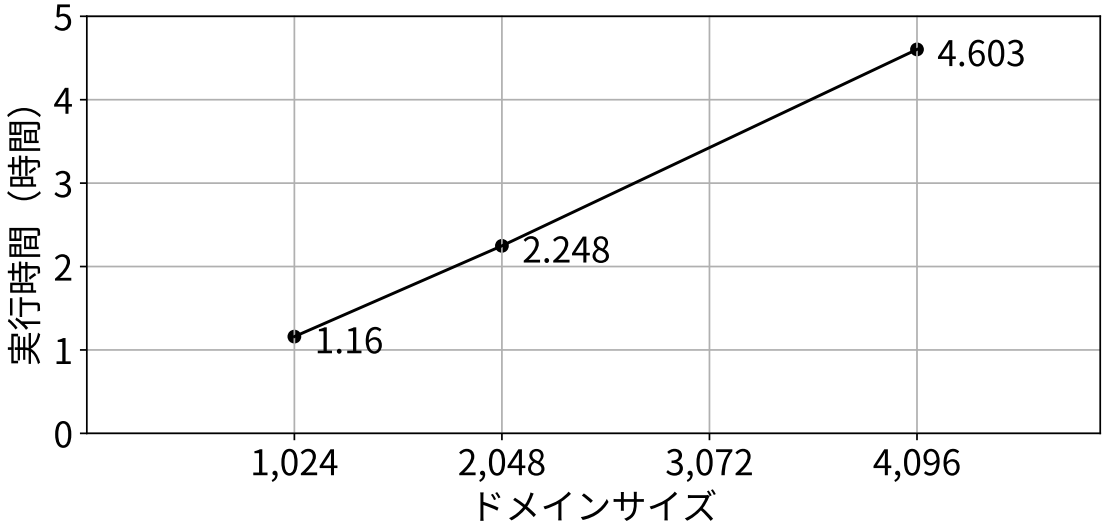
データセット	説明	ヒストグラムの属性
Nettrace [43]	大学のゲートウェイルータで収集した IP レベルのネットワーク追跡データ	内部ホストの IP アドレス
Adult [44]	米国国勢調査データ	資本損失
Medical Cost [45]	国際的な家庭及びホスピスケアの調査（2007 年～）	個人の医療費
Search Logs [43]	検索ワード「オバマ」の出現頻度の変化（2004 年～2010 年）	検索クエリの記録
Income [46]	IPUMS 米国地域社会調査データ（2001 年～2011 年）	個人所得
Patents [47]	米国特許のサブセットにおける引用ネットワークに関するデータ	タイムスタンプ
HepPh [47]	arXiv 上の高エネルギー物理学プレプリントの引用ネットワークに関するデータ	タイムスタンプ

5.2 実行時間の評価

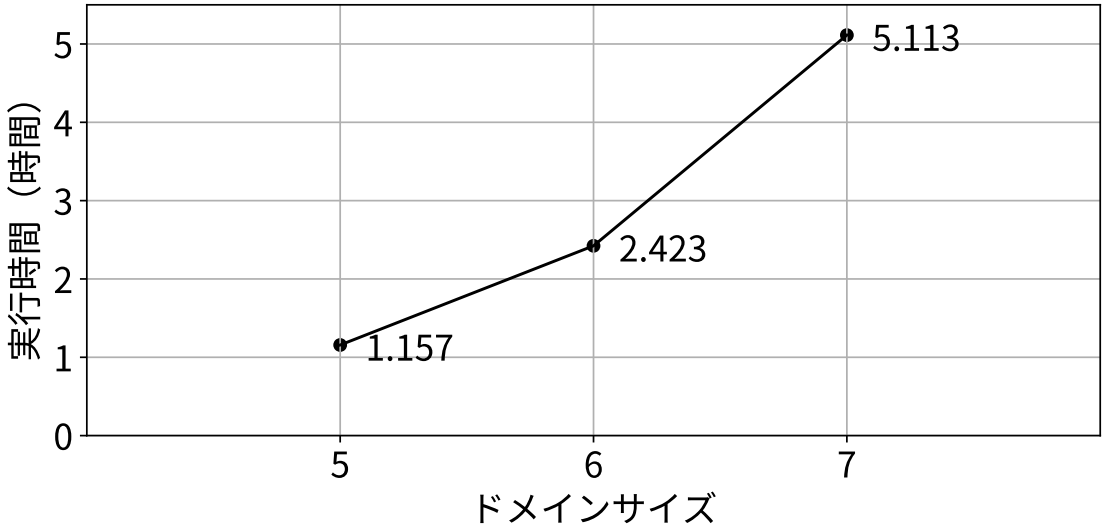
提案手法の実行時間を，DAWA のデータ依存型パーティショニング部を TFHE 上で再現した手法である DP-summary[8]と比較する．入力ヒストグラムのデータセットは，0 から 100 の整数をランダムに生成した. 準同型演算ではデータの値によって計算時間が変化しないことから，入力ヒストグラムの値は実行時間に大きな影響を与えない．実行時間は，入力ヒストグラムのドメインサイズと暗号文を表現するビットサイズのみに依存する[8]．ドメインサイズに対する実行時間の変化を確認するため，本評価実験ではビットサイズを固定した．

56 スレッドで並列化した提案手法及び DP-summary の実行時間を，C++の標準ライブラリに含まれる chrono 関数を用いて 10 回測定し，平均を取る．ここで，実行時間とは，暗号化ヒストグラムから差分プライベートヒストグラムを構築するまでに要する時間を指す．TFHE の暗号文を表現するビットサイズは，24 ビットである．全 24 ビットから，1 ビットを符号部に，19 ビットを整数部に，4 ビットを小数部に割り当てた．

図 5.1 は，提案手法の実行時間（図 5.1（a））及び DP-summary の実行時間（図 5.1（b））の測定結果を示す．図 5.1（a）より，提案手法の実行時間は，ドメインサイズに対して線形に増加することが確認できる．また，ドメインサイズが 4,096 であるヒストグラムに対する提案手法の実行時間が，4.603 時間である．提案手法の実行は，データ解析者がクエリを入力する前に一度のみ必要であるため，提案手法の実行時間は許容範囲であると言える．一方で，図 5.1（b）より，DP-summary の実行時間はドメインサイズに対して指数増加しており，ドメインサイズが 7 であるヒストグラムに対して 5 時間を超える実行時間が要求される．したがって，ドメインサイズが大きい場合，DP-summary の実行時間を許容することはできない．上記の結果より，提案手法は，暗号文上でのデータ依存パーティショニングの実行時間を，実用的な計算時間に短縮したと言える．



（a）　提案手法



（b）　DP-summary

図 5.1　ドメインサイズごとの実行時間の測定結果

5.3 精度の評価

本節では，(1) 暗号文上でのデータ依存パーティショニング手法と提案手法の精度比較及び（2）平文上でのデータ依存パーティショニング手法と提案手法の精度比較の 2 つの精度評価実験を行う．

暗号文上でのデータ依存パーティショニング手法と提案手法の精度比較では，競合手法として，Identity[14]と DP-summary[8]を用いる．Identity とは，入力ヒストグラムの個々のデータにラプラスノイズを加算するナイーブな差分プライバシアルゴリズムである．Identity は，平文上の実装を前提にした手法であり，加算のみで構成されるため，容易に暗号文上に実装できる．DP-summary は，最先端のデータ依存パーティショニングアルゴリズムである DAWA[25]のデータ依存パーティショニング部を，暗号文上に実装した手法である．

平文上でのデータ依存パーティショニング手法と提案手法の精度比較では，データ依存パーティショニングの代表的な競合手法である PHP[17]，StructureFirst[18]，AHP[19]を使用する．PHP 及び AHP では，DAWA や提案手法と同様に，ヒストグラム内で分割されるバケット数が，入力データに応じて動的に決定される．一方で，StructureFirst では，バケット数kを，入力として与える必要がある．kの値は，アルゴリズムの使用者に委ねられるが，StructureFirst の評価実験では，ドメインサイズをnとして，kの値が約$n/10$である時に，高精度を達成する傾向があることが報告されている[18]．したがって，本精度評価実験においては，$k = 400$とした．

TFHE では，復号結果はノイズを含まないため，暗号化に起因する精度劣化は発生しない．そのため，本精度評価実験では，暗号文を対象とした手法においても，平文上でアルゴリズムの処理をシミュレーションする．また，差分プライバシ下において，固定小数点方式を採用することで切り落とされる小数部の値によって発生する誤差は，無視できるほど小さいことが報告されているため[8]，平文上でのシミュレーションでは浮動小数点方式を採用した．誤差の測定方法に関しては，(1) プレフィックスワークロード，(2) 単一ワークロード，(3) 一様ランダムワークロードの 3 つのワークロードを，表 5.2 に示される 7 つの実データセットに対して 1000 回実行し，平均 L2 ノルムを取得した．

5.3.1　暗号文を対象とした競合手法との比較

本項では，提案手法の精度が DP-summary（すなわち，最先端のデータ依存パーティショニング手法）の精度に対して劣化していないことを確認する目的で，DP-summary と提案手法の精度を比較する．また，暗号文上においてもパーティショニングを行うことの優位性を確認するために，Identity と提案手法を比較する．

図 5.2 から図 5.4 はそれぞれ，プレフィックスワークワークロード，単一ワークロード，一様ランダムワークロードにおける精度の測定結果を示す．横軸はデータセット，縦軸は平均 L2 ノルムを表す．

3 つの実験結果より，提案手法の精度が，DP-summary の精度と同等以上であることが確認できる．具体的に，図 5.2（c）に注目すると，Nettrace，Adult，Medical Cost，Patents において，提案手法は DP-summary の精度を上回る．一方で，Search Logs, Income, HepPh においては，DP-summary の精度は提案手法よりも低い誤差を達成している．

また，提案手法のデータセットに対する特徴（すなわち，どのデータセットに対して高精度を達成するか）も DP-summary と同様であった．具体的に，図 5.2（c）に注目すると，提案手法は Nettrace，Adult，Medical Cost，Search Logs に対して Identity よりも優れた精度を示しており，これは DP-summary と同様の特徴である．図 5.2 から図 5.4 より，このような特徴は，3 つのワークロード及び複数のプライバシパラメータを通して類似していることが分かる．上記の結果より，DP-summary が DAWA のデータ依存パーティショニング部を暗号文上に再現していることを考えると，提案手法は最先端のデータ依存パーティショニング手法と比較しておおよそ同等の精度を達成したと言える．

Identity との比較に関して，提案手法は入力ヒストグラムがパーティショニングに適したデータである場合に Identity より優れた精度を達成した．ここで，パーティショニングに適したデータとは，「連続した値が近いデータがまとまって分布するような傾向を持つヒストグラム」を指し，本精度評価実験で使用したデータでは Nettrace, Adult, Medical Cost が該当する．一方で，Income，Patents，HepPh のような，連続する値の大きさが異なる傾向を持つデータセットでは，Identity が提案手法の精度を上回る傾向が見られた．この傾向は，DP-summary と同様である．したがって，提案手法は，入力データが連続した値が近いデータがまとまって分布するような傾向を持つヒストグラムの場合，Identity に対して精度の観点から優位となることが分かった．

入力ヒストグラムがパーティショニングに適したデータであるかは，元データ（ヒストグラム化する前のデータ）の特性やヒストグラムの属性に依存する．具体的に，Nettrace では内部ホストの IP アドレスがヒストグラムの属性となっており，ヒストグラムの値は各内

部ホストによって行われた外部接続の数を報告している．一般的なネットワークでは外部と頻繁に通信を行う内部ホストは限られており，大半の内部ホストは外部接続を行わない．このような特性がヒストグラムに反映されており，Nettrace はヒストグラム内の 96.61%のデータが 0 であるような，パーティショニングに適したデータとなっている．同様に，Adult では資本損失がヒストグラムの属性となっており，多くの人の資本損失が 0 であるという背景から，ヒストグラム内の 97.81%の値が 0 であるヒストグラムを形成する．また，Medical Cost に関しても，お年寄りや体の弱い人は医療費を多く必要とする一方で健康な若者はあまり医療費を必要としないため，ヒストグラム内の 74.80%のデータが 0 であり，1 や 2 といった小さい数値も頻繁に分布するようなヒストグラム構成となっている．一方で，特許及び物理学論文の引用数をそれぞれ時系列に報告する Patents 及び HepPh には，同一の特許や論文でも新しい技術の誕生や社会の需要の変化によって年代ごとに引用回数にばらつきがあるという背景から，ヒストグラム内の隣接するデータの値がばらつくという傾向がある．このような性質を持つデータセットでは，パーティショニングを行っても精度が向上しないことがある．Income に関しても，個人所得はさまざまであることから，パーティショニングに適さないヒストグラムとなっている．

上記のように，パーティショニングに適したデータセットは，ヒストグラム内に値が 0 であるデータを多く持つ傾向がある．各データセットのカウント値が 0 になる割合を表 5.3 に示す．表 3 より，カウント値が 0 になる割合が 50%を超えているとパーティショニングを行うことで精度が向上し，50%を下回る場合は Identity（すなわち，パーティショニングを行わない手法）の方が高精度である．したがって，入力ヒストグラム内の値が 0 となるデータの数をカウントすることが，適したアルゴリズムを選択する際の指標になる可能性がある．具体的には，「ヒストグラム内の 0 の数がドメインサイズの 50%を超えている場合は提案手法を採用し，50%以下である場合は Identity を採用する」などが考えられる．入力ヒストグラム内の値が 0 となるデータ数のカウントは，TFHE を使用することで暗号文上においても，ドメインサイズに対して線形増加の計算量で行うことができる．

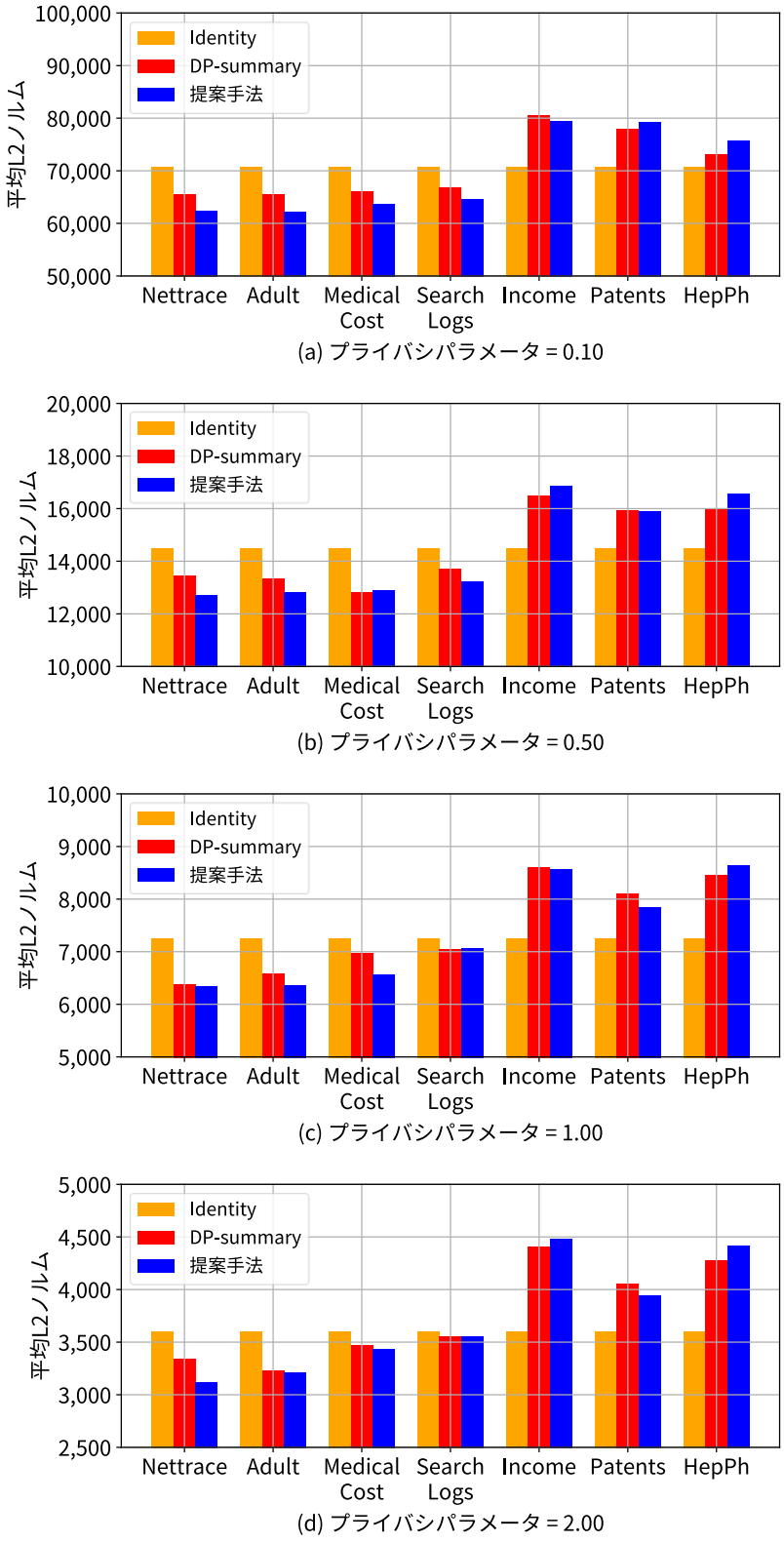


図 5.2　Identity，DP-summary，提案手法の精度比較結果
（プレフィックスワークロード）

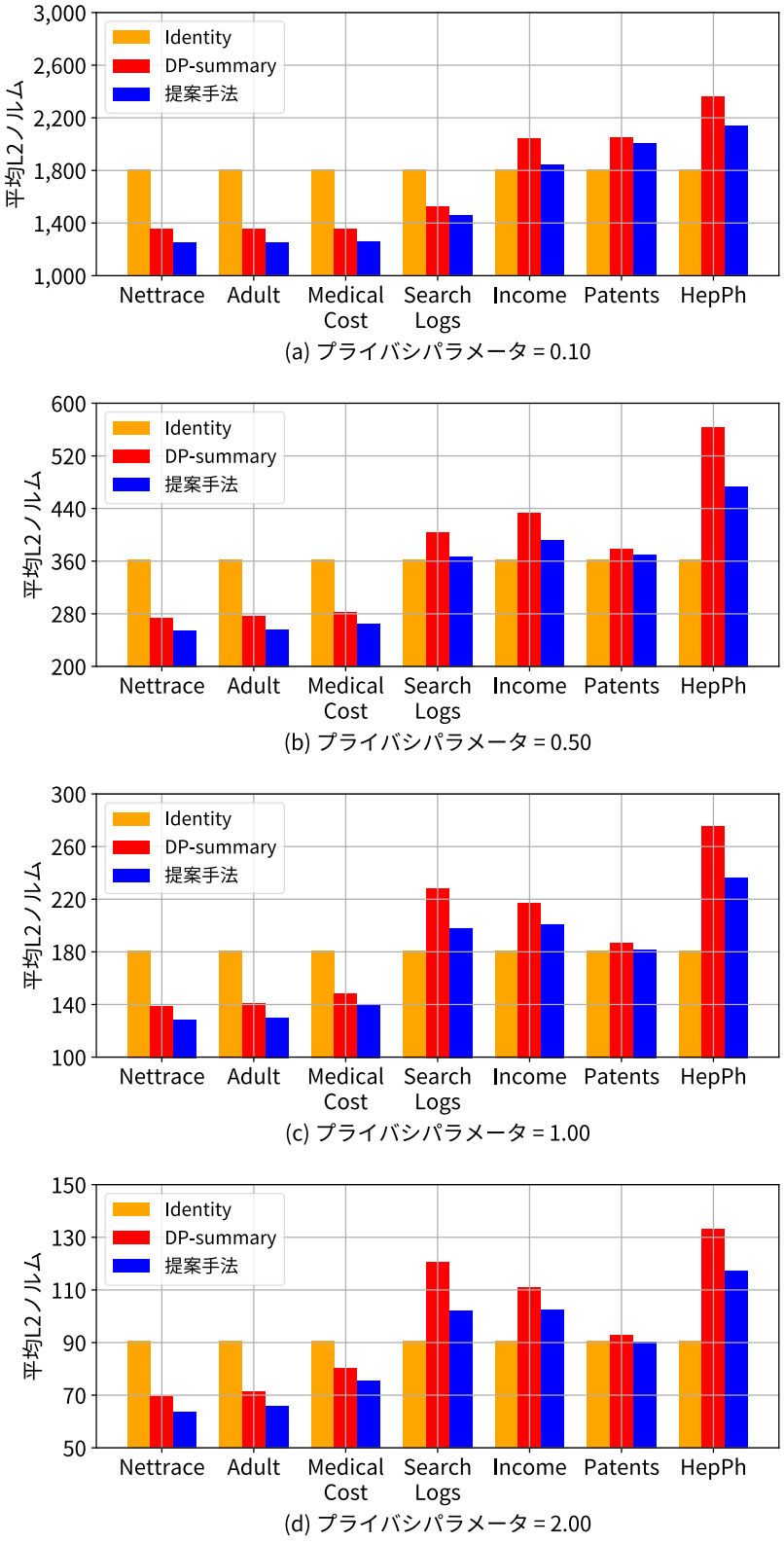


図 5.3　Identity，DP-summary，提案手法の精度比較結果
(単一ワークロード)

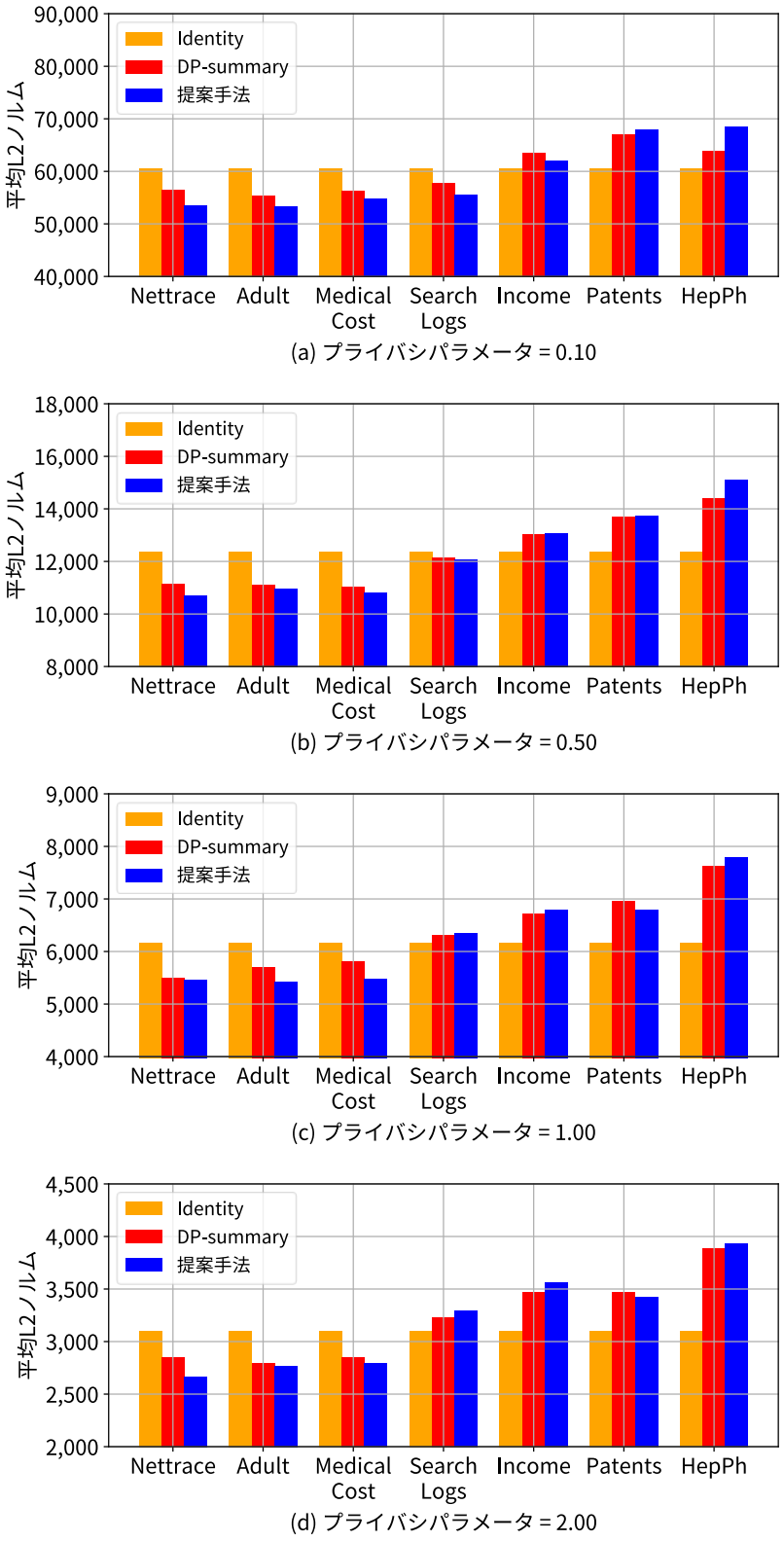


図 5.4　Identity，DP-summary，提案手法の精度比較結果
(一様ランダムワークロード)

表 5.3　各データセットのカウント値が 0 になる割合[37]

データセット	カウント値が 0 になる割合
Nettrace	96.61%
Adult	97.80%
Medical Cost	74.80%
Search Logs	51.03%
Income	44.97%
Patents	6.20%
HepPh	21.17%

5.3.2　平文を対象とした競合手法との比較

本項では，提案手法が，平文を対象とした既存手法と比較した時，精度の観点から実用的であることを示す目的で，提案手法を 3 つの代表的な競合手法[17, 18, 19]と比較する．

図 5.5 から図 5.7 は各データセットに対する，平文上の代表的なデータ依存パーティショニング手法[17, 18, 19]と提案手法の，3 つのワークロードの平均 L2 ノルムを示す．測定結果より，いくつかのデータセット（特に，Search Logs，Income，Patents，HepPh）において，提案手法は他競合手法と比較して高い精度を示した．

図 5.5 から図 5.7 から分かるように，競合のパーティショニングアルゴリズムの性能は，データセットの特性に対して敏感である．すなわち，特定のデータセットに対しては低誤差を達成するが，他のデータセットでは大きな誤差を生む．一方で，提案手法は，7 つのデータセットに対して同程度の精度を示しており，データセットの特性にとらわれることなく安定した精度を発揮することが分かる．上記の結果から，提案手法は，データセットに対するアルゴリズムの選択性の観点においても，3 つの競合手法[17, 18, 19]と比較して優れていると言える．

PHP と AHP が入力データの特性に対して敏感である理由は，PHP と AHP がバケット数を小さくする傾向があるアルゴリズムであるためと考える．PHP，AHP，提案手法で算出されたバケット数の 1,000 回平均を表 5.4 に示す．表 5.4 内の「DAWA (Non-DP)」は，DAWA のパーティショニングアルゴリズムを, 差分プライバシを適用しないで（すなわち，ランダムノイズを加えないで）実行した時のバケット数を示している．DAWA (Non-DP)の値はランダム性を含まないため，一意に定まる．また，パーティショニング時のエラーコスト推定にプライバシパラメータϵを使用していることから，プライバシパラメータが小さいほど，DAWA (Non-DP)の値は小さくなる．DAWA が考えられる全てのパーティションのパターンから最小コストのパーティションを探索する最高精度のアルゴリズムであることから，DAWA (Non-DP)の値は各データセット及びプライバシパラメータϵに対する最適なバケット数であると考えることができる．

表 5.4 より，PHP と AHP は DAWA (Non-DP)と比較して，バケット数が小さいことが分かる．すなわち，PHP と AHP はバケット数が小さくなる傾向があるアルゴリズムであると言える．バケット数が小さいということは，平均化に起因する集計誤差が大きく，ノイズに起因する摂動誤差が小さくなることを意味する．DAWA (Non-DP)でバケット数が大きくなるようなデータセットにおいて，PHP や AHP のようにバケット数が小さいと，値が大きく異なるデータを一つのバケットに統合している可能性が高い．このような場合は，集計

誤差が大きくなり，その結果としてクエリ応答の精度が劣化する．図 5.6 に注目すると，Patents に対する PHP と AHP の精度がプライバシパラメータ間で大きく異なっていないことが分かる．これは，プライバシパラメータに起因する摂動誤差の大きさが無視されるほどに，集計誤差が大きいことを意味している．一方で，提案手法は DAWA(Non-DP)と比較して，バケット数が大きくなる傾向がある．パーティショニングにおいてバケット数が大きい場合は，アルゴリズムは Identity に近づく．そのため，バケット数が過大であったとしても，クエリの応答誤差の劣化はバケット数を過少にする時と比べて小さい．上記の理由から，バケット数を過少にする傾向がある PHP 及び AHP は，バケット数を過大にする傾向がある提案手法と比較して，データセットとの相性が悪い場合に大きな誤差を生じるような，入力データセットに対して敏感なアルゴリズムであると考える．

同様に，StructureFirst はバケット数kをアルゴリズムの動作前に与える必要があるため（本評価実験では$k = 400$），kの大きさが入力データセットに対して充分でない場合は，集計誤差が増加する．そのため，kの値が事前に判明していな場合は，StructureFirst は入力データセットに対して敏感なアルゴリズムとなる．

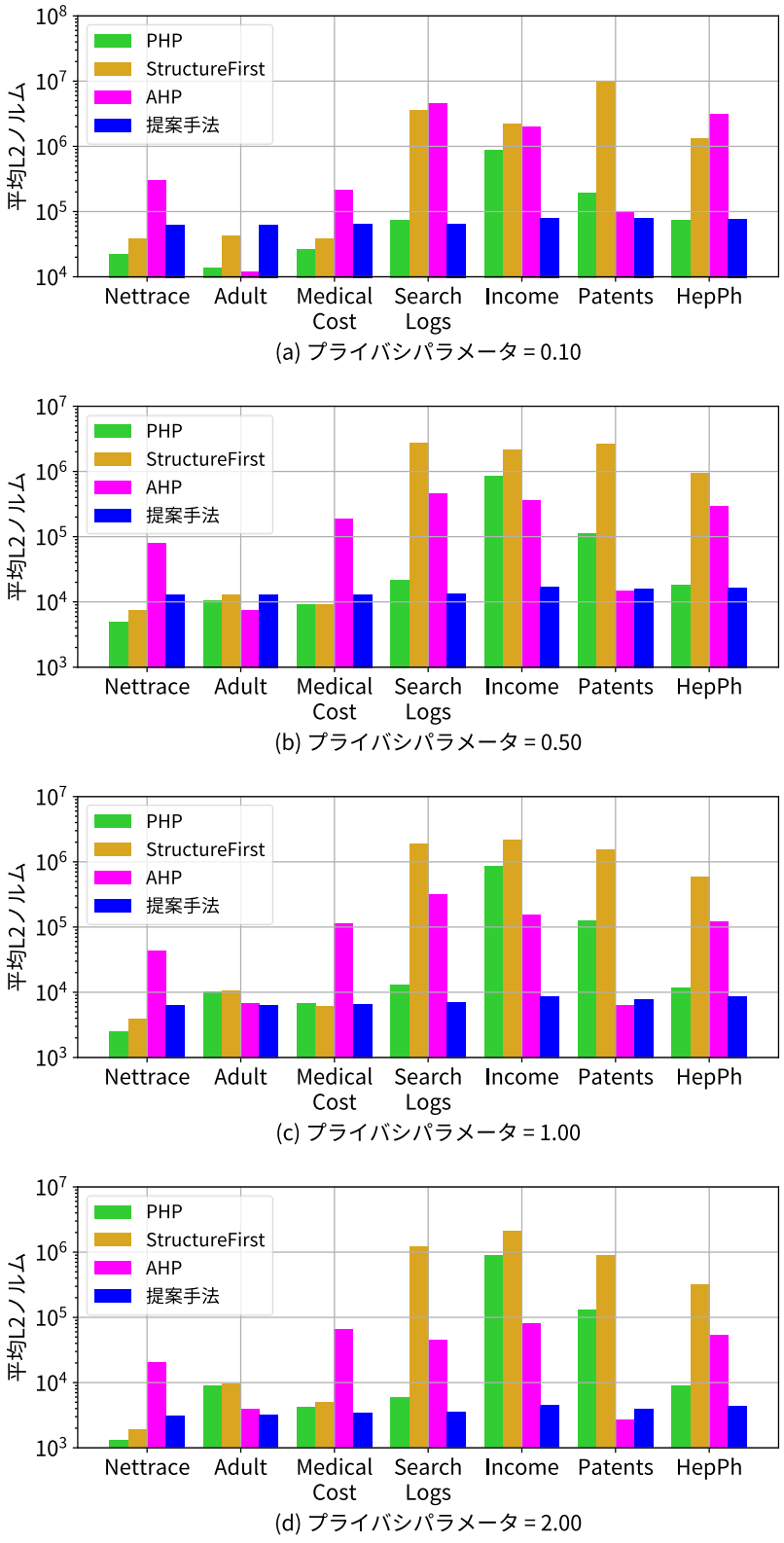
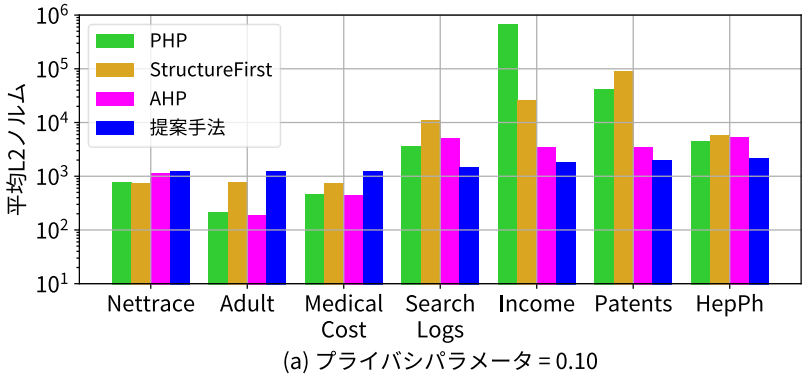
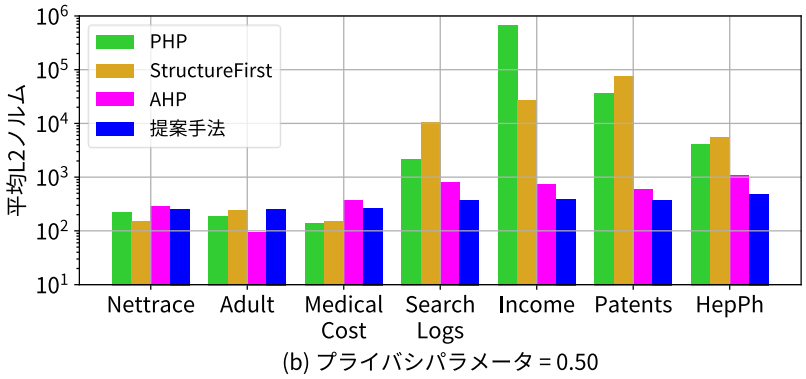


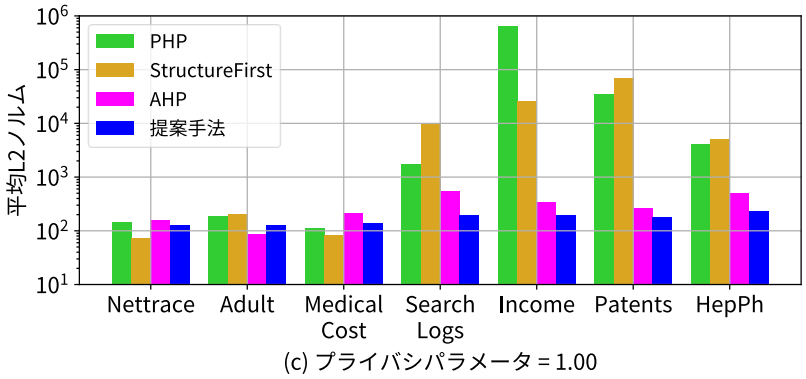
図 5.5　PHP，StructureFirst，AHP，提案手法の精度比較結果
（プレフィックスワークロード）



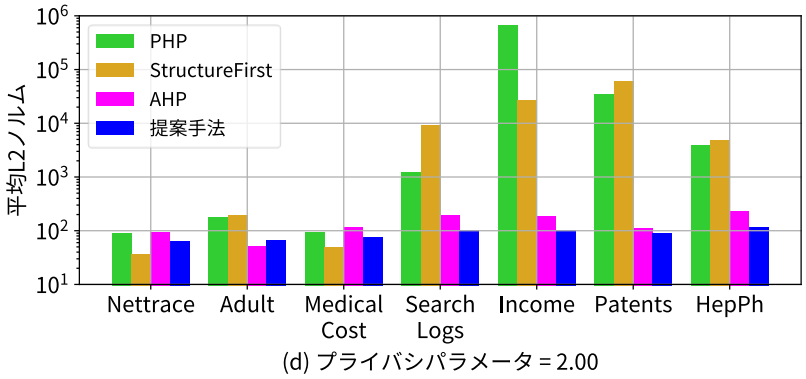
(a) プライバシパラメータ = 0.10



(b) プライバシパラメータ = 0.50



(c) プライバシパラメータ = 1.00



(d) プライバシパラメータ = 2.00

図 5.6　PHP，StructureFirst，AHP，提案手法の精度比較結果
(単一ワークロード)

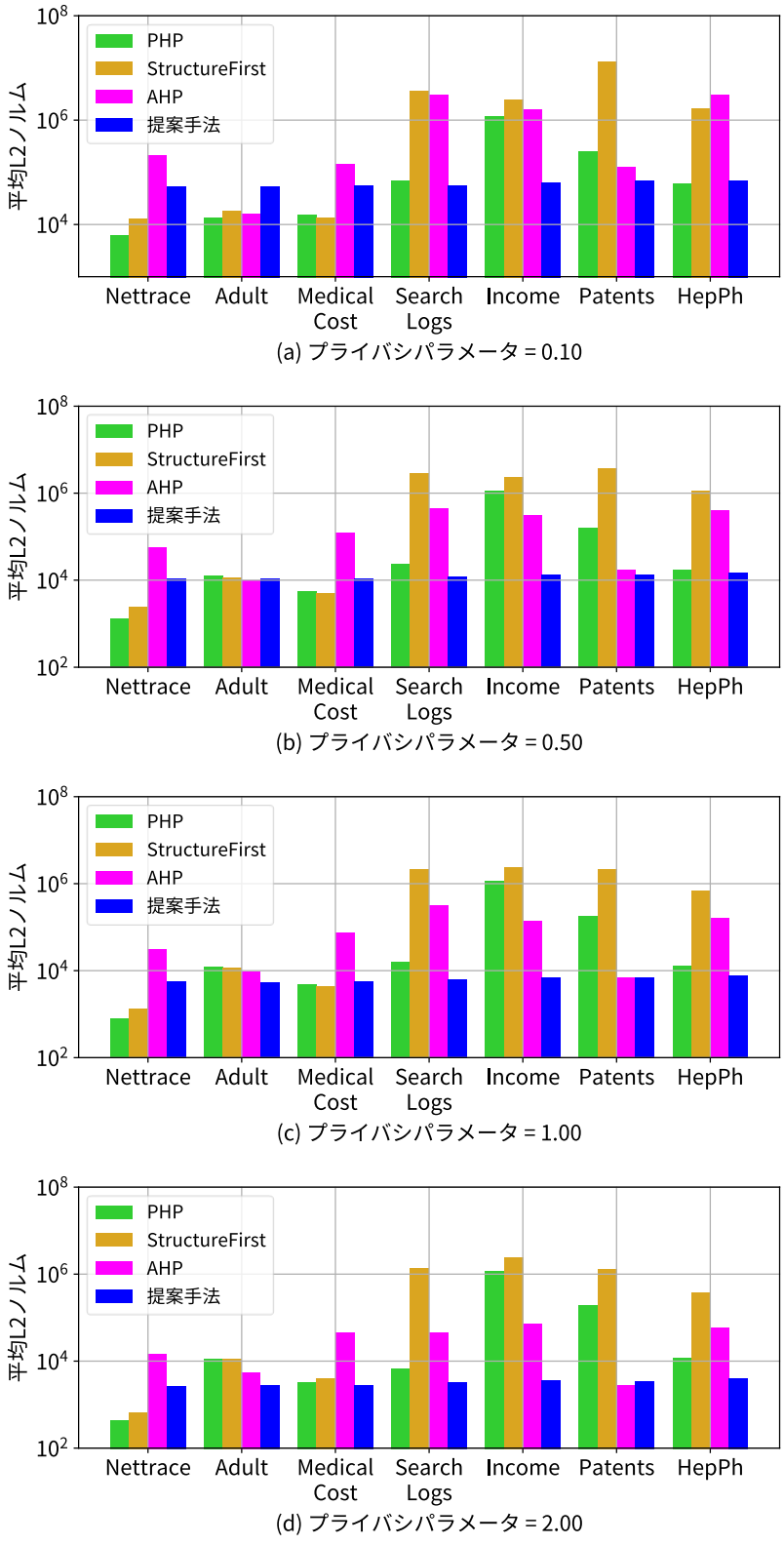


図 5.7　PHP，StructureFirst，AHP，提案手法の精度比較結果
（一様ランダムワークロード）

表 5.4　PHP，AHP，提案手法のバケット数

$\epsilon = 0.10$	Nettrace	Adult	Medical Cost	Search Logs	Income	Patents	HepPh
DAWA (Non-DP)	33	32	27	682	1,614	1,891	2,426
PHP	53	8	30	1,083	622	987	945
AHP	11	2	2	42	344	549	16
提案手法	1,740	1,741	1,749	1,927	2,527	2,700	2,568

ϵ=0.50	Nettrace	Adult	Medical Cost	Search Logs	Income	Patents	HepPh
DAWA (Non-DP)	47	96	185	1,581	2,027	1,908	3,115
PHP	68	9	115	1,096	620	1,063	1,023
AHP	35	4	6	155	536	1,336	112
提案手法	1,750	1,762	1,814	2,276	2,764	1,805	3.198

第6章　おわりに

本研究では，暗号文上でのデータ依存パーティショニングを高速化することを目的に，準同型暗号に適したプライバシ保護パーティショニングアルゴリズムを提案した．提案手法は，一次元ヒストグラムに対するレンジクエリ処理を対象にしている．提案手法は，差分プライバシと完全準同型暗号を組み合わせることで，複数のデータ所有者から収集した生データを，クラウドサーバ及びデータ解析者から保護する．既存手法[8]の計算量は入力ヒストグラムの大きさに対して指数増加するのに対して，提案手法では計算量を線形増加に低減した．実行時間の評価実験では，ドメインサイズが 4,096 であるヒストグラムの処理に，約 4 時間 35 分かかることが示された．提案手法の実行はクエリ受信前に一度だけ行われることを考えると，提案手法の実行時間は実用的であると言える．また，精度の評価実験から，提案手法の精度は DAWA[25]のデータ依存パーティショニング手法と同等であり，いくつかの代表的な競合手法[17, 18, 19]に対して優位な精度を達成することを確認した．

謝辞

本研究の一部は，国立情報学研究所 CRIS 共同研究（2020~2022）の助成を受けています.

本研究及び執筆作業を行うにあたり，多くのご指導を頂いた山名早人教授に厚く御礼申し上げます．また，本研究を行うにあたり，多くの助言を頂きました髙橋翼氏（現 LINE 株式会社）に心より感謝いたします．さらに，本研究及び執筆作業を行うにあたり，多くの助言を頂きました工藤雅士先輩，鈴木拓也先輩に深く感謝いたします．最後に，本研究を行うにあたり，多くの意見を頂きました山名研究室の皆様に感謝いたします.

参考文献

[1] Komawar, S., Batwal, M., Shah, S., Shahani, S. and Abraham, J.: Privacy Preserving Data Aggregation on Secure Cloud, Proceedings of the 4th International Conference on Computing, Communication Control and Automation (2018).

[2] Raisaro, J. L., Choi, G., Pradervand, S., Colsenet, R., Jacquemont, N., Rosat, N., Mooser, V. and Hubaux, J.-P.: Protecting Privacy and Security of Genomic Data in i2b2 with Homomorphic Encryption and Differential Privacy, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 15, pp.1413–1426 (2018)

[3] Raisaro, J. L., Troncoso-Pastoriza, J. R., Misbach, M., Sousa, J. S., Pradervand, S., Missiaglia, E., Michielin, O., Ford, B. and Hubaux, J.-P.: MEDCO: Enabling secure and privacy-preserving exploration of distributed clinical and genomic data, IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, pp.1328–1341 (2019).

[4] Froelicher, D., Misbach, M., Troncoso-Pastoriza, J. R., Raisaro, J. L. and Hubaux, J.-P.: MeDCO2: Privacy-preserving cohort exploration and analysis, Proceedings of the 30th Medical Informatics Europe Conference, Studies in Health Technology and Informatics, vol. 270, pp.317–321 (2020).

[5] Froelicher, D., Troncoso-Pastoriza, J. R., Sousa, J. S. and Hubaux, J.-P.: Drynx: Decentralized, Secure, Verifiable System for Statistical Queries and Machine Learning on Distributed Datasets, IEEE Transactions on Information Forensics and Security, vol. 15, pp.3035–3050 (2020).

[6] Roth, E., Newatia, K., Ma, Y., Zhong, K., Angel, S. and Haeberlen, A.: Mycelium: Large-Scale Distributed Graph Queries with Differential Privacy, Proceedings of the 28th ACM Symposium on Operating Systems Principles, pp.327–343 (2021).

[7] Chowdhury, A. R., Wang, C., He, X., MacHanavajjhala, A. and Jha, S.: Cryptε: Crypto-Assisted Differential Privacy on Untrusted Servers, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.603–619 (2020).

[8] Ushiyama, S., Takahashi, T., Kudo, M. and Yamana, H.: Construction of Differentially Private Summaries Over Fully Homomorphic Encryption, Proceedings of the 32nd International Conference on Database and Expert Systems Applications, vol. 12924 LNCS, pp.9–21 (2021).

[9] Zhang, N., Li, M. and Lou, W.: Distributed data mining with differential privacy, Proceedings of the IEEE International Conference on Communications (2011).

[10] Eigner, F., Kate, A., Maffei, M., Pampaloni, F. and Pryvalov, I.: Differentially

private data aggregation with optimal utility, Proceedings of the 30th Annual Computer Security Applications Conference, ACM International Conference Proceeding Series, vol. 2014-December, pp.316–325 (2014).

[11] Cheu, A., Smith, A., Ullman, J., Zeber, D. and Zhilyaev, M.: Distributed differential privacy via shuffling, Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, vol. 11476 LNCS, pp.375–403 (2019).

[12] Gu, X., Li, M., Cao, Y. and Xiong, L.: Supporting Both Range Queries and Frequency Estimation with Local Differential Privacy, Proceedings of the 2019 IEEE Conference on Communications and Network Security, pp.124–132 (2019).

[13] Cormode, G., Kulkarni, T. and Srivastava, D.: Answering range queries under local differential privacy, Proceedings of the 45th International Conference on Very Large Data Bases, vol. 12, pp.1126–1138 (2019).

[14] Dwork, C., McSherry, F., Nissim, K. and Smith, A.: Calibrating noise to sensitivity in private data analysis, Proceedings of the 3rd Theory of Cryptography Conference, vol. 3876 LNCS, pp.265–284 (2006).

[15] Dwork, C.: Differential privacy, Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, vol. 4052 LNCS, pp.1–12 (2006).

[16] Qardaji, W., Yang, W. and Li, N.: Differentially private grids for geospatial data, Proceedings of the 29th International Conference on Data Engineering, pp.757–768 (2013).

[17] Acs, G., Castelluccia, C. and Chen, R.: Differentially Private Histogram Publishing through Lossy Compression, Proceedings of the 12th IEEE International Conference on Data Mining, pp.1–10 (2012).

[18] Xu, J., Zhang, Z., Xiao, X., Yang, Y. and Yu, G.: Differentially Private Histogram Publication, Proceedings of the 28th IEEE International Conference on Data Engineering, pp.32–43 (2012).

[19] Zhang, X., Ghent, R., Xu, J., Meng, X. and Xie, Y.: Towards accurate Histogram publication under differential privacy, Proceedings of the 4th SIAM International Conference on Data Mining, vol. 2, pp.587–595 (2014).

[20] Xiao, Y., Xiong, L., Fan, L., Goryczka, S. and Li, H.: DPCube: Differentially private histogram release through multidimensional partitioning, Transactions on Data Privacy, vol. 7, pp.195–222 (2014).

[21] Zhang, J., Xiao, X. and Xie, X.: PrivTree: A differentially private algorithm for hierarchical decompositions, Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. 26, pp.155–170 (2016).

[22] Kotsogiannis, I., Hay, M., Machanavajjhala, A. and Miklau, G.: Pythia: Data dependent differentially private algorithm selection, Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. F127746, pp.1323–1337 (2017).

[23] Kato, F., Takahashi, T., Takagi, S., Cao, Y., Liew, S. P. and Yoshikawa, M.: HDPview: Differenitally private Materialized View for Exploring High Dimensional Relational Data, Proceedings of the 48th International Conference on Very Large Data Bases, vol.15, pp.1766–1778 (2022).

[24] Li, H., Cui, J., Meng, X. and Ma, J.: IHP: improving the utility in differential private histogram publication, Distributed and Parallel Databases, vol. 37, pp.721–750 (2019).

[25] Li, C., Hay, M., Miklau, G. and Wang, Y.: A data- and workload-aware algorithm for range queries under differential privacy, Proceedings of the 40th International Conference on Very Large Data Bases, vol. 7, pp.341–352 (2014).

[26] Kotsogiannis, I., Tao, Y., He, X., Fanaeepour, M., Machanavajjhala, A., Hay, M. and Miklau, G.: PrivateSQL: A differentially private SQL query engine, Proceedings of the 45th International Conference on Very Large Data Bases, vol. 12, pp.1371–1384 (2018).

[27] Li, C., Miklau, G., Hay, M., McGregor, A. and Rastogi, V.: The matrix mechanism: optimizing linear counting queries under differential privacy, VLDB Journal, vol. 24, pp.757–781 (2015).

[28] McKenna, R., Miklau, G., Hay, M. and Machanavajjhala, A.: Optimizing error of high-dimensional statistical queries under differential privacy, Proceedings of the 44th International Conference on Very Large Data Bases, vol. 11, pp.1206–1219 (2018).

[29] Johnson, N., Near, J. P. and Song, D.: Towards practical differential privacy for SQL queries, Proceedings of the 44th International Conference on Very Large Data Bases, vol. 11, pp.526– 539 (2018).

[30] Ge, C., Ilyas, I. F., He, X. and Machanavajjhala, A.: APEX: Accuracy-aware differentially private data exploration, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.177–194 (2019).

[31] Rogers, R., Subramaniam, S., Peng, S., Durfee, D., Lee, S., Kancha, S. K., Sahay, S. and Ahammad, P.: LINKEDIN'S AUDIENCE ENGAGEMENTS API: A PRIVACY PRESERVING DATA ANALYTICS SYSTEM AT SCALE, Journal of Privacy and Confidentiality, vol. 11 (2021).

[32] Matsuoka, K., Banno, R., Matsumoto, N., Sato, T. and Bian, S.: Virtual secure

platform: A five-stage pipeline processor over TFHE, Proceedings of the 30th USENIX Security Symposium, pp.4007–4024 (2021).

[33] Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices, Proceedings of the Annual ACM Symposium on Theory of Computing, pp.169–178 (2009).

[34] Chillotti, I., Gama, N., Georgieva, M. and Izabach`ene, M.: Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds, Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security, vol. 10031 LNCS, pp.3–33 (2016).

[35] Ushiyama, S., Takahashi, T., Kudo, M. and Yamana, H.: Homomorphic Encryption-Friendly Privacy-Preserving Partitioning Algorithm for Differential Privacy, Proceedings of the 2022 IEEE International Conference on Big Data (2022).

[36] McSherry F. and Mironov, I.: Differentially private recommender systems: Building privacy into the net-flix prize contenders, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.627–635 (2009).

[37] Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y. and Zhang, D.: Principled evaluation of differentially private algorithms using DPBENCH, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.139–154 (2016).

[38] Barbosa, M., Catalano, D. and Fiore, D.: Labeled homomorphic encryption: Scalable and privacy-preserving processing of outsourced data, Proceedings of the 22nd European Symposium on Research in Computer Security, vol. 10492 LNCS, pp.146–166 (2017).

[39] Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N. and Boneh, D.: Privacy-preserving matrix factorization, Proceedings of the ACM Conference on Computer and Communications Security, pp.801–812 (2013).

[40] Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D. and Taft, N.: Privacy-preserving ridge regression on hundreds of millions of records, Proceedings of the 34th IEEE Symposium on Security and Privacy, pp.334–348 (2013).

[41] Kim, S., Kim, J., Koo, D., Kim, Y., Yoon, H. and Shin, J.: Efficient privacy-preserving matrix factorization via fully homomorphic encryption, Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, pp.617–628 (2016).

[42] Giacomelli, I., Jha, S., Joye, M., Page, C. D. and Yoon, K.: Privacy-preserving ridge regression with only linearly-homomorphic encryption, Proceedings of the 16th International Conference on Applied Cryptography and Network Security, vol.

10892 LNCS, pp.243–261 (2018).

[43] Hay, M., Rastogi, V., Miklau, G. and Suciu, D.: Boosting the accuracy of differentially private histograms through consistency, Proceedings of the VLDB Endowment, vol. 3, pp.1021–1032 (2010).

[44] Bache K. and Lichman, M.: UCI machine learning repository, https://archive.ics.uci.edu/ml/index.php (2013).

[45] United States Department of Health and Human Services, Centers for Disease Control, and Prevention.: "National Center for Health Statistics." National home and hospice care survey, https://www.cdc.gov/nchs/nhhcs/index.html (2007).

[46] Ruggles, S., Alexander, J., Genadek, K., Goeken, R., Schroeder, M. and Sobek, M.: Integrated public use microdata series: Version 5.0, https://www.ipums.org/ (2010).

[47] Jure L. [n.d.]: Stanford Network Analysis Project, Stanford University, http://snap.stanford.edu/.

研究業績

[1] 牛山翔二郎，工藤雅士，髙橋翼，井上紘太朗，鈴木拓也，山名早人：差分プライバシと準同型暗号の組み合わせに関する研究動向調査，コンピュータセキュリティシンポジウム 2020，pp.207–214 (2020).

[2] 牛山翔二郎，髙橋翼，工藤雅士，井上紘太朗，鈴木拓也，山名早人：完全準同型暗号下での差分プライバシを保証した問い合わせ応答システムの検討，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，no.G25-4，pp.1–8 (2021).

[3] 工藤雅士，髙橋翼，牛山翔二郎，山名早人：パッシブ認証の精度向上を目指した模倣データ自動生成 ―スマートフォンを対象として―，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，no.F14-2，pp.1–8 (2021).

[4] Ushiyama, S., Takahashi, T., Kudo, M. and Yamana, H.: Construction of Differentially Private Summaries Over Fully Homomorphic Encryption, Proceedings of the 32nd International Conference on Database and Expert Systems Applications, vol. 12924 LNCS, pp.9–21, https://doi.org/10.1007/978-3-030-86475-0_2 (2021).

[5] 工藤雅士，髙橋翼，牛山翔二郎，山名早人：スマートフォンにおける耐模倣性向上を目指したパッシブ認証学習手法の提案，第 14 回データ工学と情報マネジメントに関するフォーラム，no.K43-1，pp.1–8 (2022).

[6] Ushiyama, S., Takahashi, T., Kudo, M. and Yamana, H.: Homomorphic Encryption-Friendly Privacy-Preserving Partitioning Algorithm for Differential Privacy, Proceedings of the 2022 IEEE International Conference on Big Data, pp.5802–5812 (2022).

[7] 工藤雅士，髙橋翼，牛山翔二郎，山名早人：タッチベース認証における公正な誤認証率評価フレームワークの提案，第 14 回データ工学と情報マネジメントに関するフォーラム (2023).（発表予定）

[8] Kudo, M., Takahashi, T., Yamana, H. and Ushiyama, S.: Fair and Robust Metric for Evaluating Touch-based Continuous Mobile Device, The 28th Annual Conference on Intelligent User Interface (2023).（投稿中）

[9] 牛山翔二郎，髙橋翼，工藤雅士，山名早人：完全準同型暗号上での差分プライベートパーティショニングアルゴリズムの高速化，IPSJ TOD 98 号 (2023).（投稿中）

付録

図 5.2 から図 5.7 で示される 6 つの差分プライバシアルゴリズム（Identity, DP-summary, PHP, StructureFirst, AHP, 提案手法）の精度の実験結果の実測値を表 A-1.1 から表 A-6.3 にそれぞれ示す．実験条件や評価指標の説明は 5.1 節及び 5.3 節に記載されている．

表 A-1.1　Identity の精度評価実験の測定値（本文中の図 5.2 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	70,720.4	14,474.8	7,241.78	3,601.53
Adult	70,720.4	14,474.8	7,241.78	3,601.53
Medical Cost	70,720.4	14,474.8	7,241.78	3,601.53
Search Logs	70,720.4	14,474.8	7,241.78	3,601.53
Income	70,720.4	14,474.8	7,241.78	3,601.53
Patents	70,720.4	14,474.8	7,241.78	3,601.53
HepPh	70,720.4	14,474.8	7,241.78	3,601.53

表 A-1.2　Identity の精度評価実験の測定値（本文中の図 5.3 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	1,810.77	362.248	181.078	90.524
Adult	1,810.77	362.248	181.078	90.524
Medical Cost	1,810.77	362.248	181.078	90.524
Search Logs	1,810.77	362.248	181.078	90.524
Income	1,810.77	362.248	181.078	90.524
Patents	1,810.77	362.248	181.078	90.524
HepPh	1,810.77	362.248	181.078	90.524

表 A-1.3　Identity の精度評価実験の測定値（本文中の図 5.4 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	60,464.0	12,358.0	6,158.82	3,098.94
Adult	60,464.0	12,358.0	6,158.82	3,098.94
Medical Cost	60,464.0	12,358.0	6,158.82	3,098.94
Search Logs	60,464.0	12,358.0	6,158.82	3,098.94
Income	60,464.0	12,358.0	6,158.82	3,098.94
Patents	60,464.0	12,358.0	6,158.82	3,098.94
HepPh	60,464.0	12,358.0	6,158.82	3,098.94

表 A-2.1　DP-summary の精度評価実験の測定値（本文中の図 5.2 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	65,463.7	13,433.1	6,385.18	3,339.49
Adult	65,606.5	13,335.2	6,574.38	3,233.55
Medical Cost	66,033.4	12,819.5	6,972.30	3,468.06
Search Logs	66,927.9	13,717.0	7,038.37	3,555.62
Income	80,494.5	16,470.1	8,601.31	4,411.50
Patents	77,885.1	15,913.8	8,102.90	4,055.90
HepPh	73,115.3	15,988.6	8,457.35	4,283.25

表 A-2.2　DP-summary の精度評価実験の測定値（本文中の図 5.3 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	1,359.33	274.245	138.459	69.5494
Adult	1,357.57	276.806	140.526	71.5476
Medical Cost	1,357.72	281.983	148.115	80.4876
Search Logs	1,525.40	403.654	228.151	120.802
Income	2,044.86	432.705	217.315	111.135
Patents	2,050.90	379.063	187.047	93.0131
HepPh	2,365.55	563.789	275.492	133.374

表 A-2.3　DP-summary の精度評価実験の測定値（本文中の図 5.4 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	56,510.1	11,158.4	5,500.18	2,844.76
Adult	55,415.3	11,094.2	5,693.45	2,794.91
Medical Cost	56,255.7	11,015.9	5,799.81	2,844.55
Search Logs	57,842.0	12,151.1	6,301.73	3,226.21
Income	63,419.9	13,046.4	6,724.92	3,466.81
Patents	67,052.1	13,686.6	6,949.50	3,469.31
HepPh	63,890.1	14,396.6	7,630.58	3,881.17

表 A-3.1　PHP の精度評価実験の測定値（本文中の図 5.5 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	21,889.1	4,924.18	2,520.40	1,317.38
Adult	13,630.4	10,500.8	10,298.6	9,178.05
Medical Cost	26,338.6	9,190.85	6,738.95	4,219.47
Search Logs	73,175.6	21,527.1	13,056.0	5,976.55
Income	869,665	852,474	864,016	889,314
Patents	189,850	110,883	124,381	132,533
HepPh	74,217.7	18,126.8	11,553.3	9,074.65

表 A-3.2　PHP の精度評価実験の測定値（本文中の図 5.6 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	775.132	223.756	146.050	90.0892
Adult	212.518	191.115	190.175	182.594
Medical Cost	464.649	141.254	110.817	96.7182
Search Logs	3,571.61	2,212.23	1,716.17	1,233.05
Income	687,070	662,800	660,983	663,122
Patents	41,680.1	35,802.8	35,213.0	35,598.9
HepPh	4,472.73	4,153.92	4,063.53	4,005.20

表 A-3.3　PHP の精度評価実験の測定値（本文中の図 5.7 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	6,099.44	1,345.30	779.103	425.409
Adult	13,427.5	12,681.2	12,444.7	10,974.9
Medical Cost	15,374.2	5,617.26	4,864.41	3,306.85
Search Logs	68,032.8	23,958.4	15,457.6	6,806.11
Income	1,177,420	1,156,100	1,172,000	1,203,640
Patents	251,762	156,350	179,445	194,471
HepPh	61,136.8	17,602.9	12,979.8	11,654.6

表 A-4.1　StructureFirst の精度評価実験の測定値（本文中の図 5.5 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	38,230.1	7,401.97	3,888.54	1,914.79
Adult	42,182.0	12,765.3	10,525.3	9,686.32
Medical Cost	37,923.0	9,176.97	6,194.72	5,097.15
Search Logs	3,542,600	2,749,200	1,920,710	1,247,180
Income	2,227,560	2,172,560	2,178,830	2,153,480
Patents	9,693,200	2,690,390	1,537,380	914,009
HepPh	1,313,260	932,004	594,694	322,816

表 A-4.2　StructureFirst の精度評価実験の測定値（本文中の図 5.6 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	745.400	149.069	74.6413	37.3665
Adult	768.752	241.525	204.084	193.558
Medical Cost	744.597	153.039	82.1283	50.2151
Search Logs	11,004.0	10,646.4	10,149.3	9,205.31
Income	26,615.9	26,602.7	26,607.3	26,593.2
Patents	91,043.9	74,656.5	68,397.1	61,371.9
HepPh	5,806.78	5,466.83	5,177.91	4,978.23

表 A-4.3　StructureFirst の精度評価実験の測定値（本文中の図 5.7 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	12,977.1	2,488.96	1,297.44	653.450
Adult	18,036.4	11,736.6	11,370.3	11,136.0
Medical Cost	13,543.3	5,007.44	4,376.33	4,079.58
Search Logs	3,608,470	2,883,390	2,083,590	1,377,090
Income	2,427,150	2,376,980	2,400,510	2,371,250
Patents	13,234,500	3,757,740	2,148,790	1,275,310
HepPh	1,632,210	1,123,920	691,550	379,089

表 A-5.1　AHP の精度評価実験の測定値（本文中の図 5.5 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	304,170	79,838.9	42,602.2	20,283.0
Adult	11,902.8	7,437.26	6,884.11	3,989.52
Medical Cost	213,668	188,956	112,498	67,345.0
Search Logs	4,575,900	461,407	314,080	44,865.3
Income	1,976,930	357,099	155,362	81,063.5
Patents	100,043	14,802.9	6,256.42	2,696.11
HepPh	3,074,820	291,120	119,552	53,948.2

表 A-5.2　AHP の精度評価実験の測定値（本文中の図 5.6 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	1,143.46	283.085	157.759	94.7831
Adult	191.527	96.3925	85.5664	52.6426
Medical Cost	442.218	374.889	217.773	118.846
Search Logs	5,109.71	818.284	558.915	196.007
Income	3,424.72	749.444	348.776	187.609
Patents	3,437.17	593.939	261.791	110.661
HepPh	5,430.31	1,067.49	511.036	236.201

表 A-5.3　AHP の精度評価実験の測定値（本文中の図 5.7 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	214,276	57,191.0	30,824.8	14,783.8
Adult	16,231.7	10,027.1	9,279.36	5,372.69
Medical Cost	142,600	125,947	74,417.3	44,409.9
Search Logs	3,086,590	451,687	314,934	45,476.2
Income	1,557,770	306,956	139,016	73,877.4
Patents	125,155	17,571.3	6,873.68	2,850.67
HepPh	2,998,290	413,507	162,263	58,774.7

表 A-6.1　提案手法の精度評価実験の測定値（本文中の図 5.2 及び図 5.5 に対応）

（プレフィックスワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	62,447.1	12,697.8	6,340.75	3,123.86
Adult	62,216.9	12,797.4	6,357.80	3,214.78
Medical Cost	63,682.5	12,903.2	6,570.74	3,437.08
Search Logs	64,538.5	13,231.7	7,072.96	3,559.64
Income	79,510.3	16,872.3	8,563.90	4,480.61
Patents	79,234.3	15,900.6	7,844.49	3,942.00
HepPh	75,807.8	16,548.0	8,632.66	4,417.85

表 A-6.2　提案手法の精度評価実験の測定値（本文中の図 5.3 及び図 5.6 に対応）

（単一ワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	1,253.82	254.252	128.181	63.8337
Adult	1,255.26	256.141	129.973	65.7920
Medical Cost	1,257.97	264.880	139.605	75.5381
Search Logs	1,458.22	366.796	197.650	102.244
Income	1,847.61	392.495	200.435	102.586
Patents	2,006.61	369.007	181.599	90.3529
HepPh	2,138.59	473.125	236.136	117.357

表 A-6.3　提案手法の精度評価実験の測定値（本文中の図 5.4 及び図 5.7 に対応）

（一様ランダムワークロード）

	$\epsilon = 0.10$	$\epsilon = 0.50$	$\epsilon = 1.00$	$\epsilon = 2.00$
Netrrace	53,557.4	10,688.5	5,465.64	2,658.85
Adult	53,321.6	10,943.1	5,424.02	2,769.62
Medical Cost	54,773.4	10,815.6	5,479.14	2,794.91
Search Logs	55,509.0	12,048.8	6,345.75	3,296.42
Income	61,990.5	13,078.7	6,782.67	3,559.12
Patents	67,971.5	13,745.3	6,799.59	3,420.19
HepPh	68,546.9	15,084.8	7,794.30	3,931.97