Computer Security Symposium 2019
21 - 24 October 2019

一般化とランダム化を組み合わせた匿名化手法の提案及びその具体的な識別リスク計算方法の検討

長谷川聡[1,a)]　三浦尭之[1]

概要：本研究では, 一般化とランダム化を組み合わせた匿名化手法を提案する. 一般化とランダム化を組み合わせた手法は NP-困難な問題であるため, ヒューリスティック最適値探索アルゴリズムを用いることで, 実用的な時間で実行可能なアルゴリズムを示す. 加えてランダム化を施した際の具体的な識別リスクを計算する方法を提案する. 通常時間計算量が階乗オーダ必要なアルゴリズムを指数オーダに改善したことを報告する.

キーワード：匿名化, 識別リスク

Proposal of anonymization method by generalization and randomizaiton and how much is explicit identification risk?

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

Abstract: In this paper, we propose an anonymization method that combines generalization and randomization. Because the method of combining generalization and randomization is a NP-hardness problem, by using a heuristic optimal value search algorithm, an algorithm that can be executed in a practical time is shown. We propose a method to calculate explicit identification risk of randomized. We denote that the algorithm that requires $O(N!)$ for naive method has been improved to 2^N.

Keywords: anonymity, identification risk

1. はじめに

近年の深層学習を始めとした人工知能技術の発達により, 個人に紐づく情報 (パーソナルデータ) の二次利用が注目を浴びている. しかしながらパーソナルデータは, 個人特定可能な情報が含まれており, 安易に第三者へデータを提供するなどの二次利用を行うと, プライバシを侵害するリスクが生じる恐れがある.

パーソナルデータ利活用に向け, 個人情報保護法が改正され, 2017 年 5 月 30 日に全面施工された. 改正個人情報保護法では, パーソナルデータを**匿名加工情報**と呼ばれる一定の基準を満たした情報に加工すれば, 収集時とは異なる目的の利用を可能にするよう規定された.

パーソナルデータを匿名加工情報にする加工技術の 1 つとして, 「匿名化」技術がある [7], [11], [23]. 匿名化は, データベース中に含まれるレコードを加工し, 個人識別を困難にすることでプライバシを保護する技術である. 匿名化データがどの程度プライバシを保護しているかを示す指標として, k-匿名性 [23] が提案されている. k-匿名性とは, 「データベース中に同じ準識別子 (**Q**uasi **I**dentifier)[*1] を持つレコードが少なくとも k 個以上存在する」ことでプライバシ保護の度合いを表す. k-匿名性は個人識別を防ぐことができるが, 属性推定といった別の攻撃を防ぐことはできず, その派生系として l-多様性 [15] といった指標が提案されている.

近年では「個人が含まれていてもいなくても出力結果にほとんど違いがない」という基準のプライバシ保護指標である $\epsilon-$ 差分プライバシ [5] が提案されており, データベー

[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

[*1] 準識別子とは, それ単体では個人が特定できないが, 組み合わせることで個人が特定可能な属性のことをいう. 例えば, 年齢, 住所, 性別などがそれらに該当すると言われている.

ス, 統計, 機械学習分野 [3] で盛んに研究されている [6]. 差分プライバシは, 任意の攻撃者に対して有効な手法として知られている. しかしながら, 具体的な個々の攻撃に対して, どの程度リスクがあるかは自明ではない. 例えば, ある ϵ について差分プライバシを満たしたとしても, 個人識別のリスクはそれ以外の要素, 例えば計算する統計値や母集団の分布に強く依存してしまうことが知られている [12]. また, 機械学習の出力に対する攻撃として近年 Membership Inference Attack[21] といった具体的な手法が提案されており, 差分プライバシが Membership Inference Attack を防ぐことができるか非自明であるという議論 [26] もある.

本研究では, 差分プライバシ基準に加えて k-匿名性といった特定の攻撃のリスクを評価可能なデータベース匿名化手法を対象とする. 特に, 差分プライバシを満たしつつ [14], 特定の攻撃に対するリスク指標である k-匿名性 [9], [23] や l-多様性 [15], [27] を評価可能な Post Randomisation(以降 PRAM)[25] と呼ばれる方式に着目する.

1.1 PRAM に関する問題

PRAM はデータベース内の各値を一定の確率に従いランダムに書き換えることでプライバシの保護を行う方式である [25]. Randomized Response[24] とも呼ばれており, データベース開示に加え, 局所差分プライバシ [4] の枠組みなど幅広く利用されている技術である.

PRAM は幅広く利用されている手法である一方, 対象となるデータベースの属性数および属性値のバリエーションが多くなればなるほど, プライバシ保護に必要なノイズの量が大きくなる [9], [14]. その結果プライバシ保護されたデータベースは, 元のデータベースと比べて誤差が大きくなってしまう.

1.2 ランダム化の安全性評価方法の問題

ランダム化による匿名性評価の方式として, ランダム化手法に対して k-匿名性と同等の安全性を評価可能にした Pk-匿名性が提案されており, PRAM は Pk-匿名性を満たすことが示されている [9]. Pk-匿名性を満たした PRAM はその性質上, 攻撃者の持つ背景知識として最も識別確率が高くなるようなデータベースを想定してランダム化を行うことから, 実際に匿名化されたデータベースには強めにノイズが加えられている場合がある.

現状では Pk-匿名性を満たすようにランダム化されたデータベースは, データベース全体が Pk-匿名性を満たすということしか言えない. 言い換えれば, Pk-匿名性を満たすようにランダム化されたデータベースの各レコードが, 具体的にどの程度識別確率を有するかは自明ではない. そのため, 個人単位の識別リスクの評価ができず, より細かい有用性の向上手法を検討した際に問題となる.

1.3 貢献

本研究では, これらの課題に対し 2 つの提案を行う. 1 つは, PRAM に一般化を組み合わせた方式を提案する. これにより PRAM に関する課題である属性値のバリエーション増加による誤差の増大を防ぐ. もう 1 つは, PRAM 等のランダム化方式によりプライバシ保護されたデータベースの各レコード単位の具体的な識別確率を計算する方法を提示する. 特に,

- ベイズ最適化を利用した一般化とランダム化を組み合わせる時間効率の良い方式の提案,
- ランダム化されたデータベースの識別確率の時間計算量を階乗オーダから指数オーダへ改善,

が本研究の技術的な貢献である. 一般化による最適な k-匿名化は NP-困難であることが知られており [16], ランダム化を組み合わせた方式でも同様に問題規模が大きくなると計算が実用的な時間で終了しない可能性がある. そこで, ヒューリスティックに最適解を探索するアルゴリズムであるベイズ最適化 [18] を用いることで短時間処理を実現する.

ランダム化されたデータベースの識別確率の計算には, データベース内のレコードのシャッフルのパターンの組み合わせ数分計算が必要となる. より具体的には, レコードのシャッフルのパターンの組み合わせはレコード数を N とした場合, $N!$ のパターンが存在するため, $O(N!)$ の計算量が必要となる. その計算量を回避するため, 識別確率の計算を行列のパーマネント [19] の計算に置き換える方針をとる. 行列のパーマネントは愚直に計算すると $O(N!)$ 要するが, $O(2^N)$ で済む効率の良い方法が知られており, 本研究でもその方法を利用することで計算量を改善する.

	住所	年齢	性別
1	東京都武蔵野市	50	男
2	神奈川県横須賀市	55	男
3	東京都武蔵野市	50	男

図 1 テーブル t の例. 属性全体の集合 $\mathcal{A}$ = { 住所, 年齢, 性別 }, 性別の属性値の集合 $\mathcal{A}_{gender}$ = { 男, 女 }, であり, [3] = {1,2,3} から属性値への写像が t である.

2. 準備

記号定義や提案手法に用いる技術の説明を行う. またデータベースを表形式 (以降テーブルと呼ぶ) に限定して議論を進める.

2.1 記号定義

集合 A, B に対し, A から B への写像全体の集合を $\Rightarrow$ と記す. ベクトルをボールドフォント $\mathbf{n}$, 行列を大文字のボールドフォント $\mathbf{N}$ と記す. また, 正の整数 $N \in \mathbb{Z}_+$ に対し, $[N] := \{1, ..., N\}$ とおく.

2.1.1 テーブルの定義

$\mathcal{R}$ を個人からなる集合とする. $|\mathcal{R}| = N$ とし, $\mathcal{R}$ の各個人に $1, ..., N$ のラベルを付与する. $\mathcal{A}$ を属性全体の集合, 各 $a \in \mathcal{A}$ に対し, $\mathcal{V}_a$ を属性値の集合とし, $\mathcal{V} := \prod_{a \in \mathcal{A}} \mathcal{V}_a$ とおく. このとき, 正の整数 $N \in \mathbb{Z}_+$ に対し, $T := [N] \Rightarrow \mathcal{V}$ とおき, 各 $t \in T$ をテーブルと呼ぶこととする.

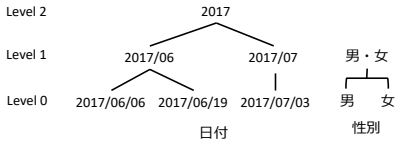

図 2 一般化階層の例．それぞれ日付 (高さ 3), 性別 (高さ 2) の一般化階層を表す．例えば, 日付は $g^{(1)}_{\mathrm{date}}$("2017/06/19") = "2017/06", $g^{(2)}_{\mathrm{date}}$("2017/06/19") = "2017" となる．

2.1.2 度数分布の定義

テーブル t 内の属性値集合 $\mathcal{V}_t = \{t(i) \in \mathcal{V} | i \in [N]\}$ のある属性値 $v \in \mathcal{V}_t$ の出現頻度を $h_t : \mathcal{V}_t \to \mathbb{Z}_+$ と表すこととし, テーブル t 内のすべての属性値の出現頻度を並べたベクトルを $\mathbf{h}_t$ と表し度数分布と呼ぶこととする．例えば図 1 の度数分布は, $\mathbf{h}_t = (2,1)^T$ となる．

2.1.3 テーブル保護

写像 $\Delta : T \to T'$ をテーブル保護と呼ぶ．これは入力されたテーブル $t \in T$ に対して, t に応じた確率分布に従ってテーブル t' を出力するランダム化を表す．

2.1.4 シャッフル

写像 $\gamma : [n] \to [n]$ をシャッフルと呼ぶこととする．また写像 γ 全体の集合を Γ とする (シャッフルのすべてのパターンの集合)．

2.2 一般化と *k*-匿名性

テーブル中の個人のプライバシを保護する処理として一般化がある．一般化は属性値を上位概念に置き換える処理のことを言う．一般化は, 図 2 に示すような概念関係を表した階層構造に基づき値の置換処理が行われる．ある属性 $a \in \mathcal{A}$ の属性値 v_a を j 番目の上位概念に置換する処理を $g^{(j)}_a : \mathcal{V}_a \to \mathcal{V}'_a$ とおき, 上位概念を m 個持つ一般化階層を $\mathcal{G}_a = \{g^{(0)}_a, ..., g^{(m)}_a\}$ とする．ただし, $g^{(0)}_a$ は恒等写像 (上位概念に置き換わらず自身に置き換わる) とする．

k-匿名性を満たすよう一般化を行う方法として, [11], [13] 等がある．これらは, 階層構造を用いた上位概念への置換パターンを網羅的に試行し, k-匿名性を満たしつつ有用性が最も高い上位概念への置換パターンを発見することで実現する (図 3)．

2.3 ランダム化と *Pk*-匿名性

一般化と異なりランダムにテーブルを書き換えることでプライバシを保護する方法もある．ランダム化による匿名性の評価方法として, k-匿名性をランダム化に対して適用可能にする Pk-匿名性 [9] が提案されている．Pk-匿名性は個人の識別推定確率を $1/k$ で抑えているようなテーブル及びランダム化に対して持つ性質のことをいう．

個人の識別推定確率 $r, r' \in [N]$, $t' \in T'$, X_T を T-値確率変数 (その確率分布を f_T とする) とき,

$$\eta(f_T, t', r, r') := \Pr[\Gamma(r) = r' | \Delta(X_T) = t' \circ \Gamma], \quad (1)$$

を識別推定の確率とする．任意の背景知識を持つ攻撃者を想定し, どのレコードかを識別する確率を表す．

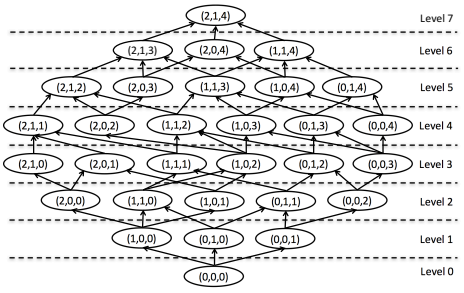

図 3 高さ 3, 高さ 2, 高さ 5 の一般化階層の上位概念への置換パターンの例である．(0,0,0) はそれぞれレベル 0 の値であり, 例えば日付を 1 階層上げた場合 (1,0,0) となる．図の右側に示すレベルは, 階層上げを何回行ったかを表している．この置換パターンを網羅的に探索し, 有用性が最も良い置換パターンを採用する

***Pk*-匿名性** $k \in \mathbb{Z}_+$ とする．テーブル保護系 $\mathcal{P} = (T, T', \Gamma, \Delta)$ とする．匿名化テーブル $t' \in T'$ とテーブル保護系の組 $(\mathcal{P}, t')$ が Pk-匿名性を満たすとは, 任意の攻撃者 f_T, $r \in [N], r' \in [N]$ に対して,

$$\eta(f_T, t', r, r') \leq \frac{1}{k}, \quad (2)$$

が成り立つことをいう．個人の識別推定確率が $1/k$ で抑えられていることを表す．

2.3.1 PRAM

Pk-匿名性を満たすランダム化手法として, PRAM がある [25]．PRAM は遷移確率行列と呼ばれる行列に基づき, テーブルの各レコードの属性値を確率的に置換することでプライバシを保護する．遷移確率行列とは, ある属性 $a \in A$ の属性値 $v_a \in \mathcal{V}_a$ から属性値 $v'_a \in \mathcal{V}'_a$ に置換される確率 $\Pr(v'_a | v_a)$(遷移確率) を要素に持つ $|\mathcal{V}_a| \times |\mathcal{V}'_a|$ 行列 $\mathbf{P}_a$ であり, この行列に基づきテーブル内の各値を 1 つずつランダムに書き換える ($\mathcal{V}_a = \mathcal{V}'_a$ とする)．

遷移確率の設定の方法として式 (3) のように, 一定の確率 ρ_a で値を保持し, それ以外の確率 $1 - \rho_a$ で値をランダムに書き換える方法が考えられる．

$$\Pr(v'_a | v_a) = \begin{cases} \rho_a + \frac{(1-\rho_a)}{|\mathcal{V}_a|} & (v_a = v'_a) \\ \frac{(1-\rho_a)}{|\mathcal{V}_a|} & (v_a \neq v'_a) \end{cases}. \quad (3)$$

属性値 $v \in \mathcal{V}$ の遷移確率行列 $\mathbf{P}$ は, 各属性 $a \in \mathcal{A}$ の遷移確率行列 $\mathbf{P}_a$ のクロネッカー積 $\otimes$ で表現される．

$$\mathbf{P} = \mathbf{P}_{|\mathcal{A}|} \otimes \cdots \otimes \mathbf{P}_a \otimes \cdots \otimes \mathbf{P}_1 \quad (4)$$

2.4 ベイズ最適化

ベイズ最適化とは, 式 (5) に示すような, 関数 f の最小値 (もしくは最大値) を求める最適化手法の 1 つである [18]．

$$\arg\min_{\mathbf{x} \in \mathcal{X}} f(x). \quad (5)$$

ここで, $\mathcal{X}$ は入力 $\mathbf{x}$ の集合を表す．ベイズ最適化は特に, (a) 関数 f の形状が不明, (b) 関数 f の評価コストが高い 際

Algorithm 1 ベイズ最適化アルゴリズム

Require: f, $\mathcal{X}$, $iter \in \mathbb{Z}_+$, $init \in \mathbb{Z}_+$
Ensure: 関数 f の最小値を与える $\mathbf{x}^*$
1: 関数 f を推定する代理関数 f' を用意し, 初期サンプル $\{(\mathbf{x}_1, f(\mathbf{x}_1)), ..., (\mathbf{x}_{init}, f(\mathbf{x}_{init}))\}$ を用いて f を推定する (f' を学習する).
2: $i = init + 1$
3: **while** $i \leq iter$ **do**
4: f' を用いて獲得関数を更新し, 次の探索すべき $\mathbf{x}_i$ を選ぶ.
5: $\mathbf{x}_i$ より $f(\mathbf{x}_i)$ を計算し, 代理関数 f' を更新する.
6: $i \leftarrow i + 1$
7: **end while**
8: $iter$ 回分の $\mathbf{x}, f(\mathbf{x})$ のうち最小値をとる $f(\mathbf{x})$ の $\mathbf{x}$ を $\mathbf{x}_*$ とする.

に効率良く最適化を行う手法である.

ベイズ最適化アルゴリズムは大まかにアルゴリズム 1 に示すように, 逐次的なアルゴリズムとなっている. まず $\mathcal{X}$ からランダムに初期値をサンプルし, 関数 f を推定 (推定した関数 f' は Surrogate Function(代理関数) と呼ばれる) する. 推定した関数 f' を用いて次に探索すべき点 $\mathbf{x}$ を選択 (Acquisiton Function(獲得関数) に相当) し, $f(\mathbf{x})$ を評価する. そしてこれまで得られた入出力 $\{\mathbf{x}, f(\mathbf{x})\}$ を用いて再び関数 f を推定 (代理関数 f' を学習) する. 代理関数の学習・獲得関数の評価を繰り返すことで最適な $\mathbf{x}^*$ を探索する. 代理関数 f' はガウス過程回帰 [2] が良く用いられ, 獲得関数としては Lower Confidence Bound(LCB)[22] 等が利用される.

2.4.1 LCB とガウス過程回帰を用いたベイズ最適化

LCB Lower Confidence Bound(下側信頼限界) とは式 (6) に示すように, $f'(\mathbf{x})$ の信頼区間の下限が最も低い点を次に探索すべき点 $\mathbf{x}_{i+1}$ にする方法である [22].

$$\arg\min_{\mathbf{x}_{i+1}} \mathbb{E}[f'(\mathbf{x}_{i+1})] - \sqrt{\frac{\log(i)}{i}}\mathbb{V}[f'(\mathbf{x}_{i+1})] \qquad (6)$$

式 (6) は非凸な関数であり勾配降下法等では局所解に陥ってしまうため, CMA-ES[8] 等を用いて大域的最適解を求めると良い.

ガウス過程回帰 ガウス過程回帰は入力 $\mathbf{x} \in \mathbb{R}^d$ から出力 $y \in \mathbb{R}$ への関数 $y = f(\mathbf{x})$ を推定するモデルの 1 つである [2]. 非線形性があるため線形回帰では表せない関数を表現できる. 加えてベイズ推定を用いていることにより, 関数が 1 つに定まらず分布として求まるため, 推定の不確かさを表現可能である (この推定の不確かさは式 (6) で用いる分散そのものである)

ここでは, ガウス過程回帰の詳細は省き, LCB の計算に必要な $\mathbb{E}[\mathbf{x}]$ および $\mathbb{V}[\mathbf{x}]$ の計算方法のみ示す. i 番目までのデータ $\{(\mathbf{x}_1, f(\mathbf{x}_1)), ..., (\mathbf{x}_i, f(\mathbf{x}_i))\}$ が観測されているとし, $\mathbf{x}_{i+1}$ を入力した際の $\mathbb{E}[f'(\mathbf{x}_{i+1})], \mathbb{V}[f'(\mathbf{x}_{i+1})]$ を求める. ガウス過程回帰の計算に必要となるカーネル関数 $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ と呼ばれる関数を用意する[*2]. 期待値 $\mathbb{E}[f'(\mathbf{x}_{i+1})]$ は, 対象となる $\mathbf{x}_{i+1}$ とこれまで得た i 個のデータとのカーネル関数値を並べたベクトル $\boldsymbol{\kappa} = (\kappa(\mathbf{x}_{i+1}, \mathbf{x}_1), ..., \kappa(\mathbf{x}_{i+1}, \mathbf{x}_i))^T$, $\kappa_{j,l} = \kappa(\mathbf{x}_j, \mathbf{x}_l)$ を要素に持つ $i \times i$ 行列 $\mathbf{K}$, $\mathbf{y} = (f(\mathbf{x}_1), ..., f(\mathbf{x}_i))^T$ から

$$\mathbb{E}[f'(\mathbf{x}_{i+1})] = \boldsymbol{\kappa}^T\mathbf{K}^{-1}\mathbf{y}, \qquad (7)$$

と求めることができる. 加えて分散 $\mathbb{V}[f'(\mathbf{x}_{i+1})]$ は,

$$\mathbb{V}[f'(\mathbf{x}_{i+1})] = \kappa(\mathbf{x}_{i+1}, \mathbf{x}_{i+1}) - \boldsymbol{\kappa}^T\mathbf{K}^{-1}\boldsymbol{\kappa} \qquad (8)$$

と求めることができる.

代理関数 f' の更新は $\mathbf{x}_{i+1}$ を追加して期待値と分散を計算することであるゆえ, ガウス過程の場合は代理関数の更新は明示的に行わなくて良く, 獲得関数の計算をすれば良い.

[*2] カーネル関数は対称性と $\mathbf{K}$ が正定値な行列となる性質を持つ必要がある

3. 一般化とランダム化を組み合わせた方式

一般化と PRAM を組み合わせて一定のプライバシ強度でテーブルを保護する方式を提案する. 提案手法はプライバシ強度として $(P)k$-匿名性や ϵ-差分プライバシを保持しつつ, 有用性が最も良くなるように適切な上位概念への一般化処理及び PRAM を行う.

これを実現するにあたり, 一般化とランダム化両方で評価可能な有用性指標の定義, ランダム化処理の期待値評価方法を提案する. 加えて計算効率の良い近似値計算アルゴリズムを提案する.

3.1 有用性評価指標

一般化処理を行うことにより匿名化前後で $\mathcal{V} \neq \mathcal{V}'$ となるため, 度数分布の KL-Divergence や $L1$-距離といったテーブル間の相違を測るための一般的な有用性評価指標を用いることができない. そこで一般化処理を行った匿名化テーブルの属性値集合 $\mathcal{V}'$ と元テーブルの属性値集合 $\mathcal{V}$ とを結びつける処理を行うこととする.

匿名化テーブル t' に対する属性値 $v \in \mathcal{V}$ の度数を定義する. アイディアは, 一般化によって複数の属性値が上位概念に置き換わり増えた度数を, 元の概念に均等に度数を割り振るということを行う. 匿名化テーブル t' 作成の際に利用した属性 a の一般化階層を $\hat{g}_a$ とする. 匿名化前の属性値 v を, 匿名化の際に利用した一般化階層 $\hat{v} \in \{\prod_{a\in\mathcal{A}} \hat{g}_a(v_a)\}$ を行い,

$$h'_t(v) := \frac{h'_t(\hat{v})}{\prod_{a\in\mathcal{A}} |\hat{g}_a^{-1}(\hat{g}_a(v))|} \qquad (9)$$

とすることとする (図 4 参照).

これにより $L1$-距離は,

$$L1(t, t') = \sum_{i\in[N]} |h_t(t(i)) - h'_t(t(i))|, \qquad (10)$$

となる.

3.2 ランダム化時の期待値評価

ランダム化処理は処理ごとに結果がランダムに変わるた

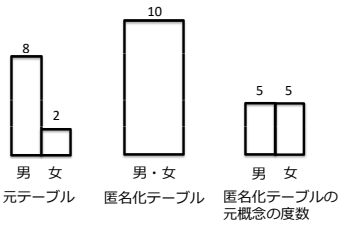

図 4 例として性別のみを持つテーブルのヒストグラムを想定. 匿名化によって男・女となった度数から, 男:5, 女:5 という形で均等に割り振る.

め, 一般化と組み合わせた際有用性評価結果が都度異なってしまう. そこでランダム化処理による結果の期待値および分散を導出することで区間推定を行い, その推定した区間の値をランダム化の処理結果として代用する.

区間推定を行う方法として, チェビシェフの不等式が一般的に用いられる. 確率変数 X に対し期待値 $\mathbb{E}[X]$ 及び分散 $\mathbb{V}[X]$ が計算可能である場合, 期待値 $\mathbb{E}[X]$ と X の差は, 式 (11) に示すチェビシェフの不等式に従う.

$$\Pr\left(|X - \mathbb{E}[X]| < \sqrt{\frac{\mathbb{V}[X]}{\theta}}\right) \geq 1 - \theta. \tag{11}$$

これは $1-\theta$ の確率で, 期待値からの誤差 $|X - \mathbb{E}[X]|$ が $\sqrt{\frac{\mathbb{V}[X]}{\theta}}$ に収まることを示している. ランダム化の結果を $X = \mathbb{E}[X] \pm \sqrt{\frac{\mathbb{V}[X]}{\theta}}$ で代用することにより決定的に評価を行うものとする.

3.2.1 PRAM の期待値と分散

PRAM による度数分布の期待値は, [1] に示されているとおり,

$$\mathbb{E}[\mathbf{h}'] = \mathbf{P}\mathbf{h}, \tag{12}$$

で算出できることが知られている.

式 (3) に示す遷移確率行列による PRAM に限定した場合, 値を維持するかそれ以外の確率で等確率にランダムに置換するかの 2 択の選択肢となるため, 度数分布の分散は二項分布の分散と捉えることができる. 行列形式で表すと,

$$\mathbb{V}[\mathbf{h}'] = (\mathbf{P} .* (\mathbf{1} - \mathbf{P}))\mathbf{h}, \tag{13}$$

とすることで計算可能である. なお, $.*$ は要素ごとの積, $\mathbf{1}$ は要素がすべて 1 である $|\mathcal{V}| \times |\mathcal{V}|$ 行列である.

3.3 ベイズ最適化との組み合わせ

節 3.1 で述べたとおり, 一般化とランダム化を併用した際の有用性の評価が可能であり, 加えて節 3.2 で述べたとおりランダム化時の期待値が計算可能である.

最適な上位概念の置換およびランダム化との組み合わせを計算する場合, 上位概念の組み合わせの網羅的なパターンの探索に加えて, ランダム化を実施するかどうかのパターンを加えた組み合わせ最適化問題を解く必要が生じる. これは一般化による k-匿名化と同様に, NP-困難な問題であることは明らかであり, 匿名化対象となる属性数が増えるほど時間計算量が膨大になる.

Algorithm 2 評価関数 f

Require: $t \in T$, u, $g = \{g_a^{(i)} | a \in \mathcal{A}, i \in [|\mathcal{G}_a|]\} \in \mathcal{G}, b \in \mathcal{B}$, $s \in \{(P)k - anonymity, \epsilon - DP\}$, θ
Ensure: 評価結果
テーブル t を一般化階層集合 g を用いて一般化を行う.
if $b =$ true **then**
  プライバシ指標 s を満たすような PRAM のパラメータを算出する (Pk-匿名性は [9],ϵ-差分プライバシは [14] を参照).
  式 11 を用いて度数の区間推定を算出し, 有用性評価関数 u を算出する.
else
  if $s =$ (P)k-anonymity **then**
    if 匿名化されたテーブル t' の k-匿名性を満たす **then**
      有用性評価関数 u を算出する.
    else
      最も低い有用性の評価結果を返す.
    end if
  else
    ϵ-差分プライバシは満たさないので, 最も低い有用性の評価結果を返す.
  end if
end if

本論文では, 大規模データへの適用を想定し, 節 2.4 にて導入したヒューリスティックな最適値探索アルゴリズムであるベイズ最適化を用いて近似解を導出する方法を提案する. ベイズ最適化を用いることで, 探索すべきパターンを減らすことができるため計算時間の短縮につながる. 加えてレコード数が多い大規模データの場合, 有用性評価や一般化処理の計算等に時間を要する. 代理関数を用いて最適化を行うベイズ最適化を用いることで効率よく近似解を見つけることが可能となる.

3.3.1 ベイズ最適化の適用

アルゴリズム 1 の入力として, 匿名化対象となるテーブル t, 一般化階層集合 $\mathcal{G} = \{\mathcal{G}_a | a \in \mathcal{A}\}$, PRAM の適用の有無 $\{\text{true}, \text{false}\} \in \mathcal{B}$ (探索するパラメータの集合は $\mathcal{X} = \{\mathcal{G}, \mathcal{B}\}$ である), プライバシパラメータ $s \in \{k - anonymity, \epsilon - DP\}$, 有用性評価関数 u, パラメータ θ, 評価関数 f を入力とし, 有用性の高い保護テーブル t' を出力する. 例えば評価関数 $f(t, \mathcal{X}, s, \theta, u)$ として, アルゴリズム 2 などが考えられ, これらを用いてアルゴリズムを動かせば良い[*3].

4. 識別推定確率の具体的な計算方法

本節では, 元テーブル $t \in T$ と匿名化テーブル $t' \in T'$ およびランダム化手法 Δ が決定した際に, 具体的な識別推定確率 η の計算方法を示す. これにより, 各レコードごとの識別推定確率がわかり, テーブルの安全性をより詳細に評価することが可能となる.

[*3] プライバシパラメータとして ϵ-差分プライバシを選んだ場合同時に Pk-匿名性を満たす. 必要に応じて [9] に示す関係式を用いて計算すれば良い

表 1　元テーブル

	attr1	attr2
1	a	A
2	b	B
3	c	C

表 2　匿名化テーブル

	attr1	attr2
1	a	C
2	b	B
3	b	A

4.1　計算方法

攻撃者の背景知識を元テーブル t とすると, 式 (1) は,

$$\eta(t,t',r,r') := \Pr[\Gamma(r)=r'|\Delta(t)=t'\circ\Gamma], \qquad (14)$$

となる. Γ が確率変数なので, そのインスタンス γ となるように式を変形する.

$$\begin{aligned}
&\eta(t,t',r,r')\\
=&\frac{\Pr[\Gamma(r)=r'\wedge\Delta(t)=t'\circ\Gamma]}{\Pr[\Delta(t)=t'\circ\Gamma]}\\
=&\frac{\sum_{\gamma\in\Gamma}\Pr[\Gamma=\gamma]\Pr[\gamma(r)=r'\wedge\Delta(t)=t'\circ\gamma]}{\sum_{\gamma in\Gamma}\Pr[\Gamma=\gamma]\Pr[\Delta(t)=t'\circ\gamma]}\\
=&\frac{\sum_{\gamma(r)=r'}\frac{1}{N!}\Pr[\Delta(t)=t'\circ\gamma]}{\sum_{\gamma}\frac{1}{N!}\Pr[\Delta(t)=t'\circ\gamma]}\\
=&\frac{\sum_{\gamma(r)=r'}\Pr[\Delta(t)=t'\circ\gamma]}{\sum_{\gamma}\Pr[\Delta(t)=t'\circ\gamma]}. \qquad (15)
\end{aligned}$$

$\sum_\gamma \Pr[\Delta(t)=t'\circ\gamma]$ の計算方法を考える. テーブル t の $i\in[N]$ レコードがランダム化によってテーブル t' の $j\in[N]$ レコードとなる確率を a_{ij} とすると,

$$\sum_{\gamma}\Pr[\Delta(t)=t'\circ\gamma]=\sum_{\gamma}\prod_{i\in[N]}a_{i\gamma(i)}, \qquad (16)$$

となる. これは行列式の定義に非常に似ているがパーマネント [19] という別のものである. 行列式の場合は愚直に計算すると $O(N!)$ を必要とするが, $O(N^3)$ で計算可能な方法がある. パーマネントは行列式に似ているものの正確に計算する方法は多項式時間で計算する方法が知られていない. 愚直に計算すると $O(N!)$ 必要とするが, Ryser's Algorithm[20] を用いれば $O(2^N)$ で計算できることが知られており, 計算に近似を許容できれば確率的多項式時間で計算する方法が知られている [10].

4.2　具体的な計算例

具体的なテーブルを例に, 識別推定確率の計算方法を示す. 本節では, PRAM によるランダム化を行った場合を想定したテーブルに対する各レコードごとの識別確率を計算する. 今回対象とするテーブルを表 1, 2 に示す.

また遷移確率行列 $\mathbf{P}_{\text{attr1}}$ および $\mathbf{P}_{\text{attr2}}$ は,

$$\mathbf{P}_{\text{attr1}}=\mathbf{P}_{\text{attr2}}=\begin{pmatrix}0.8&0.1&0.1\\0.1&0.8&0.1\\0.1&0.1&0.8\end{pmatrix} \qquad (17)$$

とする. 今回はシャッフルされた結果が偶然にも元テーブルの並びと一致したとする.

ナイーブな方法 $\eta(t,t',1,1)$ を式 (15) に基づいて計算を行う例を示す. γ による並びの変更がなかった場合, $\Pr[\Delta(t)=t']$ は, 各テーブルの要素が遷移する確率の積で表現できる.

$$\Pr[\Delta(t)=t']=0.8\times0.1\times0.8\times0.8\times0.1\times0.1=0.000512$$

となる. これをシャッフルの全パターン分同様に計算し, 総和をとることで式 (15) の分母が計算可能である. また, 分子はすべてのシャッフルのパターンのうち, $\gamma(1)=1'$ のケースのみ総和を取れば良い.

パーマネントを用いる方法 $\eta(t,t',1,1')$ を行列のパーマネントを用いて計算する方法を示す. まず, ある元テーブルのレコード i と匿名化テーブルのレコード j の遷移確率を a_{ij} として持つ行列 $\mathbf{A}$ を定義する. 例えば元レーブルの 1 番目のレコードと匿名化テーブルの 1 番目のレコードの遷移確率は, a が a となる確率 0.8 及び A が C となる確率 0.1 の積, すなわち 0.08 となる. これらを同様に計算すると,

$$\mathbf{A}=\begin{pmatrix}0.08&0.01&0.08\\0.01&0.64&0.08\\0.08&0.01&0.01\end{pmatrix} \qquad (18)$$

となる. このパーマネントを計算すると,

$$\begin{aligned}
\text{perm}(\mathbf{A}) &= 0.08\times0.64\times0.01+0.01\times0.08\times0.08\\
&+ 0.08\times0.01\times0.01+0.08\times0.64\times0.08\\
&+ 0.01\times0.01\times0.01+0.08\times0.08\times0.01\\
&= 0.004745
\end{aligned}$$

となる (今回は例のために愚直に計算したが, Ryser's Algorithm を用いることで効率良く計算できる).

また分子の計算は行列の余因子展開に似た操作を行うことで計算可能である (ここでは便宜上疑似余因子展開と呼ぶこととする). 行列 $\mathbf{A}$ に対する (i,j) の疑似余因子展開を, A の (i,j) 要素と i 行 j 列を除く $N-1\times N-1$ 行列のパーマネントの積とする. 例えば $r=1,r'=1$ の場合分子は疑似余因子展開を用いることで,

$$0.08\times\text{perm}\begin{pmatrix}0.64&0.08\\0.01&0.01\end{pmatrix}=0.000576 \qquad (19)$$

となる.

これらより, $\eta(t,t',1,1)=0.000576/0.004745=0.12139$ となる. これを同様にすべての r,r' の組み合わせに対して計算し, 1 行目は $r=1$ に対する $r'=1,2,3$ の結果, 2 行目は $r=2$ に対する $r'=1,2,3$ の結果, 3 行目は $r=3$ に対する $r'=1,2,3$ の結果を並べた行列を作成すると

$$\begin{pmatrix}0.121390&0.001896&0.876712\\0.013698&0.971127&0.015173\\0.864910&0.026975&0.108113\end{pmatrix} \qquad (20)$$

となる.

5.　実験

一般化とランダム化をベイズ最適化で組みあわせた手法の有効性を確認するため, 実行速度の比較および有用性の誤差を評価する.

5.1　実験環境

実験に使用した計算機は, CPU : Intel Xeon E5 3.0GHz 8 コア (16 スレッド), Memory : DDR3 128GB, SSD : 1TB, OS : Mac OS X Mojave , JVM : Amazon Correto 1.8 で

表 3　対象データの属性

属性値	取りうる値
名前	5000 種類の名字と 5000 種類の名前とのランダムな掛け合わせ
職業	1,...,24 のいずれかで符号化済み.
性別	男 or 女
住所	5000 種類の住所 (丁目や番地などを除く) とランダムな丁目, 番地のランダムな掛け合わせ
生年月日	1950 年 1 月 1 日以降で, 年-月-日のような記述
購入店舗	A,...,Z の 1 文字
購入日	年-月-日 時-分 で記述 (ある一ヶ月間)
購入カテゴリ	1,...,24 のいずれか
購入金額	1000,...,100000 のいずれか
付与ポイント	0,...,10000 のいずれか

表 4　作成した人工データの一覧

レコード数	属性数	ファイルサイズ	属性ごとの値の平均重複率
1,000,000	100	677.7MB	0.954
10,000,000	100	6.5GB	0.983
50,000,000	100	32GB	0.986

ある.

またプログラムの実装は [29] による実装方式を用いる. ベイズ最適化の実装として, 行列演算ライブラリに Nd4j[*4], 獲得関数の最適化には apache commons math の CMA-ES[*5]を用いた. カーネル関数は Matern5/2 カーネルを用い [17] を用いた.

5.2　対象データ

実装したライブラリの性能を測定するため, データの規模を任意で変更可能な人工データにより評価を行う. 人工データとして, [29] と同様の購買履歴を模したデータを作成した. データの属性及び取りうる値を表 3 に示す. また, 測定対象とするデータについて表 4 に示す.

今回は, QI を職業, 性別, 住所, 生年月日とする. それぞれ一般化階層について, 性別, 生年月日は, 図 2 と同様で, 住所は, レベル 1 が町域, レベル 2 が市区町村, レベル 3 が都道府県という高さ 4 の一般化階層を用意した. 職業に関してはレベル 1 が [1-6],[7-12],[13-18],[19-24], レベル 2 が [1-12], [13-24] となる階層構造を用意した.

5.3　実験結果

ベイズ最適化はランダム性を伴うことから 5 回試行しその平均値を結果として用いる. またベイズ最適化の初期サンプル数は 5 とし, 実行時間の比較では繰り返し回数は 20 回とした.

5.3.1　実行時間の比較

一般化と PRAM を組み合わせた手法は, プライバシ保護指標を $(P)k$-匿名性に限定した場合, ベースは一般化による k-匿名化アルゴリズムであり, (必要に応じて PRAM を適用するかどうかという観点で) 従来の k-匿名化の拡張となっている. そこで数値実験の実行速度の比較では安全性指標として k-匿名性を採用し, 従来の高速に動作する k-匿名化方式と提案方式 (ベイズ最適化を用いた方式) との比較を行う. 比較対象は, 並列深さ優先探索 [29] および ARX Anonymization Tool で採用されている Flash[11] の 2 つの方式を対象とする.

表 5　QI に対するのアルゴリズムの実行時間の比較

レコード	ARX(Flash)	並列深さ優先探索	ベイズ最適化 (提案)
1,000,000	10[s]	16[s]	20[s]
10,000,000	412[s]	230[s]	190[s]
50,000,000	NA	2062[s]	1137[s]

表 6　QI に対するベイズ最適化による匿名化の相対誤差

繰り返し回数	相対誤差
10	10.4%
20	2.5%
50	2.4%

QI を $k = 3$ 匿名を満たすように加工した際の実行速度の比較結果を表 5 に示す. ARX は 100 万レコードの場合, 提案手法および並列深さ優先探索と比べて優位である. しかしながら, 1000 万レコードを超えたあたりで, 並列深さ優先探索およびベイズ最適化の方法が優位となる. 特に 5000 万レコードでは, ベイズ最適化が並列深さ優先探索と比べて 2 倍近く効率が良い. これは, 1 回あたりの有用性評価や一般化処理の計算コストが大きいことから, ベイズ最適化による必要な点に限った探索が効果を発揮していると考えられる.

5.3.2　最適値からの誤差

提案方式が最適値からどの程度誤差が生じているかを相対誤差で評価する. 特にベイズ最適化の繰り返し回数により結果が変わることから, 繰り返し回数を $10, 20, 50$ と変化させたときの最適値からの誤差を相対誤差で評価する. 今回は Pk-匿名性を満たすような条件 (すなわち一般化と PRAM を組み合わせた方式) にし, 並列深さ優先探索と比較を行った. また, 有用性の評価指標として式 (10) を用いて評価を行い, 相対誤差を, ((提案手法の有用性 - 最適値の有用性) / 最適値の有用性) とし評価した.

評価結果を表 6 に示す. 今回, 全パターンの探索を実施する場合, 具体的にはプライバシ保護対象とした属性が 4 つで階層構造の高さがそれぞれ $3, 4, 2, 3$ であり, ランダム化の有無を加えると計 144 パターンの網羅となる. 繰り返し回数 10 回の際は相対誤差が 10%ほど生じたが, 20 回では 2.5%と誤差が減った. 50 回に増やしたところ 2.4%でありほとんど改善が見られなかった. 今回のデータにおいて 20 回程度で十分改善が見られており, それ以上の改善を見込む場合 50 回以上の繰り返しを必要とすることがわかった.

6.　おわりに

本研究では, Pk-匿名性に加え ϵ-差分プライバシを満たす一般化と PRAM を組み合わせた方式を提案した. またランダム化による匿名化処理を施した場合の具体的な識別推定確率を計算する方法を示した.

今後の課題は大きく 2 つある. 1 つ目は, さらなる数値実験による提案手法の有効性の確認である. 差分プライバシを満たす方式, 例えば [28] との比較等により本手法の有用

[*4] https://deeplearning4j.org/docs/latest/nd4j-overview
[*5] https://commons.apache.org/proper/commons-math/

性がどの程度高いかを確認する．今回は一般化し上位概念となったデータと元データと同じ概念との有用性評価の際，式 (9) を用いたがそれ以外の方法も存在しうる．他手法との比較にあたっては，適切な評価方法を検討することも課題となる．

2 つ目は，識別推定確率の計算の効率化である．指数オーダーまで減らしたが，依然として計算量は大きいため多項式時間で計算可能な方式を検討する．

参考文献

[1] 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.

[2] Christopher M Bishop, et al. *Pattern recognition and machine learning*. No. 4. Springer New York, 2006.

[3] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, Vol. 12, No. Mar, pp. 1069–1109, 2011.

[4] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438. IEEE, 2013.

[5] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming*, ICALP'06, pp. 1–12, 2006.

[6] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, Vol. 9, No. 3–4, pp. 211–407, 2014.

[7] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, et al. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association*, Vol. 16, No. 5, pp. 670–682, 2009.

[8] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, pp. 312–317. IEEE, 1996.

[9] Dai Ikarashi, Ryo Kikuchi, Koji Chida, and Katsumi Takahashi. k - anonymous microdata release via post randomization method. In *International Workshop on Security 2015*, pp. 225–241, 2015.

[10] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 08 2002.

[11] Florian Kohlmayer, Fabian Prasser, Claudia Eckert, Alfons Kemper, and Klaus A Kuhn. Flash: efficient, stable and optimal k-anonymity. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pp. 708–717. IEEE, 2012.

[12] Jaewoo Lee and Chris Clifton. How much is enough? choosing ϵ for differential privacy. In *Proceedings of the 14th International Conference on Information Security*, ISC'11, pp. 325–340. Springer-Verlag, 2011.

[13] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 49–60, 2005.

[14] B. Lin, Y. Wang, and S. Rane. A framework for privacy preserving statistical analysis on distributed databases. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 61–66, 2012.

[15] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pp. 24–24. IEEE, 2006.

[16] Adam Meyerson and Ryan Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 223–228. ACM, 2004.

[17] Budiman Minasny and Alex B McBratney. The matérn function as a general model for soil variograms. *Geoderma*, Vol. 128, No. 3-4, pp. 192–207, 2005.

[18] Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pp. 400–404. Springer, 1975.

[19] Thomas Muir and William Henry Metzler. *A Treatise on the Theory of Determinants*. Courier Corporation, 2003.

[20] Herbert John Ryser. Combinatorial mathematics, ser. *The Carus Mathematical Monographs. The Mathematical Association of America*, No. 14, 1963.

[21] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

[22] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*.

[23] 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.

[24] 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.

[25] Leon Willenborg and Peter Kooiman Jose Gouweleeuw. Pram: a method for disclosure limitation of microdata. In *Reseach report*, pp. 90–91, 1997.

[26] Lei Yu, Ling gang Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[27] 五十嵐大, 千田浩司, 高橋克巳. Pl-多様性：属性推定に関する再構築法のプライバシーの定量化. コンピュータセキュリティシンポジウム 2010, pp. 813–818, 2010.

[28] 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の適用. 情報処理学会論文誌, Vol. 58, No. 9, pp. 1483–1500, sep 2017.

[29] 長谷川聡, 正木彰伍, 岡田莉奈. 大規模データを実用的な速度で処理可能な匿名化ライブラリの設計と実装評価. コンピュータセキュリティシンポジウム 2017, 2017.