Computer Security Symposium 2020
26 - 29 October 2020

一般化逐次ベイズ法を用いた局所差分プライベートな度数分布推定

長谷川 聡[1,a)]　三浦 尭之[1]

概要：我々は逐次ベイズ法を一般化することで, 遷移確率で表現可能な局所差分プライベシー基準を満たす摂動法に適用可能な, 度数分布推定フレームワークを提案する. 我々のフレームワークは非負制約および総数制約を満たすことから, 従来手法と比べて推定誤差が小さくなることが期待できる. RAPPOR に対して本フレームワークを適用する. その際, 計算機実装上の問題で扱えるデータサイズに制限が生じるが, それを 10 倍以上改善する方法を提案する. 数値実験により既存手法と比較し度数推定の二乗誤差が小さいことを確認した. 特にレコード数が多い場合二乗誤差において約 20 倍の改善が見込めたことを報告する.

キーワード：局所差分プライバシ, 逐次ベイズ法

Locally Differentially Private Frequency Estimation with Generalized Iterative Bayesian Technique

Satoshi Hasegawa[1,a)]　Takayuki Miura[1]

Abstract: We propose a method for estimating the frequency distribution that satisfies the local differential privacy. Our method satisfies both non-negative and total number constraints to reduce the estimation error compared to conventional methods. Numerical experiments show that our method has the small error in frequency estimation compared to the existing methods.

Keywords: Local Differential Privacy, Iterative Bayesian Technique

1. はじめに

近年, コンピュータの性能や人工知能技術の飛躍的な向上により, 個人に紐づく情報 (パーソナルデータ) の収集・蓄積・活用に注目が集まっている. 個人はデータ収集者にデータを提供し, データ収集者は取得したデータを蓄積・分析を行うことで既存のサービスの品質向上や新たなサービスを創出する. 個人はサービスを利用することで, データ提供の恩恵を受けることができるため, データを提供することは個人およびデータ収集者両方にメリットがある. しかしながら, パーソナルデータは個人のプライバシに関する情報が含まれており, 安易に取り扱ってしまうと個人のプライバシを侵害してしまう恐れがある. よって個人のプライバシを保護しつつデータ利活用が可能なプライバシ保護データ開示技術に注目が集まっている.

プライバシ保護データ開示技術の多くの研究は, 個人がデータ収集者に対しデータを提供する際, データ収集者は信頼できるという仮定を置く. すなわち生のデータをデータ収集者に送信する. データ収集者は, 各個人から得たデータを第三者提供や目的外利用する際にプライバシ保護を行う. 例えば, k-匿名化 [12] や差分プライバシ [3] に基づくデータ開示技術がそれに該当する. しかしながら, クラッキングや人為的な運用ミス等によりプライバシ保護前のデータベースが流出した場合やそもそもデータ収集者が悪意を持つ場合は, 個人のプライバシを保護できない恐れがある.

本研究では, このような問題に対処するため, データ収集者が信頼できない場合でも個人のプライバシを保護可能な局所差分プライバシ [5], [8] に基づくデータ利活用に関して取り扱う. 局所差分プライバシとはデータ収集者に対し

[1] NTT セキュアプラットフォーム研究所
NTT Secure Platform Laboratories
[a)] satoshi.hasegawa.ks@hco.ntt.co.jp

© 2020 Information Processing Society of Japan

個人が送信したデータの識別不可能性を保証する技術であり, 具体的には個人がデータにランダム化等のプライバシ保護処理を施すことで実現する. 局所差分プライバシに基づくデータ利活用は, Apple[13] や Google[4], Microsoft[6] といった企業がすでに実用化している. 彼らは個人が送信するプライバシ保護されたデータを収集し, プライバシ保護される前のデータの出現頻度 (度数) を推定することを行っている. 本研究でも同様に局所差分プライバシに基づく度数推定の問題を取り扱う.

1.1 関連研究

Warner らは Randomized Response と呼ばれる 2 値で答える質問の回答に対し一定の確率で嘘をつくことで, 個人のプライバシを保護しつつ質問を回答可能な方式を提案している [16]. データ収集者は質問結果を集約する際に, 一定の確率で嘘をつかれていることを考慮し度数を推定する (逆行列法という方法 [2]. 詳しくは節 2.3.1 で述べる). これにより回答の度数をある程度正確に把握することができる.

Randomized Response は 2 値の回答であるが, 多値に拡張した方法として k-Randomized Response が提案されており, 局所差分プライバシを満たすことが示されている [7]. Randomized Response と同様に逆行列法を用いて度数を推定可能であるが, 性質上度数に負の値が生じる恐れがある. その問題を解決する方法として逐次ベイズ法と呼ばれる度数に負の値が生じないアルゴリズムが提案されている [2](節 2.3.1 で述べる). Kairouz らは低いプライバシ強度において, 後に示す k-RAPPOR と比較し, k-Randomized Response の推定誤差が小さく優れていることを示している (以降, k-Randomized Response を Randomized Response と呼ぶこととする).

多値の回答方法の別のアプローチとして, k-RAPPOR と呼ばれる手法が提案されており, 局所差分プライバシを満たすことが示されている [4], [7]. k-RAPPOR は多値の回答の際に, 値をそのままランダム化せず, データを高次元バイナリ空間に射影 (符号化) し, 射影したデータを各次元ごとにランダム化するアプローチをとる (以降, k-RAPPOR を RAPPOR と呼ぶこととする). RAPPOR では各次元がデータに対応しており, 各次元ごとに度数の推定を行うことでデータ全体の度数推定を行う. この方法では各次元ごとに独立して度数を推定することから, 度数の総数の制約 (総数制約) を満たさなくなってしまう恐れがある. また近年 RAPPOR に EM アルゴリズムを適用することで度数推定を行う方法も提案されており, 逆行列法に比べて推定誤差が小さいことが示されている [18]. Randomized Response と比較し, プライバシ強度が強い場合に推定誤差が小さいことが示されている [7].

RAPPOR と同様にデータを高次元に射影 (符号化) するが, RAPPOR と異なり出力範囲を限定 (高次元中の部分空間のみに) することで推定誤差を小さくする Subset Selection という局所差分プライバシを満たす方式が提案されている [14]. [1], [14] によれば, いずれのプライバシ強度においても既存の手法の中で最も推定誤差が小さい方式であることが示されている.

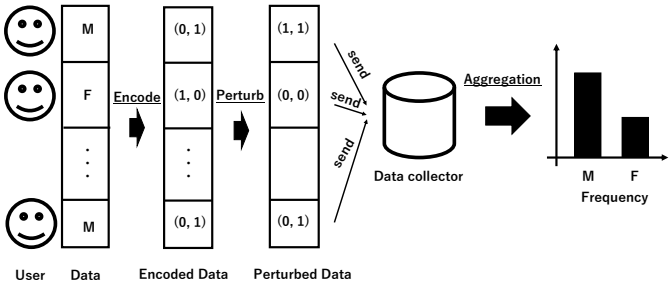

図 1 本研究における問題設定. 各ユーザは入力を符号化・摂動を行い, データ収集者に送信する. データ収集者は送信されたデータを元に度数を推定する.

RAPPOR や Subset Selection はデータを高次元に射影 (符号化) することから, 送信コストが大きくなってしまう. そこで, 送信コストが Randomized Response の定数倍程度でありつつ, 推定誤差も小さい方式として, Hadamard Response と呼ばれる局所差分プライバシを満たす手法が提案されている [1].

他にも, 利用シーン等に応じて局所差分プライバシの定義を拡張し推定誤差を小さく方法が提案されている [10], [11] が, 本研究では, 汎用性の観点から局所差分プライバシを満たす手法に限定して議論する.

1.2 貢献

本研究では既存の手法と同一のプライバシ強度を持ちつつ推定誤差の小さい度数推定方式を提供することを目標とする. 度数推定の方式として逐次ベイズ法に着目する. 逐次ベイズ法は非負制約および総数制約を満たすことから, これまで適用できていないランダム化方式にも適用できると推定誤差の改善が期待できる. 本研究の貢献は 3 つある.

(1) 逐次ベイズ法を一般化 (一般化逐次ベイズ法と呼ぶ) し, 計算量の削減方法を提案.
(2) RAPPOR への一般化逐次ベイズ法の適用および扱える入力サイズ (取りうるデータの種類数) の増加.
(3) 数値実験による既存の手法との推定誤差の比較を行い提案手法が優れていることを示した点.

逐次ベイズ法は Randomized Response のような入力と出力のサイズが同一の場合において提案されている方式であり, RAPPOR 等の入力を高次元化するような出力のサイズが異なってしまう方式には適用できるか非自明である. 本研究では一般化逐次ベイズ法が入力と出力のサイズが異なる状況下においても成立することを示す. また, 送信したユーザ数が出力サイズより小さいケースにおいて, 計算量を改善できるアルゴリズムを提案する.

RAPPOR の度数推定に対して適用できることを確認する (これは本質的に [18] と同様である). [18] らの EM アルゴリズムに比べ計算量の改善および扱える入力サイズを大

きくする方法を示す. [18] では入力サイズは数百程度であるが, 本提案手法では数千以上扱えるようにする.

既存の局所差分プライバシを満たす手法と比較し, 提案手法が優れていることを示す. 2 つの分布が異なる人工データを用意し, 推定誤差を評価した. 18 ケース中 16 ケースで提案手法が優れていることを確認した. 特に $N = 100,000$ のケースにおいては総じて提案手法が最も優れており, 既存の最も良い手法に比べ誤差が 20 倍程度改善できるケースがあることを確認した.

2. 準備

2.1 問題設定

本稿では小文字の bold 体は行ベクトルを表現し, 大文字の bold 体は行列を表す.

ユーザ数を N とする. 本稿では, 各ユーザがデータ収集者に対してデータを送信し, データ収集者は得たデータを集約し各データの出現頻度 (度数) を得ることを考える. その際, 各ユーザは D 種類の値を持つデータ集合 $\mathcal{X} = [D] = \{0, ..., D-1\}$ に含まれる自身のデータ $x \in \mathcal{X}$ を (確率的) メカニズム $M : \mathcal{X} \to \mathcal{Z}$ に入力し, 得た出力 $z \in \mathcal{Z}$ を, データ収集者に送信する (必ずしも $\mathcal{X} = \mathcal{Z}$ でなくともよい, また $|\mathcal{X}| = D$ などのデータ集合のサイズをデータサイズと呼ぶこととする). ユーザはこのメカニズム M を通すことで, データ収集者に対してプライバシ情報が漏れないようにする. データ収集者は全ユーザから得たデータ集合 $\{z_i\}_{i=0}^{N-1}$ をメカニズム M の影響を考慮しながら集約し, データ $x \in \mathcal{X}$ の度数が格納されている度数ベクトル $\mathbf{h}_{\mathcal{X}} = \{h_{\mathcal{X}}(0), ..., h_{\mathcal{X}}(D-1)\}$(ただし, $h_{\mathcal{X}}(0) \geq 0, ..., h_{\mathcal{X}}(D-1) \geq 0, \sum_{i=0}^{D-1} h_{\mathcal{X}}(i) = N$) を得る.

なお, 後の議論のために, Wang ら [15] の説明と同様に, メカニズム M を符号化, 摂動の 2 ステップに分割し, 以下の 3 ステップから成る処理であるとする (図 1).

符号化 ユーザ i は入力 $x_i \in \mathcal{X}$ を Encode : $\mathcal{X} \to \mathcal{Z}$ で符号化し, 出力 $y_i \in \mathcal{Z}$ を得る.

摂動 ユーザ i は Encode した出力 $y_i \in \mathcal{Z}$ を Perturb : $\mathcal{Z} \to \mathcal{Z}$ でプライバシ保護し, 得られた $z_i = \mathrm{Perturb}(y_i)$ をデータ収集者に送信する.

集約 データ収集者は全ユーザから得たデータ集合 $\{z_i\}_{i=1}^{N}$ を用いて, メカニズム M を通す前の度数ベクトル $\mathbf{h}_{\mathcal{X}}$ を推定する.

2.2 局所差分プライバシ

局所差分プライバシは, 直観的には「任意の 2 つのデータをメカニズム M に入力して得られた出力を見た際に, その出力がどちらの入力から得られたものかが区別がつきにくい」という基準であり, メカニズム M に対して定義される.

2.2.1 定義

確率的なメカニズム $M : \mathcal{X} \to \mathcal{Z}$ が $\epsilon-$ 局所差分プライバシを満たすとは, 任意の 2 入力 $x_a, x_b \in \mathcal{X}$ に対し,

$$\forall z \in \mathcal{Z} : \Pr[M(x_a) = z] \leq e^{\epsilon} \Pr[M(x_b) = z] \qquad (1)$$

が成り立つことをいう [15].

2.3 局所差分プライバシに基づくメカニズム

局所差分プライバシに基づくその具体的な実装として, Randomized Response[7], [16] および RAPPOR(Unary Encoding)[4] を示す.

2.3.1 Randomized Response

Randomized Response は, ある一定の確率で値を維持し, それ以外の確率で値をランダムに書き換える手法である.

符号化 Randomized Response では符号化は行わず, 入力と出力のデータの種類は同一, すわなち $\mathcal{X} = \mathcal{Z}$ であり, $x = \mathrm{Encode}(x)$ である.

摂動入力 $x \in \mathcal{X}$ が出力 $x' \in \mathcal{X}$ となる条件付き確率 (遷移確率と呼ぶ) を定義し, 遷移確率に基づきデータをランダムに変更する.

$$\Pr[x' = \mathrm{Perturb}(x)] = \Pr[x'|x] = \begin{cases} \frac{e^{\epsilon}}{e^{\epsilon}+d-1} \mathrm{if} x' = x \\ \frac{1}{e^{\epsilon}+d-1} \mathrm{if} x' \neq x \end{cases} \qquad (2)$$

全ての $x, x' \in \mathcal{X}$ について, 条件付き確率 $\Pr[x'|x]$ の値を並べた行列を遷移確率行列 $\mathbf{P}$ と呼ぶ. この方式が局所差分プライバシを満たすことは [7] を参照されたい.

集約 各ユーザから得られた値の集約方法として, 2 通り提案されている. 1 つは逆行列法 [2], [7], [15], [16], 1 つは逐次ベイズ法 [2] である.

逆行列法はランダム化による度数ベクトルの期待値の関係を利用する. 遷移確率行列 $\mathbf{P}$, データ収集者が各ユーザから得たランダム化済みデータ $\{x'_i\}_{i=0}^{N-1}$ から構成した度数ベクトルを $\mathbf{h}_{\mathcal{X}'} = \{h_{\mathcal{X}'}(0), ..., h_{\mathcal{X}'}(D-1)\}$ (ただし, $h_{\mathcal{X}'}(0) \geq 0, ..., h_{\mathcal{X}'}(D-1) \geq 0, \sum_{i=0}^{D-1} h_{\mathcal{X}'}(i) = N$), ランダム化前データ $\{x_i\}_{i=0}^{N-1}$ の度数ベクトルを $\mathbf{h}_{\mathcal{X}}$ とした場合, 遷移確率行列にしたがってランダム化されたデータの度数ベクトルの期待値 $\hat{\mathbf{h}}_{\mathcal{X}'}$ は,

$$\hat{\mathbf{h}}_{\mathcal{X}'} = \mathbf{h}_{\mathcal{X}} \mathbf{P}^T \qquad (3)$$

となる. この関係を用いて, ベクトル $\mathbf{h}_{\mathcal{X}'}$ と $\mathbf{P}$ からランダム化前データの度数ベクトル $\mathbf{h}_{\mathcal{X}}$ を推定する.

$$\mathbf{h}_{\mathcal{X}} = \mathbf{h}_{\mathcal{X}'} {\mathbf{P}^{-1}}^T \qquad (4)$$

逆行列法は線形方程式を解くことで解を求めることができるが, その性質上度数ベクトルの各値に負値が生じる恐れがある. 適宜負の値は 0 に補正するなどし用いる.

逆行列法と比較し, 得られた度数ベクトルの各値が負値とならない方式として, 逐次ベイズ法 [2] がある. 次式を繰り返し計算することで, 度数ベクトル $\mathbf{h}_{\mathcal{X}}$ を推定する.

$$\mathbf{h}_{\mathcal{X}}^{t+1} = \mathbf{h}_{\mathcal{X}}^{t} \cdot \left(\mathbf{P} \left(\frac{\mathbf{h}_{\mathcal{X}'}}{\mathbf{h}_{\mathcal{X}}^{t} \mathbf{P}} \right)^T \right)^T \qquad (5)$$

ここで $\mathbf{h}_{\mathcal{X}}^{t}$ は t 回目の繰り返しの $\mathbf{h}_{\mathcal{X}}$ を示す. また, $\cdot$ はベ

クトルの要素積, 割り算はベクトルの要素ごとの割り算を表す. 初期状態すなわち $t=0$ の場合は $\mathbf{h}^0_{\mathcal{X}} = \mathbf{h}_{\mathcal{X}'}$ とし, あらかじめ定めた $\eta > 0$ に対して $||\mathbf{h}^{t+1}_{\mathcal{X}} - \mathbf{h}^t_{\mathcal{X}}|| < \eta$ となるまで繰り返し計算することで, $\mathbf{h}_{\mathcal{X}}$ を得る. 節 3 で逐次ベイズ法を一般化するため, 詳細に関しては割愛する.

2.3.2 RAPPOR

RAPPOR(Basic One-time RAPPOR または Unary Encoding とも呼ばれる) は, $x \in \mathcal{X}$ を D 次元バイナルベクトル化し, ベクトルの各値ごとに摂動および集約を行う手法である.

符号化 Encode : $\mathcal{X} \to \{0,1\}^D$ である. ただし出力は one-hot 表記の D 次元バイナリベクトルである. すなわち, $\mathrm{Encode}(x) = (0,...,0,1,0,...,0) = \mathbf{b}$ である (i 番目の位置のみ 1 であり, それ以外が 0).

摂動 $\mathrm{Perturb}(\mathbf{b}) = \mathbf{b}'$ は, ベクトルの各値ごとに次式に示す確率に従うようにランダム化処理を行う. ベクトル $\mathbf{b}$ の i 番目の値を $\mathbf{b}[i]$ とした場合,

$$\Pr[\mathrm{Perturb}(\mathbf{b}[i]) = 1] = \begin{cases} p \text{ if } \mathbf{b}[i] = 1 \\ q \text{ if } \mathbf{b}[i] = 0 \end{cases} \tag{6}$$

となる. この方式が ϵ-局所差分プライバシを満たすことが示されており [7], [15], Symmetric Unary Encoding と呼ばれる $p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}, q = \frac{1}{e^{\epsilon/2}+1}$ とする方法や, Optimal Unary Encoding と呼ばれる $p = \frac{1}{2}, q = \frac{1}{e^{\epsilon}+1}$ とする方法が提案されている [7], [15].

集約 収集者はベクトルの各値の度数の推定を逆行列法もしくは逐次ベイズ法のいずれかを用いて行う. 例えば $x \in \mathcal{X}$ の度数を推定する場合, 収集した $\{\mathbf{b}'_i\}_{i=1}^N$ の x 番目の位置の 0, 1 の数をそれぞれカウントし, カウントした値をベクトル $\mathbf{v}$ として保持する. 式 (6) で示した遷移確率に基づく遷移確率行列を作成し, $\mathbf{v}$ を用いて逆行列法を適用すれば良い. この方法では次元ごとに独立して度数を推定することから総数制約を満たすことができない. よって総数制約を満たすよう適宜補正を行う.

3. 提案手法

本研究では, Randomized Response で用いられる逐次ベイズ法を一般化し, 遷移確率で表現可能なランダム化手法へ適用できるフレームワーク**一般化逐次ベイズ法**を提供する. フレームワークの適用例として RAPPOR への適用を行う ([18] と本質的に同様である).

計算機上で本手法の実装を試みた場合, 単純なフレームワークの適用では扱えるデータ集合 $\mathcal{X}$ のサイズ D に制限が生じる ([18] も同様). 一般化逐次ベイズ法を改良することで, 扱えるデータサイズを増やす方法を提案する.

3.1 逐次ベイズ法の一般化

逐次ベイズ法 [2] は, $\mathcal{X} = \mathcal{Z}$ を前提として議論している. 我々の知る限り $\mathcal{X} \neq \mathcal{Z}$ は議論されていない[*1]ため, $\mathcal{X} \neq \mathcal{Z}$ でも [2] らと同様の議論で逐次ベイズ法が成立することを確認する.

逐次ベイズ法はベイズの定理を応用した方式であり, 本質的には EM アルゴリズムと呼ばれる手法である. ベイズの定理を用いると $\Pr[x|z]$ は,

$$\begin{aligned} \Pr[x|z] &= \frac{\Pr[z|x]\Pr[x]}{\Pr[z]} \\ &= \frac{\Pr[z|x]\Pr[x]}{\sum_{r\in\mathcal{X}}\Pr[z|r]\Pr[r]} \end{aligned} \tag{7}$$

となる. 加えて,

$$\Pr[x] = \sum_{q\in\mathcal{Z}} \Pr[q]\Pr[x|q] \tag{8}$$

及び, $h_{\mathcal{X}}(x) = \Pr[x] \times N$ という関係を用いる. 次式を繰り返し計算することで, 度数 $h_{\mathcal{X}}(x)$ を推定する.

$$h^{t+1}_{\mathcal{X}}(x) = \sum_{q\in\mathcal{Z}} h_{\mathcal{Z}}(q) \frac{\Pr[q|x]h^t_{\mathcal{X}}(x)}{\sum_{r\in\mathcal{X}} Pr[q|r]h^t_{\mathcal{X}}(r)} \tag{9}$$

ここで $h_{\mathcal{X}}(x)^t$ は t 回目の繰り返しの $h_{\mathcal{X}}(x)$ の値を示す. 式 (9) は各値の度数の計算方法であるが, 度数ベクトルを一括して計算する場合は,

$$\mathbf{h}^{t+1}_{\mathcal{X}} = \mathbf{h}^t_{\mathcal{X}} \cdot \left(\mathbf{P} \left(\frac{\mathbf{h}_{\mathcal{Z}}}{\mathbf{h}^t_{\mathcal{X}}\mathbf{P}} \right)^T \right)^T \tag{10}$$

である. あらかじめ定めた $\eta > 0$ に対して $||\mathbf{h}^{t+1}_{\mathcal{X}} - \mathbf{h}^t_{\mathcal{X}}|| < \eta$, となるまで繰り返し計算することで, $\mathbf{h}_{\mathcal{X}}$ を得る.

逐次ベイズ法との違いは初期値 $\mathbf{h}^0_{\mathcal{X}}$ の設定方法の違いである. 逐次ベイズ法ではベクトルの次元数が同一であったことから, 初期状態すなわち $t=0$ の場合は $\mathbf{h}^0_{\mathcal{X}} = \mathbf{h}_{\mathcal{Z}}$ としていた. しかしながら, 次元数が異なるため同様のことはできない. $\mathbf{h}^0_{\mathcal{X}} = \{N/|\mathcal{X}|, ..., N/|\mathcal{X}|\}$ とすることで対応可能である[*2]

一般化逐次ベイズ法の主要な計算部分である式 (10) に要する空間計算量, 時間計算量は $O(|\mathcal{X}||\mathcal{Z}|)$ である. これを繰り返し実行することで解を得ることができる. また行列計算ですべて記述できてることから, BLAS ライブラリの利用や GPU 用の BLAS ライブラリ等が適用でき十分高速に計算が可能である.

3.2 疎性に基づく計算量の削減

素朴な方法では空間計算量, 時間計算量ともに $O(|\mathcal{X}||\mathcal{Z}|) = O(D|\mathcal{Z}|)$ である. $|\mathcal{Z}| > N$ の条件が成り立つ際に, 一般化逐次ベイズ法の計算の途中結果が 0 になることを利用することで, 空間計算量, 時間計算量ともに $O(DN)$ にまで削減する方法を提案する.

$|\mathcal{Z}| > N$ である場合の $\mathbf{h}_{\mathcal{Z}}$ の疎性に着目する. $\mathbf{h}_{\mathcal{Z}}$ のサ

[*1] [9] では, $\mathbf{P}$ の定義は $\mathbf{P} \in [0,1]^{|\mathcal{X}|\times|\mathcal{Z}|}$ としているが, Randomized Response を扱っており, 後のアルゴリズム等は $\mathcal{X} = \mathcal{Z}$ で議論している. 従って本稿では議論されていないものとして取り扱う.

[*2] 数値実験的にこの初期値で実験した結果性能が良いことを確認している. 最適性の保証等は紙面の都合上割愛させていただく.

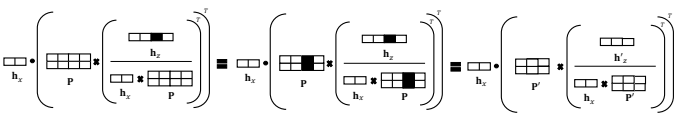

図 2 疎性を利用した計算量削減のイメージ. 黒塗箇所が 0 であるとする. 左, 中央, 右いずれも同値であることがわかる.

イズは $|\mathcal{Z}|$ であるが, $|\mathcal{Z}| > N$ であることから, $\mathbf{h}_\mathcal{Z}$ の値が一部必ず 0 となる (度数が 0 すなわち $\{z_i\}_{i=1}^N$ に出現しない. ユーザ数に対してデータサイズが大きいため).

式 (10) の計算の一部を $\mathbf{v} = \frac{\mathbf{h}_\mathcal{Z}}{\mathbf{h}_\mathcal{X}\mathbf{P}}$ とおく. すると, 式 (10) は,

$$\mathbf{h}_\mathcal{X}^{t+1} = \mathbf{h}_\mathcal{X}^t \cdot \left(\mathbf{P}\mathbf{v}^T\right)^T$$

となる. $\mathbf{v}$ の計算において, $\mathbf{h}_\mathcal{Z}$ のベクトルの値が 0 の箇所は, 得られる $\mathbf{v}$ の計算結果も必ず 0 になる. よって $\mathbf{P}$ の $\mathbf{h}_\mathcal{Z}$ のベクトルの値が 0 に対応する列は全て値が 0 でも良い. 加えて $\mathbf{h}_\mathcal{X} \cdot \mathbf{P}\mathbf{v}^T$ においても $\mathbf{v}$ の値が 0 に対応する $\mathbf{P}$ の列は全て値が 0 で良い. (図 2 参照)

これらより $\mathbf{P}$ は $\mathbf{h}_\mathcal{Z}$ が非ゼロの列のみを持つ行列 $\mathbf{P}'$ とすれば良く, $\mathbf{h}_\mathcal{Z}$ も非ゼロのみを持つベクトル $\mathbf{h}'_\mathcal{Z}$ を扱えば良い. 式 (10) は疎性を利用することで以下となる.

$$\mathbf{h}_\mathcal{X}^{t+1} = \mathbf{h}_\mathcal{X}^t \cdot \left(\mathbf{P}' \left(\frac{\mathbf{h}'_\mathcal{Z}}{\mathbf{h}_\mathcal{X}^t \mathbf{P}'}\right)^T\right)^T \tag{11}$$

3.3 一般化逐次ベイズ法の RAPPOR への適用

RAPPOR を一般化逐次ベイズ法に対応させる. 一般化逐次ベイズ法の入力は, $\mathbf{h}_\mathcal{Z}$ および $\mathbf{P}$ である. $\mathbf{h}_\mathcal{Z}$ は, ユーザ i が摂動した $\mathrm{Perturb}(\mathrm{Encode}(x_i)) = \mathbf{b}'_i$ を集約して得られた $\{\mathbf{b}'_i\}_{i=1}^N$ から, 各ベクトル $\mathbf{b}' \in \mathcal{Z}$ の度数 $h_\mathcal{Z}(\mathbf{b}')$ を求めて構成すれば良い.

$\mathbf{P}$ は行列の各要素である $\Pr[z|x]$ がわかればよい. より詳しく述べると, RAPPOR では Perturb の入力および出力がバイナリベクトルであるので, バイナリベクトルの遷移確率が求まれば良い. バイナリベクトルの遷移確率は, バイナリベクトルの各値の遷移確率の積となる. また, Perturb の入力の数は D 種類の値であるが, 出力は 2^D 種類となることから, 遷移確率行列 $\mathbf{P}$ は $D \times 2^D$ 行列となる[*3]. 例えば, $D = 3$ の場合, 入力は 3 種類であり, 出力は 8 種類となる. また, 入力が $\mathbf{b} = (0, 1, 0)$ であり出力 $\mathrm{Perturb}(\mathbf{b}) = (1, 1, 0)$ であった場合, 遷移確率は式 (6) より, $p(1-q)q$ である. $D = 3$ の場合の遷移確率行列 $\mathbf{P} \in [0, 1]^{D \times 2^D}$ を図 4(8 ページ目に掲載) に示す.

この $\mathbf{h}_\mathcal{Z}$ および $\mathbf{P}$ を用いて, 式 (10) を計算すれば良い. RAPPOR は $|\mathcal{Z}| = 2^D$ であるが, 往々にして $N < 2^D$ であることから疎性を利用した式 (11) を用いることで高速な計算が期待できる.

RAPPOR を一般化逐次ベイズ法に適用した際, D が大きくなるにつれて遷移確率行列の各値は小さくなる. よって, 計算機に実装する時の数値の精度が問題となる ([18] も同様[*4]). 例えば p, q は Symmetric Unary Encoding として RAPPOR により撹乱したとする. その際にバイナリベクトルの値が全く一致しない遷移確率は $q^{(D-1)}(1-p) = q^D = (\frac{1}{\exp(\epsilon/2)+1})^D$ となる. 例えば $\epsilon = 1$ かつ $D = 764$ および $\epsilon = 4$ の場合 $D = 350$ で遷移確率は $5E-324$ と非常に小さい値となり, それ以上小さい値は倍精度浮動小数でも扱えなくなる.

4 倍精度や多倍長精度浮動小数点演算を用いるというアプローチも考えられるが, BLAS ライブラリや GPU 演算を用いる場合, 大抵のサポートが倍精度浮動小数点までであり高速な計算が期待できなくなる. 実用上倍精度浮動小数点で扱えることが好ましいといえる.

3.4 データサイズ D の増加

$\mathcal{X}$ のサイズ D が大きい場合でも倍精度浮動小数点で扱えるようにする方法を提案する. $\mathbf{P}$ や $\mathbf{P}'$ を定数倍しても式 (10), 式 (11) の計算に影響がないことに着目することで実現する.

式 (11) (式 (10) も同様) は,

$$\begin{aligned}\mathbf{h}_\mathcal{X}^{t+1} &= \mathbf{h}_\mathcal{X}^t \cdot \left(\mathbf{P}' \left(\frac{\mathbf{h}'_\mathcal{Z}}{\mathbf{h}_\mathcal{X}^t \mathbf{P}'}\right)^T\right)^T \\ &= \mathbf{h}_\mathcal{X}^t \cdot \left(\alpha\mathbf{P}' \left(\frac{\mathbf{h}'_\mathcal{Z}}{\mathbf{h}_\mathcal{X}^t \alpha\mathbf{P}'}\right)^T\right)^T \end{aligned} \tag{12}$$

である. ここで α は何らかの定数である. この定数として適切な値を設定することで $\mathbf{P}$ の各値を大きくし, 素朴な方法では扱えなかった倍精度浮動小数でも扱えるようにする.

3.4.1 α の決定方法の直感的説明

RAPPOR の遷移確率行列 $\mathbf{P}'$ の例を図 3 に示す. この図からもわかる通り遷移確率は p, $1-p$, q, $1-q$ の計 4 つの値の組合せの積から成る. この例では各値共通して $1-q$ が出現することがわかる. $\alpha = 1-q$ にしても差支えがなく, むしろその他の値は各値ごとに異なることから, 余分な掛け算を減らした各値ごとの差異を表現可能にする適切な α であると言える. これらより, 定数 α としては, $\mathbf{P}'$ の各値には共通して出現する p, $1-p$, q, $1-q$ の組合せの積を用いれば良い.

3.4.2 扱えるデータサイズの理論解析

本節では前節と同様に RAPPOR を対象に, $\alpha\mathbf{P}'$ の行列値の最小値に関して, 扱うデータサイズ D と p, q の関係式を導き出し, 倍精度浮動小数点で扱える条件を導く.

行列の値の最小値 one-hot バイナリベクトル $\mathbf{b} \in \mathcal{Z}$ が Perturb により $\mathbf{b}' \in \mathcal{Z}$ となった際のベクトルの各値について, 1 が 1 になった個数を c_1, 1 が 0 になった個数を $1 - c_1 = \bar{c_1}$, 0 が 1 になった個数を c_0, 0 が 0 になった個数を

[*3] 入力では one-hot バイナリベクトルであったが, 出力は one-hot 表現になるとは限らないことから, Randomized Response とは異なる

[*4] 提案手法と [18] の違いは, [18] らは式 (9) に似た式を用いている点 (行列表現である式 (10) と比べ BLAS ライブラリが使えない点や実装方法によっては計算量が大きくなる. 式 (11) と比較した場合定数倍計算量が大きくなる.), およびこれから述べる扱えるデータサイズ D である

$$\begin{array}{c} \\ 0\,0\,1 \\ 0\,1\,0 \\ 1\,0\,0 \end{array} \begin{array}{c} \begin{array}{ccc} 0\,0\,0 & 0\,0\,1 & 1\,0\,0 \end{array} \\ \begin{pmatrix} (1-p)(1-q)^2 & p(1-q)^2 & (1-p)(1-q)q \\ (1-p)(1-q)^2 & (1-p)(1-q)q & (1-p)(1-q)q \\ (1-p)(1-q)^2 & (1-p)(1-q)q & p(1-q)^2 \end{pmatrix} \end{array}$$

図 3 3×2^3 の Unary Encoding による遷移確率行列 $\mathbf{P}$ のうち, $h_{\mathcal{Z}}((0,1,0)) = 0, h_{\mathcal{Z}}((0,1,1)) = 0, h_{\mathcal{Z}}((1,0,1)) = 0, h_{\mathcal{Z}}((1,1,0)) = 0, h_{\mathcal{Z}}((1,1,1)) = 0$ の場合の $\mathbf{P}'$. 共通して $1-q$ があることがわかる.

$D-1-c_0 = \bar{c_0}$ とする ($D = c_1 + (1-c_1) + c_0 + D - 1 - c_0$ である). なお遷移確率の値は $p^{c_1}(1-p)^{\bar{c_1}}q^{c_0}(1-q)^{\bar{c_0}}$ となる. 収集した $\{\mathbf{b}'_i\}_{i=1}^N$ の各 $\mathbf{b}'_i$ に対して $c_1, c_0, \bar{c_1}, \bar{c_0}$ を計算し, $\{\mathbf{b}'_i\}_{i=1}^N$ 中の最小値 ${c_1}^{min}, {c_0}^{min}, {\bar{c_0}}^{min}, {\bar{c_1}}^{min}$ および最大値 ${c_1}^{max}, {c_0}^{max}, {\bar{c_0}}^{max}, {\bar{c_1}}^{max}$ を得る. そうすると共通して出現する p, $1-p$, q, $1-q$ の組合せの積, すなわち α は, $\alpha = p^{{c_1}^{min}}(1-p)^{{\bar{c_1}}^{min}}q^{{c_0}^{min}}(1-q)^{{\bar{c_0}}^{min}}$ となる (最小値 ${c_1}^{min}, {c_0}^{min}, {\bar{c_0}}^{min}, {\bar{c_1}}^{min}$ は必ず対応する $p, q, 1-p, 1-q$ がその回数分出現することを意味する). $\{\mathbf{b}'_i\}_{i=1}^N$ から度数ベクトル $\mathbf{h}_{\mathcal{Z}}$ を算出し疎性を考慮して作成した $\mathbf{P}'$ に対して $\alpha\mathbf{P}'$ の行列の値の最小値の最悪ケースは, $p^{{c_1}^{max}-{c_1}^{min}}(1-p)^{{\bar{c_1}}^{max}-{\bar{c_1}}^{min}}q^{{c_0}^{max}-{c_0}^{min}}(1-q)^{{\bar{c_0}}^{max}-{\bar{c_0}}^{min}}$ となる.

$\alpha\mathbf{P}'$ の行列値の最小値がどの程度になるかは, ${c_1}^{max}, {c_1}^{min}, {\bar{c_1}}^{max}, {\bar{c_1}}^{min}, {c_0}^{max}, {c_0}^{min}, {\bar{c_0}}^{max}, {\bar{c_0}}^{min}$ が分かればよい. 加えて $c_1 \in \{0,1\}$ なためほとんど影響がないことから, c_0 および $\bar{c_0}$ のみの最大値と最小値の差を見積れれば十分である.

c_0 および $\bar{c_0}$ の最大値と最小値の差 RAPPOR において, バイナリベクトルの値が 0 の際に $q, 1-q$ に従い値を変更することはベルヌーイ試行を $D-1$ 回行った分布 (二項分布) に従うといえる. よって c_0 および $\bar{c_0}$ の期待値はそれぞれ $\mu(c_0) = (D-1)q, \mu(\bar{c_0}) = (D-1)(1-q)$ であり, 標準偏差はそれぞれ $\sigma(c_0) = \sqrt{(D-1)q(1-q)}, \sigma(\bar{c_0}) = \sqrt{(D-1)q(1-q)}$ である. 二項分布は期待値および標準偏差が十分大きい場合 (5 以上), 正規分布に近似できることが知られている. よってそれぞれの最大値と最小値の差は近似した正規分布の信頼区間も用いれば良い. 例えば, 標準偏差の 2 倍を信頼区間とすれば 95%の信頼度, 標準偏差の 4 倍を信頼区間とすれば 99.993666%信頼度となる. 扱うデータ数 N が多くなればなるほど試行回数が増え信頼区間に収まりにくくなることから, データ数に応じた信頼度を設定し, それに対応する信頼区間を設定すると良い.

数値例 $N = 1,000,000$, $D = 10,000$, $\epsilon = 1$, p, q は Symmetric Unary Encoding を考える ($p = 1-q$ である). このケースにおいて $\alpha\mathbf{P}'$ の行列値の最小値が倍精度浮動小数点に収まるかどうかを求める. $N = 1,000,000$ であることから信頼度を 99.9999998027% ($0.998 = 0.999999998027^{1000000}$ で成り立つ) とする. この際の信頼区間は標準偏差の 6 倍である. $\alpha\mathbf{P}'$ の最小値の見積りの値が $q^{6\sqrt{Dq(1-q)}} * (1-q)^{6\sqrt{Dq(1-q)}} = 1.17E-183 > 4.9E-324$ であることから, 約 99.8%の確率で取り扱うことができるといえる.

4. 実験

RAPPOR で撹乱し集約時に一般化逐次ベイズ法 (式 (12)) を用いる手法 (以下, 提案手法) の性能を検証するため, 度数ベクトル推定誤差の評価を行う.

4.1 データセット

データセットは人工データを用いる. 人工データとして, Zipf 則に従うデータ, 幾何分布に従うデータで実験を行う.

Zipf 則に従うデータ Zipf 則は, 都市の人口やウェブのアクセス頻度といった社会現象に成り立つことが確認されているものである. Zipf 則に従うデータは, 式 13 に示す分布に従うようにデータを N 個生成する.

$$f(D; s, N) = \frac{1/D^s}{\sum_{n=1}^{N} 1/n^s} \tag{13}$$

本実験では zipf 則の一般的なパラメータである $s = 1$ で固定し実験を行った.

幾何分布に従うデータ 幾何分布に従うデータは, 式 14 に示す分布に従うようにデータを N 個生成する.

$$f(D; s) = (1-s)^D s \tag{14}$$

本実験で [1] らと同様に $s = 0.8$ で固定し実験を行った.

4.2 比較手法

比較手法として, 節 2.3 で示した Randomized Response[7], RAPPOR[4] に加え, Subset Selection[14], [17] および Hadamard Response[1] を対象とする. Randomized Response は逐次ベイズ法を用いたもの[*5], RAPPOR はベクトルの要素ごとに独立に逆行列法を用いたものとする. Subset Selection[14], [17] は, 出力の範囲をハミング重みが一定になるよう調整した方式であり, 現在度数推定誤差が最も小さい手法である. Hadamard Response[1] は送信時に必要なデータ量と度数推定誤差のバランスを兼ね備えた手法である.

4.3 実験設定

乱数が含まれることから 10 回試行を行い, その平均値を結果として採用することとする. 人工データは $(N, D) = (1000, 1000), (10000, 1000), (100000, 1000)$ とし, $\epsilon = \{1, 2, 4\}$ で実験を実施した. 提案手法の η は D^{-4}, RAPPOR における p と q は Symmetric Unary Encoding とした. 真の度数分布を $\mathbf{h}^*_{\mathcal{X}}$ とした場合度数分布の誤差は各度数の二乗誤差の総和で評価することとする.

$$\sum_{x \in \mathcal{X}} \left(\frac{\mathbf{h}^*_{\mathcal{X}}(x)}{N} - \frac{\mathbf{h}_{\mathcal{X}}(x)}{N} \right)^2 \tag{15}$$

二乗誤差は各度数ごとの誤差が大きいほどペナルティが大

[*5] 逐次ベイズ法の推定誤差を改善する方式が [9] で提案されており, 時間計算量が $O(ND^2)$ である. 今回扱うデータ規模では, 他の手法と比べて一定時間内で処理が終わらず比較の対象外とした

きくなる．より具体的には真の度数が大きい箇所で誤差が大きいと誤差が大きくなる．真の度数が大きい箇所はなるべく値を保持すべきであることから絶対誤差でなく二乗誤差を採用した．

4.4 実験結果

Zipf 則に従うデータの実験結果を表 1, 幾何分布に従うデータの実験結果を表 2 に示す．N を増やすに従い総じて推定誤差が小さくなっていることがわかる．Zipf 則に従うデータの $\epsilon = 1$ の際の $N = 1000, 10000$ のケースにおいては, それぞれ Subset Selection および RAPPOR が良い結果となっている．一方それ以外のすべてにおいて提案手法が優れているといえる (16/18 で提案手法が優れている)．例えば, 表 1 中の $N = 10,000, \epsilon = 4$ において, SubsetSelection が RAPPOR と比べて約 0.001 の改善であるが, 提案手法は約 0.006 の改善すなわち約 6 倍改善できていることがわかる．表 2 中の $N = 10,000, \epsilon = 2$ において, SubsetSelection が RAPPOR と比べて約 0.003 の改善であるが, 提案手法は約 0.06 の改善であり約 20 倍改善できているといえる (他においても同様に大幅な改善がある)．

総じて D に対して N が多い場合は提案手法が最も優れている．表 1 中の $N = 100,000, \epsilon = 1$ において, SubsetSelection が RAPPOR と比べて約 0.003 の改善であるが, 提案手法は約 0.007 の改善であり約 2.5 倍改善できている (同様に改善多数)．そもそも度数分析を行う場合はユーザ数が多い方が好ましいことから, ユーザ数 N が多い状況下で推定誤差が小さいことは実応用上も有益であるといえる．総数制約および非負制約を満たしたアルゴリズムであることから推定性能がよかったといえる．

5. おわりに

本研究では, 逐次ベイズ法を一般化した一般化逐次ベイズ法を提案した．この手法は遷移確率で表現できるランダム化手法に対して適用できる方法である．また RAPPOR に対して一般化逐次ベイズ法を適用できることを示し, 計算量の削減および扱えるデータサイズを増やす方法を提案した．数値実験により, 多くのケースにおいて既存の手法と比べて度数推定の誤差が小さいことを確認し, あるケースにおいて二乗誤差が約 20 倍以上改善することを確認した．

参考文献

[1] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. In *The 22nd International Conference on Artificial Intelligence and Statistics 2019*, Vol. 89, pp. 1120–1129, 2019.

[2] Rakesh Agrawal, Ramakrishnan Srikant, and Dilys Thomas. Privacy preserving olap. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 251–262, 2005.

[3] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284, 2006.

[4] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

[5] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 211–222, 2003.

[6] Matthew Joseph, Aaron Roth, Jonathan Ullman, and Bo Waggoner. Local differential privacy for evolving data. In *Advances in Neural Information Processing Systems 31*, pp. 2375–2384. 2018.

[7] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. Discrete distribution estimation under local privacy. In *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48, pp. 2436–2444, 2016.

[8] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, Vol. 40, No. 3, pp. 793–826, 2011.

[9] Takao Murakami, Hideitsu Hino, and Jun Sakuma. Toward distribution estimation under local differential privacy with small samples. *Proceedings on Privacy Enhancing Technologies*, Vol. 2018, No. 3, pp. 84 – 104, 2018.

[10] Takao Murakami and Yusuke Kawamoto. Utility-optimized local differential privacy mechanisms for distribution estimation. In *28th USENIX Security Symposium*, pp. 1877–1894, Santa Clara, CA, 2019.

[11] Daniel R Ramage, Jayadev Acharya, KA Bonawitz, Peter Kairouz, and Ziteng Sun. Context-aware local differential privacy. In *International Conference on Machine Learning 2020*, 2020.

[12] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 05, pp. 557–570, 2002.

[13] Apple's Differential Privacy Team. Learning with privacy at scale. "https://machinelearning.apple.com/research/learning-with-privacy-at-scale".

[14] Shaowei Wang, Liusheng Huang, Pengzhan Wang, Yiwen Nie, Hongli Xu, Wei Yang, Xiang-Yang Li, and Chunming Qiao. Mutual information optimally local private discrete distribution estimation. *CoRR*, Vol. abs/1607.08025, , 2016.

[15] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium*, pp. 729–745, 2017.

[16] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, Vol. 60, No. 309, pp. 63–69, 1965.

[17] Min Ye and Alexander Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. *IEEE Transactions on Information Theory*, Vol. 64, No. 8, pp. 5662–5676, 2018.

[18] 堀込光, 菊池浩明. Local differential priacy によりプライバシーを考慮した位置情報分布推定. DICOMO2020 シンポジウム, pp. 1–8, 2020.

$$
\begin{array}{c} \\ 0\,0\,1 \\ 0\,1\,0 \\ 1\,0\,0 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 0\,0\,0 & 0\,0\,1 & 0\,1\,0 & 0\,1\,1 & 1\,0\,0 & 1\,0\,1 & 1\,1\,0 & 1\,1\,1 \end{array} \\
\begin{pmatrix}
(1-p)(1-q)^2 & p(1-q)^2 & (1-p)(1-q)q & p(1-q)q & (1-p)(1-q)q & p(1-q)q & (1-p)q^2 & pq^2 \\
(1-p)(1-q)^2 & (1-p)(1-q)q & p(1-q)^2 & p(1-q)q & (1-p)(1-q)q & (1-p)q^2 & p(1-q)q & pq^2 \\
(1-p)(1-q)^2 & (1-p)(1-q)q & (1-p)(1-q)q & (1-p)q^2 & p(1-q)^2 & p(1-q)q & pq(1-q) & pq^2
\end{pmatrix}
\end{array}
$$

図 4 3×2^3 の RAPPOR による遷移確率行列. 行が入力, 列が出力に対応しており, 行列の各値は式 (6) で定義した遷移確率の掛け算となる.

表 1 Zipf 則に従うデータにおける推定誤差

ϵ		1			2			4		
N		1000	10000	100000	1000	10000	100000	1000	10000	100000
method	Proposal	0.061166	0.030564	**0.012252**	**0.020096**	**0.007756**	**0.002567**	**0.004577**	**0.001811**	**0.000565**
	Randomized Response	0.080039	0.070858	0.026655	0.092624	0.043097	0.006668	0.054696	0.014717	0.0012
	Subset Selection	**0.028357**	0.02657	0.019175	0.024449	0.018555	0.007153	0.015037	0.007309	0.001114
	Rappor	0.028689	**0.026302**	0.022197	0.024666	0.020109	0.011751	0.015413	0.008151	0.002661
	Hadamard Response	0.029097	0.027173	0.024312	0.025281	0.020726	0.014566	0.017255	0.009966	0.005667

表 2 幾何分布データにおける推定誤差

ϵ		1			2			4		
N		1000	10000	100000	1000	10000	100000	1000	10000	100000
method	Proposal	**0.10464**	**0.03681**	**0.00664**	**0.01983**	**0.00508**	**0.00077**	**0.00275**	**0.00065**	**8.67E-05**
	Randomized Response	0.14762	0.14365	0.05918	0.17087	0.07861	0.01354	0.09574	0.01896	0.00164
	Subset Selection	0.10529	0.09493	0.06835	0.08899	0.06718	0.03137	0.05716	0.03073	0.00733
	Rappor	0.10543	0.09829	0.08064	0.0901	0.0709	0.04451	0.05707	0.03349	0.01344
	Hadamard Response	0.10636	0.09888	0.08693	0.09265	0.07493	0.05478	0.06371	0.03813	0.02384