明治大学総合数理学部

2023 年度

卒　業　研　究

紛失通信とアダマール行列を用いてポイズニング安全性を強化したLDP方式の提案

学位請求者　先端メディアサイエンス学科

清水正浩

目次

第 1 章

序論

1.1 研究背景

近年, スマートデバイスの普及により, サービス事業者はユーザの使用履歴を盛んに収集し, 利活用している. しかし, サービス事業者はユーザの多くのデータを収集するため, ユーザのプライバシーを守ることができないという課題があった.

その課題を解決するために Duchi らによって局所差分プライバシ（Local Differential Privacy Protocol, LDP）が提案された [10]. LDP はユーザが自身のデータにノイズを加えた後に, サービス事業者に送信する. サービス事業者はユーザから送信されたデータに脱ノイズ処理を施し, 集計を行う.

例えば, 2017 年に Apple によって提案された Count Mean Sketch(CMS), Hadamard Count Mean Sketch(HCMS) がある. これらは, 人気な絵文字の集計や Safari のメモリの使用量の集計などに使用されている [5].

しかし, 局所差分プライバシはユーザが局所的にノイズ処理を行うために, 悪意のあるユーザが意図的なデータをサーバに送信して, 集計結果を操作するポイズニング攻撃に対して脆弱であることが Cao らによって指摘されている [1]. 例えば, 2021 年に Cao ら [1] は Pure Protocol という局所差分プライバシ方式のクラスに属する 3 つの局所差分プライバシ方式に対して, 3 種類のポイズニング攻撃, Random Perturb Attack(RPA), Random Item Attack(RIA), Maximal Gain Attack(MGA) を提案し, ポイズニング攻撃に対して脆弱であることを示した. また, 2022 年に Wu ら [2] は PrivKV などの局所差分プライバシ方式に対し, Random Message Attack(RMA), Random Key-Value pair Attack(RKVA), Maximal Gain Attack(M2GA) を提案し, ポイズニング攻撃に対して脆弱であることを示した.

また, HCMS は CMS の通信量を減らすために提案された方式である. CMS が生成するベクトルをサーバに送信する前に, アダマール行列を適用することによって通信量を減らしている.

しかし, CMS と HCMS の有用性の比較は実験的には行われておらず, 通信量を減らす代わりに有用性がどのように異なるのかは明らかにされていない.

1.2 研究方法

本研究では, 2017 年に Apple から提案された局所差分プライバシ方式 Count Mean Sketch(CMS) に対する, 3 つのポイズニング攻撃をする. ポイズニング攻撃に対するロバスト性を向上させるために, CMS に, 紛失通信プロトコルを適用した OT-CMS を提案する. しかし, OT-CMS はハッシュ関数の値域に比例して送

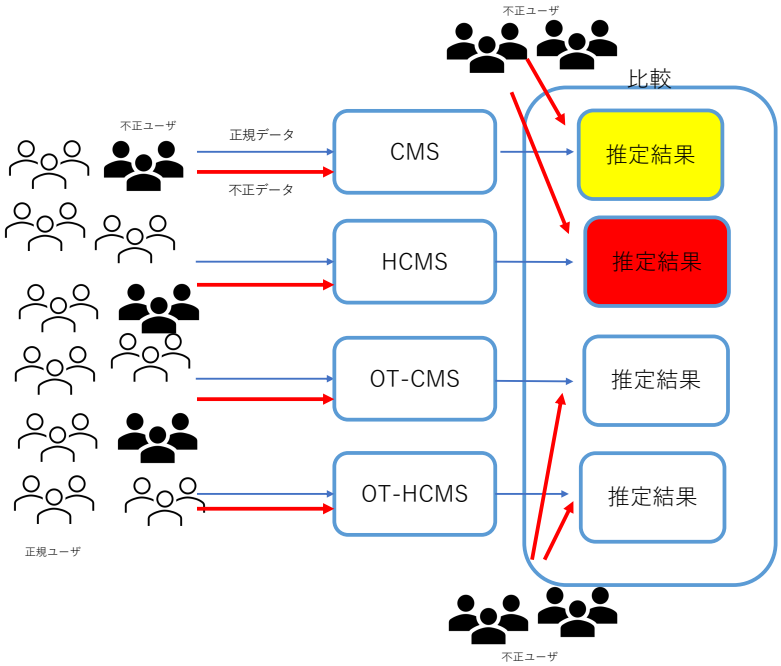

図 1.1　研究概要図

信量と処理コストが増加してしまうという問題点がある．そこで, その問題点を解決するために, Hadamard Count Mean Sketch(HCMS)[5] を導入し, 紛失通信プロトコルの通信コストを削減した OT-HCMS を提案する．本研究の概要を図を 1.1 に示す.

第 2 章

準備と従来研究

2.1 基本定義

使用する記号を表 2.1 に整理する．各ユーザーは自身のプライベートなデータ d にノイズを付与し, 摂動化されたデータ $\tilde{d}$ をサービス事業者に送信する．サービス事業者は各ユーザから収集したデータを集計し, 頻度などを推定する．各ユーザーはデータにノイズを付与する際, ランダマイズアルゴリズム $M(d, \varepsilon)$ を用いることで, プライバシー予算を ε と d を入力とする．

局所差分プライバシーは, 任意の異なる 2 つの入力に対して, ランダマイズアルゴリズム M の出力確率に区別がつかないことを保証する．これにより, サービス事業者は送信されたデータを用いてユーザの真の入力を知ることができず, ユーザのプライバシーを保証する．ランダマイズアルゴリズム M に対して局所差分プライバシーは以下のように定義される．

定義 2.1.1. D を入力の集合, Z を出力の集合と定義する. ランダマイズアルゴリズム M は入力 $d \in D$ を受け取り, $z \in Z$ を出力するとする．この時, 任意の異なる 2 つの入力 $d_1, d_2 \in D$ の出力 $z \in Z$ に対して,

$$Pr(M(d_1, \varepsilon) = z) \leq e^{\varepsilon} Pr(M(d_2, \varepsilon) = z)$$

が成立するとき, ランダマイズアルゴリズム ε-局所差分プライバシを満たすという．

表 2.1 記号一覧

記号	意味
D	アイテムの集合
ε	プライバシー予算
H	ハッシュ関数の集合
k	ハッシュ関数の集合のサイズ
m	ハッシュ関数の値域
n'	不正ユーザの数
β	不正ユーザの割合
T	ターゲットアイテムの集合
t	ターゲットアイテム
r	ターゲットアイテムの数

2.2 Count Mean Sketch

Count Mean Sketch(CMS) は 2017 年に Apple が提案した局所差分プライバシ方式の一つである [5]. CMS はユーザの使用歴を収集し, その頻度を推定する. ユーザとサーバでハッシュ関数を共有する. アルゴリズム 1, 2, 3, 4 に擬似コードを示す [5].

2.2.1 入力

ハッシュ関数の集合を $H = \{h_j | h_j : D \to [m], j \in [k]\}$ とする. 各ユーザは自身のデータ $d \in D$ を H から一様ランダムに取得したハッシュ関数を用い, 次のようにして, m 次元ベクトル $\boldsymbol{v}$ に変換する.

(例 1) あるサービスを使用しているユーザの性別の頻度を推定する場合を考える. $D = \{$“男”, “女”$\}, m = 2, k = 4$ とする. この時, ユーザが $d =$ “男” $\in D$ というデータを持っている場合, j を $\{1, 2, 3, 4\}$ から一様ランダムにサンプリングをして $j = 3 \in \{1, 2, 3, 4\}$ とする. $h_3($“男”$) = 2$ を計算する. 2 次元ベクトル $\boldsymbol{v}$ の 2 番目の要素を 1 に設定し, そのほかの要素を -1 とする. 得られる 2 次元ベクトルは $\boldsymbol{v} = (-1, 1)$ である. $\boldsymbol{v} = (-1, 1)$ に摂動化を施した $\tilde{\boldsymbol{v}}$ をサーバに送信する.

2.2.2 摂動

m 次元ベクトルを $(v_1, ..., v_m)$ とする. $i = 1, ..., m$ について確率 p で真の値 v_i を出力し, 確率 $q = 1 - p$ で $-v_i$ を出力する. すなわち,

$$\tilde{v}_i = \begin{cases} v_i & w./p. \quad p, \\ -v_i & w./p. \quad q. \end{cases}$$

$$p = \frac{e^{\frac{\varepsilon}{2}}}{1 + e^{\frac{\varepsilon}{2}}}, \quad q = \frac{1}{1 + e^{\frac{\varepsilon}{2}}}$$

のとき, ε-局所差分プライバシを満たす.

2.2.3 集計

n 人のユーザからの出力を収集し, 各 $d_i \in D$ の頻度を推定する. CMS は収集したデータを用いて, Sketch Matrix と呼ばれる $k \times m$ の行列を作成する. ユーザから収集したデータの集合を $S = \{(\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}), ..., (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)})\}$, $c_\varepsilon = \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}$ と定義する. この時, $\tilde{\boldsymbol{v}}^{(i)}, k, m$ を用いて, $\tilde{\boldsymbol{x}}^{(i)} = k(\frac{c_\varepsilon}{2}\tilde{\boldsymbol{v}}^{(i)} + \frac{1}{2}\mathbf{1})$ を計算する. $\tilde{\boldsymbol{x}}^{(i)}$ を累積して, Sketch Matrix M を構築する. $i \in [n], \ell \in [k]$ とすると, M は $j^{(i)}$ 行 ℓ 列の要素 $M_{j^{(i)},l}$ に $\tilde{x}_{\ell^{(i)}}$ の累積する. Sketch Matrix M から,

$$\tilde{f}(d) = \left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{l=1}^{k} M_{l,h_l(d)} - \frac{n}{m}\right)$$

として, アイテム d のハッシュエントリを平均化することによって, 頻度推定を行い $\tilde{f}(d)$ を求める.

(例 2) あるサービスを使用しているユーザの性別の頻度を推定する場合を考える. $n = 4, D = \{$“男”, “女”$\}, m = 2, k = 2, \varepsilon = \infty, S = \{((1, -1), 1), ((1, -1), 2), ((-1, 1), 1), ((-1, 1), 2)\}$ とする. この時, $\boldsymbol{x}^{(1)}$: “男

”, $\boldsymbol{x}^{(2)}$: “男”, $\boldsymbol{x}^{(3)}$: “女”, $\boldsymbol{x}^{(4)}$: “女” とし, $\tilde{\boldsymbol{x}}^{(\mathbf{1})}, \tilde{\boldsymbol{x}}^{(\mathbf{2})}, \tilde{\boldsymbol{x}}^{(\mathbf{3})}, \tilde{\boldsymbol{x}}^{(\mathbf{4})}$ は以下のように計算される.

$$\tilde{\boldsymbol{x}}^{(\mathbf{1})} = 2\left(\frac{1}{2}(1,-1)+\frac{1}{2}(1,1)\right) = (2,0)$$

$$\tilde{\boldsymbol{x}}^{(\mathbf{2})} = 2\left(\frac{1}{2}(1,-1)+\frac{1}{2}(1,1)\right) = (2,0)$$

$$\tilde{\boldsymbol{x}}^{(\mathbf{3})} = 2\left(\frac{1}{2}(-1,1)+\frac{1}{2}(1,1)\right) = (2,0)$$

$$\tilde{\boldsymbol{x}}^{(\mathbf{4})} = 2\left(\frac{1}{2}(-1,1)+\frac{1}{2}(1,1)\right) = (0,2)$$

この時, Skecth Matrix は

$$M = \begin{pmatrix} 4 & 0 \\ 2 & 2 \end{pmatrix}$$

と得られる. 従って,

$$\tilde{f}(\text{”男”}) = \frac{2}{2-1}\left(\frac{1}{2}\sum_{l=1}^{2} M_{l,h_l(\text{”男”})} - \frac{4}{2}\right) = 2$$

$$\tilde{f}(\text{”女”}) = \left(\frac{2}{2-1}\right)\left(\frac{1}{2}\sum_{l=1}^{2} M_{l,h_l(\text{”女”})} - \frac{4}{2}\right) = 2$$

Algorithm 1 Count Mean Sketch CMS

Input: $d^{(1)}, d^{(2)}, ..., d^{(n)} \in D^n; \varepsilon, k, m$, Dictinary $\hat{D} \subseteq D$

From a set of three-wise independent hashes mapping D to $[m]$, select a set $\boldsymbol{H} = \{h_1, ..., h_k\}$ of k uniformly at random.

for $i \in [n]$ **do**

$(\tilde{\boldsymbol{v}}^{(i)}, j^{(i)}) \leftarrow A_{client-CMS}(d^{(i)}; \varepsilon, \boldsymbol{H})$.

end for

Construct the sketch M $\leftarrow Sketch - CMS\left((\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}), ..., (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)}); \varepsilon, k, m\right)$.

for $d \in \hat{D}$ **do**

$\tilde{f}(d) \leftarrow A_{server}(d; M, \varepsilon, \boldsymbol{H})$.

end for

Output: $Histgram(\tilde{f}(d) : d \in \hat{D})$.

2.3 Hadamard Count Mean Sketch

HCMS は Apple に提案された CMS の亜種である [5]. CMS は, ユーザは m 次元ベクトルを送信するため, ハッシュ関数の値域の大きさに比例して送信量が大きくなってしまう. その問題点を解決するために, CMS に改良を施したのが HCMS である. HCMS はアダマール行列を適用することによって, 送信量を減らすことが可能になる. また, アダマール行列はユーザとサーバで共通しているものとする.

Algorithm 2 Client-Side $A_{client-CMS}$

Input: Data element: $d \in D$; $\boldsymbol{H}$.

Sample j uniformly at random from $[k]$.

Initialize a vector $\boldsymbol{v} \leftarrow -\mathbf{1} \in \mathbb{R}^m$.

Set $v_{h_j(d)} \leftarrow 1$.

Sample $\boldsymbol{b} \in \{-1, +1\}$, where each b_ℓ is i.i.d. where $Pr[b_\ell = +1] = \frac{e^{\frac{\varepsilon}{2}}}{e^{\frac{\varepsilon}{2}}+1}$

$\tilde{\boldsymbol{v}} \leftarrow (v_1 b_1, ..., v_m b_m)$

Output: $\tilde{\boldsymbol{v}}$, index j

Algorithm 3 Compute Sketch Matrix $Sketch - CMS$

Input: Dataset $D = \{(\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}), ..., (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)})\}$;, privacy budget ε, dimensions k, m.

Set $c_\varepsilon \leftarrow \frac{e^{\frac{\varepsilon}{2}}+1}{e^{\frac{\varepsilon}{2}}-1}$

For each $i \in [n]$ set $\tilde{\boldsymbol{x}}^{(i)} \leftarrow k(\frac{c_\varepsilon}{2}\tilde{\boldsymbol{v}}^{(i)} + \frac{1}{2}\mathbf{1})$.

Initialize $M \in \{0\}^{k \times m}$.

for $i \in [n]$ **do**

  for $\ell \in [m]$ **do**

    $M_{j^{(i)},\ell} \leftarrow M_{j^{(i)},\ell} + \tilde{x}_\ell^{(i)}$

  end for

end for

Output: Sketch Matrix M

Algorithm 4 Server-Side A_{server}

Input: Dataset element $d \in D$;, sketch $M \in \mathbb{R}^{k \times m}$, ε, hashes $\boldsymbol{H}$.

Construct $\tilde{f}$:$D \to \mathbb{R}$ where

$$\tilde{f}(d) = \left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{l=1}^{k} M_{l,h_l(d)} - \frac{n}{m}\right)$$

Output: $\tilde{f}(d)$

2.3.1 入力

ハッシュ関数の集合を $H = \{h_j : D \to [m] : j \in [k]\}$ とおく. 各ユーザは自身のデータ $d \in D$ を H から一様ランダムに取得したハッシュ関数を用いて m 次元ベクトル $\boldsymbol{v}$ に変換する. アダマール行列と積をとり m 次元ベクトル $\boldsymbol{w}$ に変換する. そのベクトル $\boldsymbol{w}$ の中から一様ランダムに 1 ビットを取得し, 摂動化する. その 1 ビットをサーバに送信する.

ここで, アダマール行列は, 以下のように再帰的に定義される.

$$H_1 = \begin{pmatrix} 1 \end{pmatrix},$$

$$H_m = \begin{pmatrix} H_{m/2} & H_{m/2} \\ H_{m/2} & -H_{m/2} \end{pmatrix}$$

(例 3)(例 1) と同様にユーザの性別の頻度を推定する場合を考える．$D = \{\text{“男”}, \text{“女”}\}, m = 2, k = 4$ とする．この時, あるユーザが $d =$ “男” $\in D$ というデータを持っている場合, 次のような処理をする．j を $\{1,2,3,4\}$ から一様ランダムにサンプリングをして $j = 3 \in \{1,2,3,4\}$ とする．$h_3(\text{“男”}) = 1$ とする．2 次元ベクトル $\boldsymbol{v}$ の 1 番目の要素を 1 に, そのほかの要素は例 1 と異なり 0 とする．結果として得られる 2 次元ベクトルは $v = (0,1)$ と表される．

$$\boldsymbol{w} = H_2\boldsymbol{v} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = (1,-1)$$

となり, 2 次元の要素の中から一様に $\ell = 1 \in \{1,2\}$ をサンプリングし, $\tilde{w}_\ell = 1, j = 3, \ell = 1$ をサーバに送信する．

2.3.2 摂動

$\boldsymbol{w}$ の中から一様ランダムに取得された 1 ビットを w とおく．CMS と同様に, 確率 p で真の値 v を出力し, 確率 $q = 1 - p$ で $-v$ を出力する．

$$\tilde{w} = \begin{cases} w & w./p. \quad p, \\ -w & w./p. \quad q. \end{cases}$$

この時,

$$p = \frac{e^\varepsilon}{1+e^\varepsilon}, \quad q = \frac{1}{1+e^\varepsilon}$$

とすれば, ε-局所差分プライバシを満たす．

2.3.3 集計

n 人のユーザからの出力を収集し, 各 $d_i \in D$ の頻度を推定する．ユーザから収集した摂動化データを $S = \left((\tilde{w}^{(1)}, j^{(1)}, \ell^{(1)}), ..., (\tilde{w}^{(n)}, j^{(n)}, \ell^{(n)})\right)$, プライバシー予算 ε, $c_\varepsilon = \frac{e^{\varepsilon/2}+1}{e^{\varepsilon/2}-1}$ と定義する．この時, $\tilde{w^{(i)}}, k, m$ を用いて, $\tilde{x}^{(i)} = kc_\varepsilon \tilde{w^{(i)}}$ を求める．$\tilde{x}^{(i)}$ を用いて, Sketch Matrix M を構築する．$i \in [n], \ell \in [m]$ とすると, M の $j^{(i)}$ 行 $\ell^{(i)}$ 列の要素 $M_{j^{(i)},\ell^{(i)}}$ に $\tilde{x}_{\ell^{(i)}}$ を累積する．Sketch Matrix に, アダマール行列を適用し, 正規化する．

$$\tilde{f}(d) = \left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{\ell=1}^{k} M_{\ell,h_\ell(d)} - \frac{n}{m}\right)$$

各アイテムのハッシュエントリを平均化することによって, 頻度推定を行う．

(例 4)$n = 4, D = \{\text{“男”}, \text{“女”}\}, m = 2, k = 2, \varepsilon = \infty, S = ((1,1,1),(1,1,2),(1,2,1),(1,2,1))$ とする．$x^{(1)}$: 男, $x^{(2)}$: 男, $x^{(3)}$: 男, $x^{(4)}$: 男 とするこの時, $\tilde{x}^{(1)}, \tilde{x}^{(2)}, \tilde{x}^{(3)}, \tilde{x}^{(4)}$ は以下のように計算される．

$$\tilde{x}^{(1)} = 2 \cdot 1 \cdot 1 = 2$$

$$\tilde{x}^{(2)} = 2 \cdot 1 \cdot 1 = 2$$

$$\tilde{x}^{(3)} = 2 \cdot 1 \cdot 1 = 2$$

$$\tilde{x}^{(4)} = 2 \cdot 1 \cdot 1 = 2$$

この時, Skecth Matrix は以下のように表される.

$$M = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 4 & 0 \end{pmatrix}$$

M に, 正規化を施すと,

$$MH_2^{-1} = \begin{pmatrix} 2 & 2 \\ 4 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 4 & 0 \\ 4 & 4 \end{pmatrix},$$

$$\tilde{f}(\text{"男"}) = \left(\frac{2}{2-1}\right)\left(\frac{1}{2}\sum_{\ell=1}^{2} M_{\ell,h_\ell(\text{"男"})} - \frac{4}{2}\right) = 4$$

$$\tilde{f}(\text{"女"}) = \left(\frac{2}{2-1}\right)\left(\frac{1}{2}\sum_{\ell=1}^{2} M_{\ell,h_\ell(\text{"女"})} - \frac{4}{2}\right) = 2$$

このように頻度を推定する, HCMS はあるアイテム d_1 がユーザからサーバに送信されると, d_1 のみではなく, 他のアイテムを増加させる可能性がある.

Algorithm 5 Hadamard Count Mean Sketch HCMS

Input: $d^{(1)}, d^{(2)}, ..., d^{(n)} \in D^n; \varepsilon, k, m$, Dictinary $\hat{D} \subseteq D$

From a set of three-wise independent hashes mapping D to $[m]$, select a set $\boldsymbol{H} = \{h_1, ..., h_k\}$ of k uniformly at random.

for $i \in [n]$ **do**

$(\tilde{w}^{(i)}, j^{(i)}, \ell^{(i)}) \leftarrow A_{client-HCMS}(d^{(i)}; \varepsilon, \boldsymbol{H})$.

end for

Construct the sketch M $\leftarrow Sketch - CMS\left((\tilde{w}^{(1)}, j^{(1)}, \ell^{(1)}), ..., (\tilde{w}^{(n)}, j^{(n)}, \ell^{(n)}); \varepsilon, k, m\right)$.

for $d \in \hat{D}$ **do**

$\tilde{f}(d) \leftarrow A_{server}(d; M, \varepsilon, \boldsymbol{H})$.

end for

Output: $Histgram(\tilde{f}(d) : d \in \hat{D})$

Algorithm 6 Client-Side $A_{client-HCMS}$

Input: Data element: $d \in D; \boldsymbol{H}$.

Sample j uniformly at random from $[k]$.

Initialize a vector $\boldsymbol{v} \leftarrow \{0\}^m$.

Set $v_{h_j(d)} \leftarrow 1$.

Transform $\boldsymbol{w} \leftarrow H_m \boldsymbol{v}$

Sample ℓ uniformly at random from $[m]$.

Sample $b \in \{-1, +1\}$, which is $+1$ with Probability $\frac{e^\varepsilon}{e^\varepsilon + 1}$.

$\tilde{w} \leftarrow b w_\ell$

Output: $\tilde{w}$, index j, index ℓ.

Algorithm 7 Compute Sketch Matrix $Sketch - HCMS$

Input: Dataset $D = \{(\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}, \ell^{(1)}), ..., (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)}), , \ell^{(n)}\}$;, privacy budget ε, dimensions k, m.

Set $c_\varepsilon \leftarrow \frac{e^\varepsilon+1}{e^\varepsilon-1}$

For each $i \in [n]$ set $\tilde{x}^{(i)} \leftarrow kc_\varepsilon \tilde{w}^{(i)}$.

Initialize $M^H \in \{0\}^{k\times m}$.

for $i \in [n]$ **do**

$M_{j^{(i)},\ell^{(i)}} \leftarrow M_{j^{(i)},\ell^{(i)}} + \tilde{x}^{(i)}$

end for

Transform the rows of sketch back: $M^H \leftarrow M^H H_m^T$

Output: Sketch Matrix M^H

2.4 1-out-of-2 Oblivious Transfer

1-out-of-2 Oblivious Transfer[7] は, 受信者は送信者から送られた 2 つの情報のうち片方しか知ることができず, 送信者は受信者に送信した 2 つの情報のうちどの情報を得られたか知ることができないことを保証する 2 パーティの暗号プロトコルである.

2.5 局所差分プライバシに対するポイズニング攻撃

ポイズニング攻撃は, 悪意のあるユーザが意図的なデータをサーバに送信して推定結果を操作する不正行為である. 例えば, オンラインショッピングサービスの場合を考える. オンラインショッピングサービスの運営者はどの商品がよく売れているかを参考にして仕入れる商品を選択する. そのため, 商品を製造しているメーカは自社商品がよく売れていると偽装させることによって利益の向上を狙う動機がある. 不正なデータを送信することによって自社製品の売れ行きを操作する.

各不正ユーザは, 局所差分プライバシ方式を改ざんすることができ, 任意のデータをサーバに送信することができると想定する.

攻撃者は, システム上で $n^{'}$ 人の不正ユーザを操作することができる. 不正ユーザの数を $n^{'}$ とする. 攻撃者が, 操作する r 個のアイテムをターゲットアイテムとし, その集合を $T = \{t_1, t_2, ..., t_r\}$ とする. サーバは n 人の真のユーザと m 人の不正ユーザの出力から統計量を推定する.

n 人の真のユーザのアイテム t に対する頻度の推定値を $\hat{f}_t$, 不正ユーザを含めた $n + n^{'}$ 人のアイテム t に対する頻度の推定値を $\tilde{f}_t$ とする. ポイズニング攻撃による頻度の推定値の変化量を $\Delta\tilde{f}_t = \tilde{f}_t - \hat{f}_t$ とする.

[1] によるポイズニング攻撃には Random Perturb Attack(RPA), Random Item Attack(RIA), Maximal Gain Attack(MGA) がある. RPA は各不正ユーザが摂動されたデータ集合からランダムに一つ選びサーバに送信する攻撃である. RIA は各不正ユーザがターゲットアイテムの中からランダムに一つ選択し, そのデータを定められた方法で正しく摂動し, サーバに送信する. MGA は摂動されたデータを不正ユーザの意図したデータに置換してサーバに送信する. ポイズニング攻撃の概要を 2.1 に示す.

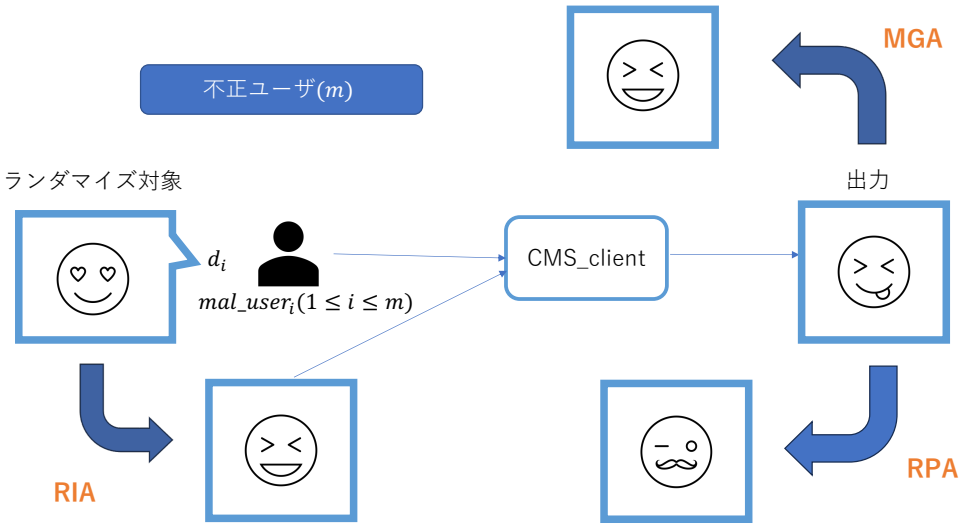

図 2.1　ポイズニング攻撃の概要

第 3 章

提案方式

3.1 CMS と HCMS に対するポイズニング攻撃の評価

3.1.1 Random Perturb Attack

RPA は, 出力をランダムに選択する攻撃である. CMS では, 2^m の数だけ選択肢があり, 各選択肢を $\frac{1}{2^m}$ の確率一つ選びでサーバに送信する. 例えば, $m = 2$ の場合, 不正ユーザは $(1,1),(-1,1),(1,-1),(-1,-1)$ から一様ランダムに取得したデータをサーバに送信する. HCMS の場合では, 1 または-1 を $\frac{1}{2}$ の確率でサーバに送信する.

3.1.2 Random Item Attack

RIA は, 摂動対象を操作する攻撃である. 不正ユーザはランダムに $t \in T$ を選択する. また, 不正ユーザは t を CMS または HCMS を適用し, 得られた出力をサーバに送信する.

3.1.3 Maximal Gain Attack

MGA は, 出力を操作する攻撃である. 不正ユーザは Frequency Gain(FG) を $FG = \sum_{t \in T} \boldsymbol{E}[\tilde{f}_t - \hat{f}_t]$ と定める. FG が最大になるように出力を生成する.

CMS に対する FG の期待値は次のように計算される.

$$
\begin{aligned}
FG &= \sum_{t \in T} \boldsymbol{E}[\hat{f}_t - \tilde{f}_t] \\
&= \sum_{t \in T} \boldsymbol{E}\left[\left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{\ell=1}^{k} \tilde{M}_{\ell,h_\ell(d)} - \frac{n}{m}\right)\right. \\
&\quad \left. - \left(\frac{m}{m-1}\right)\left(\frac{1}{k}\sum_{\ell=1}^{k} M_{\ell,h_\ell(d)} - \frac{n}{m}\right)\right] \\
&= \frac{1}{k}\left(\frac{m}{m-1}\right)\sum_{t \in T} \boldsymbol{E}\left[\sum_{\ell=1}^{k}\left(\tilde{M}_{\ell,h_\ell(d)} - M_{\ell,h_\ell(d)}\right)\right]
\end{aligned}
\tag{3.1}
$$

このとき, $d \in D$ として, i 番目のユーザによる, Sketch Matrix の ℓ 行 $h_\ell(d)$ 列のエントリ $M_{\ell,h_i(d)}$ を $Y_\ell^{(i)}(d)$ とおく. また, i 番目の不正ユーザによる, Sketch Matrix の ℓ 行 $h_\ell(d)$ 列のエントリ $M_{\ell,h_i(d)}$ を $X_\ell^{(i)}(d)$ とお

くと,

$$\begin{aligned}\tilde{M}_{\ell,h_\ell(d)} &= \sum_{i=1}^{n} Y_\ell^{(i)}(d) + \sum_{i=1}^{n'} X_\ell^{(i)}(d) \\ M_{\ell,h_\ell(d)} &= \sum_{i=1}^{n} Y_\ell^{(i)}(d)\end{aligned} \tag{3.2}$$

より, (2) を用いて (1) を変形すると,

$$FG = \frac{1}{k}\left(\frac{m}{m-1}\right)\sum_{t\in T}\boldsymbol{E}\left[\sum_{\ell=1}^{k}\sum_{i=1}^{n'} X_\ell^{(i)}(d)\right] \tag{3.3}$$

これより, $X_\ell^{(i)}(d)$ を最大すればよい. つまり, 攻撃者は H から任意のハッシュ関数 h_j を選択し, $T = \{t_1, t_2, ..., t_r\}$ に対応するハッシュ関数の出力 $h_j(t_1), h_j(t_2), ..., h_j(t_r)$ を調べ, 得られた出力の位置を 1 に設定し, その他は -1 を設定して, $(\boldsymbol{v}, j)$ を送信する. 例えば, $D = \{$“男”, “女”$\}, m = 2, k = 2, H = \{h_j : D \to [2], j \in [2]\}$ のとき, 不正ユーザが男の集計結果を増加させるシナリオを考える. 攻撃者は任意のハッシュ関数 hh_j を選択し, h_j(”男”) $= 0$ であれば, $(1, -1)$ とハッシュ関数番号 j を送信する.

一方, HCMS に対する MGA は複数の戦略が考えられる. 2.4 節で述べたように, あるアイテムの入力が他のアイテムの推定結果に影響を及ぼすことに起因している. ここでは最も簡単な戦略を述べる.

$d \in D$ として, i 番目のユーザによる, Sketch Matrix の ℓ 行 $h_\ell(d)$ 列のエントリ $M_{\ell,h_i(d)}$ を $Z_j^{(i)}(d)$ とおく. また, i 番目の不正ユーザによる, Sketch Matrix の ℓ 行 $h_\ell(d)$ 列のエントリ $M_{\ell,h_i(d)}$ を $\tilde{Z}_j^{(i)}(d)$ とおくと, (1) は次のように変形できる.

$$FG = \frac{1}{k}\left(\frac{m}{m-1}\right)\sum_{t\in T}\boldsymbol{E}\left[\sum_{l=1}^{k}\sum_{i=1}^{n'} \tilde{Z}_l^{(i)}(d)\right] \tag{3.4}$$

この時,

$$\tilde{Z}_l^{(i)}(d) = k \cdot c_\varepsilon \cdot 1 \tag{3.5}$$

とすれば, FG が最大となる. つまり, 攻撃者は $w = 1, l = 0, j$ は任意 を送信すればよい.

3.2 Oblivious Transfer を用いた局所差分プライバシプロトコルの提案

MGA は, 正規の摂動化プロセスの後, 不正ユーザが送信するデータを意図的なものに変更することによって実現される. つまり, 送信されたデータは摂動化が行われていないまま送信される.

そこで, Oblivious Transfer を用いて, ユーザに摂動化を強制させる OT-CMS と OT-HCMS を提案する. 本来, CMS は摂動化した m 次元ベクトル $\tilde{\boldsymbol{v}}$ とハッシュ関数の番号 j を送信するが, OT-CMS においては, ユーザは m 次元ベクトル $\boldsymbol{v}$ の摂動化を単独に行わず, 1-out-of-2 OT を用いてベクトルの要素をサーバの協力により選択する. 従って, 不正な摂動化が防止される. ハッシュ関数の番号は従来通り送信する.

例えば, ユーザが $\boldsymbol{v} = (1, -1)$ をサーバに送信する場合を考える. ユーザが 1 番目の要素 1 を送信するとき, ユーザは $1, -1$ のどちらも OT の送信候補とする. サーバはその内, 真のデータを確率 p, 偽データを確率

$q = 1 - p$ で取得する．このとき，

$$p = \frac{e^{\frac{\varepsilon}{2}}}{1 + e^{\frac{\varepsilon}{2}}}, \quad q = \frac{1}{1 + e^{\frac{\varepsilon}{2}}}$$

この試行を m 回繰り返す．この際, ユーザとサーバは 1-out-of-1/p OT を用いて通信をしているため, サーバはどちらか片方の情報のみ得ることができ, ユーザはサーバがどちらの情報を取得したか知ることができない．

OT-HCMS では，

$$p = \frac{e^{\varepsilon}}{1 + e^{\varepsilon}}, \quad q = \frac{1}{1 + e^{\varepsilon}}$$

に変更し, 試行は 1 ビット分行えば十分である．

Algorithm 8 1-out-of-1/p Oblivious Transfer($OT^1_{1/p}$)

Input: a message v, Probability p

choice a bit $b \in \{0, 1\}$, which is 1 with Probability p.

Sender and Receiver engage in protocol

Receiver learns $\tilde{v}$

Sender learns nothing about b

Output: $\tilde{v}$

3.3 OT-CMS と OT-HCMS の実装

本研究では, python を用いて実装し, Oblivious Transfer で使用する公開鍵は RSA を使用している．ライブラリでは numpy, Python-RSA を使用している．また, 局所差分プライバシ方式や Oblivious Transfer は通信で使われる技術だが, 今回は実際の通信は行わず関数渡しによって擬似実装をしている．アルゴリズム 9, 10, 11, 12, 13 に擬似コードを示す．

Algorithm 9 Oblivious Transfer Count Mean Sketch OT-CMS

Input: $d^{(1)}, d^{(2)}, ..., d^{(n)} \in D^n; \varepsilon, k, m$, Dictinary $\hat{D} \subseteq D$

From a set of three-wise independent hashes mapping D to $[m]$, select a set $\boldsymbol{H} = \{h_1, ..., h_k\}$ of k uniformly at random.

for $i \in [n]$ **do**

$(\boldsymbol{v}^{(i)}, j^{(i)}) \leftarrow A_{client-OT-CMS}(d^{(i)}; \boldsymbol{H})$

$\tilde{\boldsymbol{v}}^{(i)} \leftarrow OT_{CMS}(\boldsymbol{v}^{(i)}, \frac{e^{\frac{\varepsilon}{2}}}{1+e^{\frac{\varepsilon}{2}}})$

end for

Construct the sketch M $\leftarrow Sketch - CMS\left((\tilde{\boldsymbol{v}}^{(1)}, j^{(1)}), ..., (\tilde{\boldsymbol{v}}^{(n)}, j^{(n)}); \varepsilon, k, m\right)$

for $d \in \hat{D}$ **do**

$\tilde{f}(d) \leftarrow A_{server}(d; M, \varepsilon, \boldsymbol{H})$

end for

Output: $Histgram(\tilde{f}(d) : d \in \hat{D})$

Algorithm 10 Client-Side $A_{client-OT-CMS}$

Input: Data element: $d \in D; H$.

Sample j uniformly at random from $[k]$.

Set $v_{h_j(d)} \leftarrow 1$.

Output: $\boldsymbol{v}$, index j

Algorithm 11 OT_{CMS}

Input: a message $\boldsymbol{v}$, Probability p

For each $i \in [m]$ set $\tilde{v}_i \leftarrow OT^1_{1/p}(v_i, p)$.

Output: $\tilde{\boldsymbol{v}}$

Algorithm 12 Oblivious Transfer Hadamard Count Mean Sketch HCMS

Input: $d^{(1)}, d^{(2)}, ..., d^{(n)} \in D^n; \varepsilon, k, m$, Dictinary $\hat{D} \subseteq D$

From a set of three-wise independent hashes mapping D to $[m]$, select a set $\boldsymbol{H} = \{h_1, ..., h_k\}$ of k uniformly at random.

for $i \in [n]$ **do**

$(w^{(i)}, j^{(i)}, \ell^{(i)}) \leftarrow A_{client-HCMS}(d^{(i)}; \varepsilon, \boldsymbol{H})$

$\tilde{w}^{(i)} \leftarrow OT^1_{1/p}(w, \frac{e^\varepsilon}{1+e^\varepsilon})$

end for

Construct the sketch M $\leftarrow Sketch - CMS\left((\tilde{w}^{(1)}, j^{(1)}, \ell^{(1)}), ..., (\tilde{w}^{(n)}, j^{(n)}, \ell^{(n)}); \varepsilon, k, m\right)$

for $d \in \hat{D}$ **do**

$\tilde{f}(d) \leftarrow A_{server}(d; M, \varepsilon, \boldsymbol{H})$

end for

Output: $Histgram(\tilde{f}(d) : d \in \hat{D})$

Algorithm 13 Client-Side $A_{client-HCMS}$

Input: Data element: $d \in D; \boldsymbol{H}$.

Sample j uniformly at random from $[k]$.

Initialize a vector $\boldsymbol{v} \leftarrow \{0\}^m$.

Set $v_{h_j(d)} \leftarrow 1$.

Transform $\boldsymbol{w} \leftarrow H_m \boldsymbol{v}$

Sample ℓ uniformly at random from $[m]$.

Output: w_ℓ, index j, index ℓ.

3.4 CMS と HCMS の送信量の評価

CMS は m に比例して, 送信量が大きくなってしまう. その課題を解決するために考案された局所差分プライバシ方式が Hadamard Count Mean Sketch である. 通信速度 B を $1 \times 10^9 bps$ であると仮定し, 図 3.1 に CMS と HCMS の m に対する送信時間の推移を示す. ここで, 送信時間は $\frac{m}{B}$ と表せる.

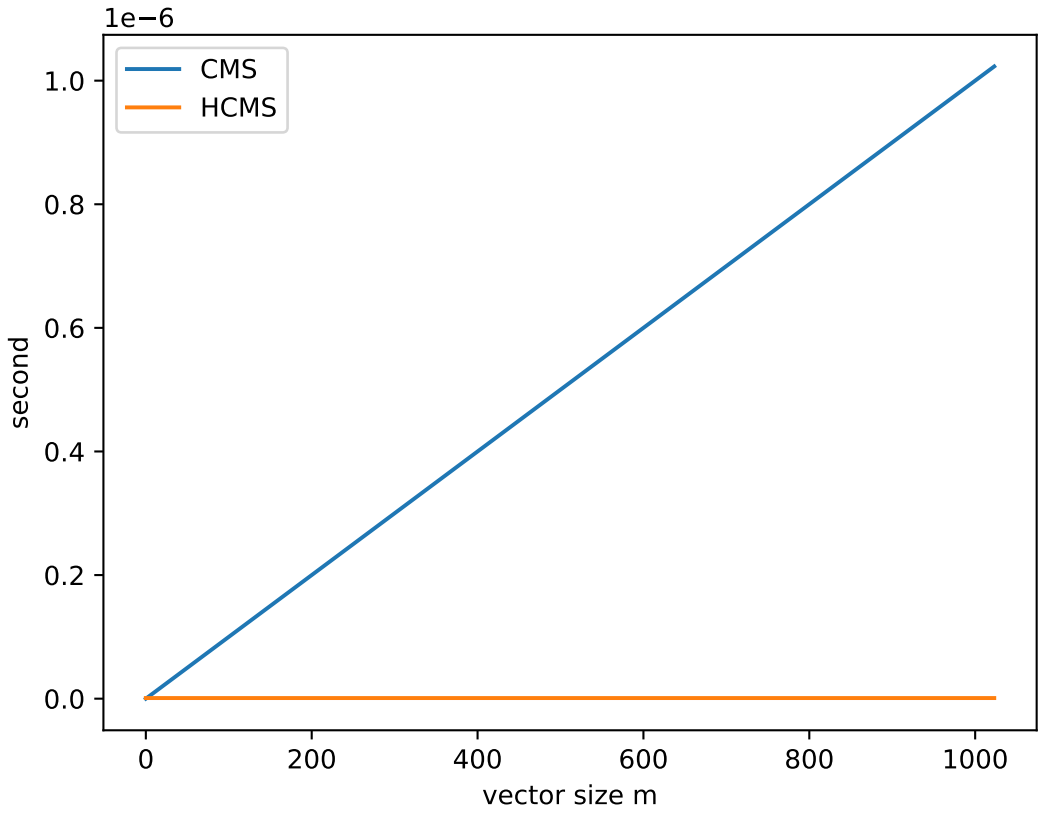

図 3.1　CMS と HCMS の送信時間

第 4 章

実験

4.1 実験目的

アダマール行列を用いて削減する送信量と安全性を評価する．有用性と安全性を用いて評価する．また, 提案手法の安全性と効率を評価する．

4.2 データセット

オンラインショッピングサービスの購入履歴のデータセット [11] を使用する．図 4.1 にアイテムの購入頻度の分布を示す．例えば, 商品 A2 は 3013 件ある．アイテム数は $|D| = 43$ である．

4.3 評価方法

実験で使用するデフォルトパラメータを表 4.1 に整理する．真の分布と推定分布の平均二乗誤差 MSE を用いて有用性を評価する．FG を用いて安全性を評価する．試行を 50 回繰り返しその平均値を評価値とする．ただし, OT-CMS, OT-HCMS の安全性評価では, 試行は 10 回行い, アイテムは “$A18$”, “$A34$” について評価する．MGA ポイズニング攻撃を評価する．OT-CMS と OT-HCMS の効率を OT の処理時間を用いて評価する．

4.4 実験結果

有用性

表 4.1　評価に用いるデフォルトパラメータ

パラメータ	
プライバシー予算 ε	1.0
ベクトル長 m	2^7
ハッシュ関数の数 k	2^{10}
不正ユーザの割合 β	0.01
ターゲットアイテムの数 r	1

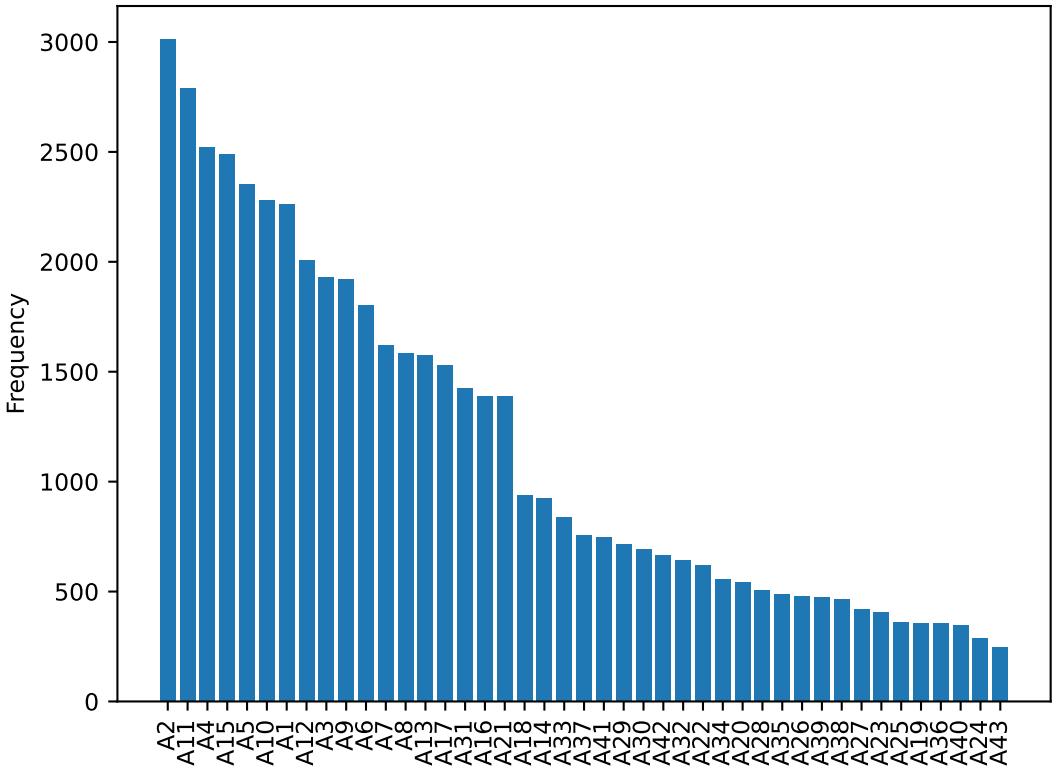

図 4.1　データセットのアイテムの購入頻度の分布

図 4.2, 図 4.3, 図 4.4, 表 4.2 にそれぞれプライバシー予算 ε, CMS の符号化ベクトル長 m, ハッシュ関数の数 k に対する CMS と HCMS の MSE を示す.

プライバシー予算については CMS, HCMS どちらも単調に MSE が減少している. CMS は HCMS に比べ $\varepsilon = 0.1$ の場合を除き, MSE が小さくなっている. 特に $\varepsilon = 1.0$ のとき, 89.7% だけ CMS の方が精度が良い. ベクトル長については全ての場合で CMS の方が HCMS よりも MSE が小さい. 特に $m = 1024$ のとき, 52.8% だけ CMS の方が精度が良い. ハッシュ関数の数についても CMS, HCMS どちらも単調に MSE が減少している.

安全性

図 4.5, 図 4.6, 図 4.7, 表 4.3, 表 4.4, 表 4.5 にそれぞれプライバシー予算 ε, 不正ユーザの割合 β, ターゲットアイテムの数 r に対する CMS と HCMS の FG を示す.

MGA については, ε, β, r の全ての条件で HCMS の方が CMS より FG が平均で 16.0% 小さい. RPA についても同様に, ε, β, r の全ての条件で HCMS の方が CMS より FG が小さい. 特に $beta = 0.1$ の時, 最大で 40.35 小さい. 一方で, RIA では, ε, β, r における FG は CMS と HCMS の差は小さい.

OT-CMS と OT-HCMS の安全性

図 4.8, 4.9, 表 4.6 に不正ユーザの割合に対する OT-CMS と OT-HCMS の FG を示す.

OT-CMS, OT-HCMS のどちらの場合でも, CMS と HCMS よりも FG が小さくなった. 特に, $beta = 0.10$

のとき, OT-CMS は CMS より 267.83 安全であり, OT-HCMS は HCMS にくらべ 211.83 安全であった.

OT-CMS と OT-HCMS の効率

表 4.7 にハッシュ関数のベクトル長 m に対する OT の処理時間 (s) を示す.

全ての条件で OT-HCMS は OT-CMS より処理時間が小さくなった. 特に, $m = 128$ のとき 733.42s の送信時間の差が見られた.

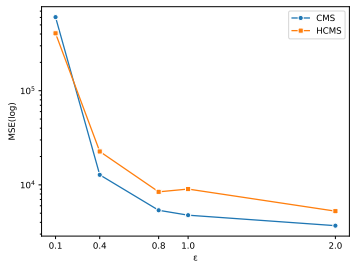

図 4.2　プライバシー予算 ε

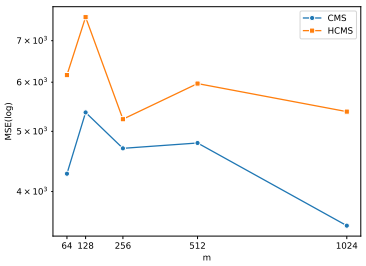

図 4.3　ベクトル長 m

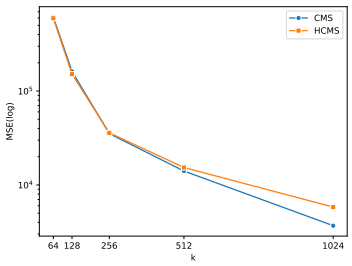

図 4.4　ハッシュ関数の数 k

表 4.2　各パラメータによる MSE($\times 10^3$) の変化

ε	CMS	HCMS
0.1	604.66	407.07
0.4	12.78	22.63
0.8	5.37	8.43
1.0	4.76	9.03
2.0	3.69	5.26

m	CMS	HCMS
64	4.27	6.16
128	5.36	7.64
256	4.70	5.23
512	4.79	5.97
1024	3.52	5.38

k	CMS	HCMS
64	613.03	596.29
128	162.13	151.32
256	35.22	35.89
512	14.07	15.34
1024	3.68	5.81

4.5　考察

4.2 より, HCMS は CMS に比べて誤差が大きく, 有用性が低い. これは, ユーザが行う HCMS の処理に m 次元ベクトルから 1 ビットをランダムにサンプリングすることが原因だと考えられる. HCMS はサンプリングが一様ランダムになるとき CMS と同じ性能となることが Apple により示されている [5]. そのため, 実験的

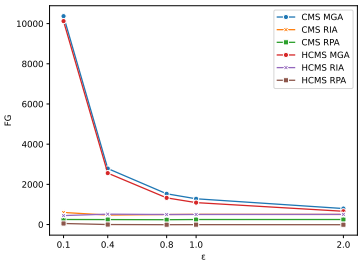

図 4.5　プライバシー予算 ε についての安全性 FG

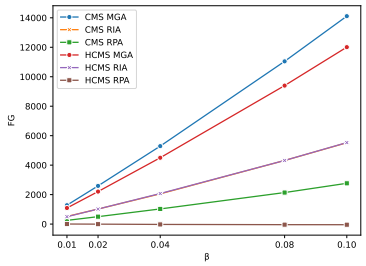

図 4.6　不正ユーザの割合 β についての安全性 FG

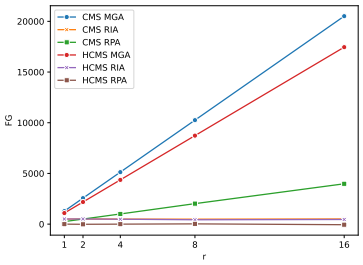

図 4.7　ターゲットアイテムの数 r についての安全性 FG

表 4.3　プライバシー予算 ε による FG($\times 10^2$) の変化

ε	CMS RPA	HCMS RPA	CMS RIA	HCMS RIA	CMS MGA	HCMS MGA
0.1	2.51	0.53	6.01	4.46	103.70	101.24
0.4	2.47	0	4.27	5.17	27.87	25.58
0.8	2.40	−0.08	4.99	4.98	15.31	13.28
1.0	2.50	−0.05	5.08	5.00	12.82	10.91
2.0	2.50	−0.08	5.03	5.04	7.96	6.60

表 4.4　不正ユーザの割合 β による FG($\times 10^2$) の変化

β	CMS RPA	HCMS RPA	CMS RIA	HCMS RIA	CMS MGA	HCMS MGA
0.01	2.51	0.04	5.18	5.01	12.82	10.91
0.02	5.03	−0.05	10.19	10.21	25.92	22.06
0.04	10.28	−0.24	20.52	20.73	52.91	45.03
0.08	21.39	−0.40	43.15	43.18	110.44	93.99
0.10	27.67	−0.43	55.19	55.30	141.11	120.09

表 4.5　ターゲットアイテムの数 r による FG($\times 10^2$) の変化

r	CMS RPA	HCMS RPA	CMS RIA	HCMS RIA	CMS MGA	HCMS MGA
1	2.50	−0.01	4.93	5.04	12.82	10.91
2	4.99	−0.17	5.09	5.07	25.64	21.82
4	10.02	−0.02	4.95	4.84	51.28	43.64
8	20.20	0.25	4.89	4.29	102.55	87.27
16	39.77	−0.58	5.10	4.56	205.11	174.54

表 4.6　OT-CMS と OT-HCMS の不正ユーザの割合 β による FG の変化

β	CMS	OT-CMS	HCMS	OT-HCMS
0.01	38.30	18.56	32.6	17.15
0.02	76.61	30.59	65.19	38.75
0.04	158.33	59.70	134.73	65.34
0.08	331.97	129.44	282.51	135.75
0.10	423.90	156.07	360.74	148.91

に行うとサンプリングにある程度の偏りが生じ, HCMS は CMS に比べ有用性が低下する.

MGA に対して, CMS に比べ HCMS の方が安全であった. その原因はノイズ除去の方法によって引き起こされている.

CMS はユーザから送信されたデータのノイズを除去するときに, $\frac{e^{\frac{\varepsilon}{2}}+1}{e^{\frac{\varepsilon}{2}}-1}$ と積をとる. 一方, HCMS は $\frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}$ と積をとる [5]. ε が同じ値であれば, $\frac{e^{\frac{\varepsilon}{2}}+1}{e^{\frac{\varepsilon}{2}}-1}$ の方が大きな値をとる. そのため, 不正ユーザから MGA を用いて攻撃された際に, 攻撃の効果がより増幅される. RIA についてはほぼ同様の安全性を示した. HCMS の方が CMS よりも安全と言える.

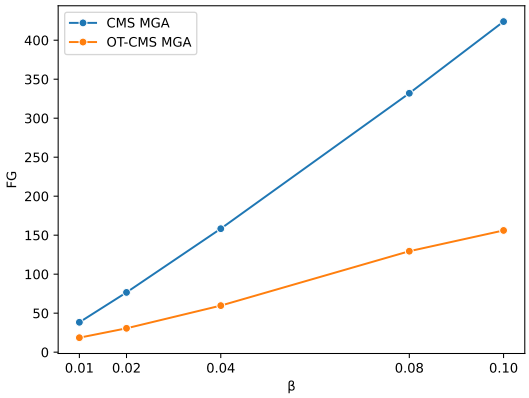

図 4.8　不正ユーザの割合 β についての OT-CMS の安全性 FG

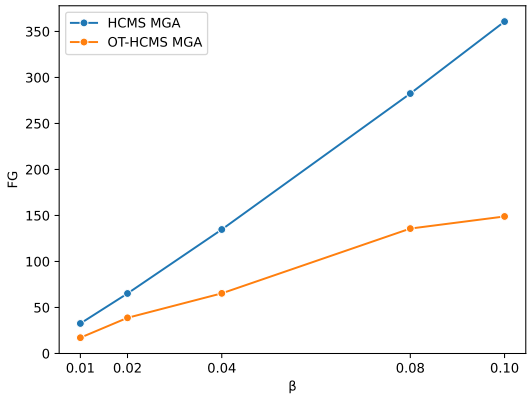

図 4.9　不正ユーザの割合 β についての OT-CHMS の安全性 FG

表 4.7　ベクトル長 m に対する OT の処理時間 (s) の変化

m	OT-CMS	OT-HCMS
8	32.89	3.92
16	61.71	4.54
32	130.40	4.23
64	262.56	3.77
128	737.18	3.76

OT-CMS と OT-HCMS では, 1-out-of-2 OT を適用し, 摂動を強制的に行うため不正ユーザは意図的なデータを送信することができない. FG が小さくなり, 安全性が向上した. また, OT-CMS と OT-HCMS の OT の処理時間を比較すると, OT-HCMS の方が処理時間が小さくなった.

4.6　提案手法の限界

局所差分プライバシー方式は, ユーザのプライバシーを保護するために, サーバに信頼しないモデルとして提案された. 本研究で提案した OT-CMS と OT-HCMS はサーバがユーザのデータを摂動することを補助しているため, サーバをある程度信頼しているモデルである. そのため, 本来の局所差分プライバシー方式の考え方と異なる. 私は実際の利用環境によって使用するモデルを変更するべきだと考える. 例えば, サーバが信用できないような状況やユーザが持つデータがユーザにとってより重要なものであればあるほど本来の局所差分プライバシ方式を使用するべきである. 一方で, サーバの信用度が保証されていたり, ユーザの持つデータが秘匿性が大きなものでなければ, 局所差分プライバシ方式の安全性を高めるために Oblivious Transfer を用いた方式を採用するべきであると考える.

また, サーバは選択ビット b と最終的に得られた摂動化データを利用することによってユーザの真のデータを知ることができる. そのため, データの管理者に選択ビット b を秘匿する必要がある.

第 5 章

まとめ

本稿では, 局所差分プライバシプロトコル Count Mean Sketch(CMS), Hadamard Count Mean Sketch(HCMS) の有用性を比較し, ポイズニング攻撃に対する安全性を評価した. ポイズニング攻撃に対するロバスト性を向上させるために, Oblivious Transfer を適用する手法を提案した. 実験に基づき, 有用性では CMS の方が高く, 安全性では HCMS の方が高いこと分かった. また, OT の処理時間は HCMS の方が小さくなり, CMS より効率が向上した.

参考文献

[1] X. Cao, J. Jia, N. Z. Gong, “Data poisoning attacks to local differential privacy protocols”, USENIX Security Symposium, pp. 947-964, 2021.

[2] Y. Wu, X. Cao, J.Jia,N.Z. Gong, “Poisoning Attacks to Local Differential Privacy Protocols for Key-Value Data”,USENIX Security Symposium, pp. 519-536, 2022.

[3] Hikaru Horigome, Hiroaki Kikuchi, Chia-Mu Yu, “Expectation-Maximization Estimation for Key-Value Data Randomized with Local Differential Privacy”, The 37th International Conference on Advanced Information Networking and Applications (AINA-2023), 2023.

[4] Hikaru Horigome1, Hiroaki Kikuchi and Chia-Mu Yu, Local Differential Privacy Protocol for Key-Value Data Robust against Poisoning Attacks.

[5] Differential Privacy Team, “Learning with Privacy at Scale”.

[6] Moni Naor and Benny Pinkas. 1999. Oblivious transfer and polynomial evaluation. In Proceedings of the thirty-first annual ACM symposium on Theory of Computing (STOC ’99). Association for Computing Machinery, New York, NY, USA, 245-254. https://doi.org/10.1145/301250.301312

[7] Even, Shimo Goldreich, Oded Lempel, Abraham. (1982). A Randomized Protocol for Signing Contracts.. Communications of the ACM. 28. 205-210. 10.1145/3812.3818.

[8] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally differentially private protocols for frequency estimation. In Proceedings of the 26th USENIX Conference on Security Symposium (SEC’17). USENIX Association, USA, 729-745.

[9] Gadotti, Andrea Houssiau, Florimond Annamalai, Meenatchi Montjoye, Yves-Alexandre. (2023). Pool Inference Attacks on Local Differential Privacy: Quantifying the Privacy Guarantees of Apple’s Count Mean Sketch in Practice.

[10] J. C. Duchi, M. I. Jordan and M. J. Wainwright, ”Local Privacy and Statistical Minimax Rates,” 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Berkeley, CA, USA, 2013, pp. 429-438, doi: 10.1109/FOCS.2013.53.

[11] clickstream data for online shopping. (2019). UCI Machine Learning Repository. https://doi.org/10.24432/C5QK7X. rates ”, Foundations of Computer Science, pp. 429-438, 2013.

謝辞

本論文は筆者が明治大学総合数理学部先端メディアサイエンス学科に在学中の研究成果をまとめたものである．本研究を行うにあたり, 多くの方々から多大なご指導を承りました．特に, 明治大学総合数理学部先端メディアサイエンス学科の菊池浩明教授には, 日々の研究活動に関する多くのご援助を賜りました．また, 学校生活を支えて下さった家族に深く感謝申しあげます.

付録 A

匿名化された健康診断と診療履歴の時系列データによる糖尿病罹患予測

A.1　はじめに

近年, 機械学習や AI の発展によりビッグデータの利活用が様々な場面で盛んになっている. なかでも, 健康診断データは病気の罹患を予測する有効な情報と考えられる. 診断結果と傷病の相関を見るために, 従来は, コホート研究が主流である. 例えば, 野田ら [1] は 10 万人について 8 年間の追跡調査を行い, 住民検診の検査結果とその後の脳卒中等による死亡の関係を明らかにした. 対象者の健康状態を追跡し, 詳細に分析することは生活改善や健康施設作りに, 有益な知見を得ることにつながると考える. しかし, 従来のコホート研究は, 考えられる要因を持つ集団（曝露群）と持たない集団（非曝露群）を追跡し、両群の疾病の罹患率または死亡率を比較する方法であり, 追跡時の時間的な推移を考慮することができていないという課題があった. また, 個人情報を含んだデータにはプライバシー保護のための匿名加工技術についても利用・研究されている. 池上ら [3] は 20 万人分の健康診断データと 28 万人分のレセプトデータ, 32 万人分の適用データを使用し, 従来のコホート研究と比較することによって匿名加工情報の有用性を示した. また, 伊藤ら [2] はレセプトデータや健康診断データから得られる個人の身体的特徴や問診表への回答がどの程度一意であるのか調査し, データから個人が識別されるリスクを評価した. しかし, これらの研究では, 各個人の診療履歴を考慮していない. かつ, 身体的特徴量の時系列変化は個人を識別する際に, 有効な情報であると考える. そこで本研究では, あるヘルスケア企業が取得し, 匿名加工情報とした約 230 万人分の健康診断データと約 580 万人分の傷病レセプトデータ（レセプトデータ, 約 900 万人分の基本データを使用し, 診療履歴を考慮していないデータと診療履歴を考慮したデータを比較し, 変化を確認した. また, 診療履歴を考慮した際の一意率への影響を考察する. 本研究の概要を図 A.1 に示す. 該当のデータセットについて, データの特徴量, 分布などを調べ機械学習アルゴリズムを用いて, ある個人が 3 年後までに糖尿病に罹患しているかどうかを予測するモデルを作成する. このオリジナルのデータから作成したモデルをモデル M1 と表す. オリジナルデータに対して時間的な変化を調査しその推移をオリジナルデータに付加したものに対してモデルを作成する. このモデルをモデル M2 と表す. その後, M1 と M2 を評価し, 比較する. 時間的な推移の付加方法とモデルの作成, 評価方法はそれぞれ節 2.4 と節 3.3 に示す. また, オリジナルデータに対してレセプトデータを突合し, ロジスティック回帰を行うことで健康診断データの 種類の特徴量から糖尿に関する統計的に有意な因子を抽出する.

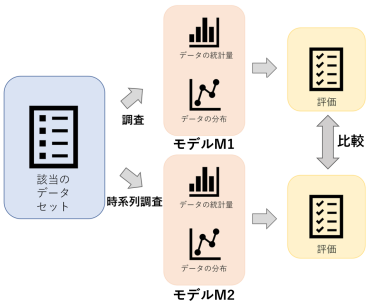

図 A.1　研究概要

A.2　データ

A.2.1　概要

本研究ではあるヘルスケア企業が取得して匿名加工した健康診断データ, 基本データ, 傷病レセプトデータを使用する. 健康診断データは, 被験者 2,345,128 名の体重や身長などの身体的特徴 53 属性と問診結果 50 属性の計 103 属性の 2014 年 4 月から 2021 年 9 月までの健康診断結果から成る. 傷病レセプトデータは, 各被験者が診断された傷病の記録である. また, 基本データには被験者の生年月日や性別を示す.

A.2.2　健康診断データの前処理について

健康診断データには, 分析の障害になる欠損値を含むレコードや相関が高い冗長なデータが含まれている. そこで, 次の前処理を行なった. ここで, 問診結果は十分な睡眠が取れているか否かなど, 生活習慣に関する質問に対する患者の結果の事を差す.

1. 複数のデータの取得方法がある特徴量（Ldl 可視, Ldl 紫外など）を平均値で統合（27 特徴量）.
2. 欠損値レコードの多い 39 特徴量を削除.
3. 多重共線性をなくすために, 相関係数が 0.7 以上ある 2 変数の一方を削除（5 特徴量）.
4. 欠損値を含むレコード（行）の削除.
5. カテゴリカル変数をダミー変数に変更.

処理前後の健康診断データの統計量を表 A.1 に示す.

表 A.1　健康診断データの詳細

	対象年数	被験者数	身体的特徴量数	問診結果数	特徴量数	レコード数	欠損値セル数
処理前	7	2,345,128	55	50	105	7,028,931	439,127,300
処理後	7	172,819	11	17	28	1,858,163	0

本分析では, 表 A.1 に示すデータの 2014 年から 2020 年の健康診断データを使用し, 2017 年のデータを基準とし, 2014 年から 2017 年のデータを診療履歴とする. 2018 年から 2020 年の間のレセプトデータを用いて, 糖尿病に罹患しているか否かを用いて 3 年後までの罹患を予測するモデルを作成する. また, 健康診断データ

に含まれる被験者 id と 受信日, 保健指導レベルコードは削除する.

A.2.3　個人データを用いた特徴量変化の調査

A.2 に可視化したグラフの 1 つの例を示す. 10 名の各特徴量の推移を調査した. すると, hba1c の値 が 6.2 以下から 6.5 以上に変化した年に様々な特徴量の変化が見られた. hba1c は, 糖尿病に罹患しているかしていないかを判定するひとつの特徴量であり, 規準となる平常値は 6.2 %以下で, 罹患と判断する基準値は 6.5 %以上である. まず量的変数について述べていく. 例えば, 10 名中 7 名のデータから, 中性脂肪の値が 1 年で最大 90 mg/dl の増加がみられ, 他の 6 名も大きな増加がみられた. また, bmi は, 10 名中 3 名の増加が確認され, うち一人は, 標準体重の規準範囲を超え, 肥満と分類される 25 以上に増加しているのを確認した. 他にも血清クレアチニンの値が 10 名中 6 名, 約 1.0 mg/dl の減少がみられ, 腹囲実測値に関しては, 10 名中 7 名が 5 から 10 cm ほど増加している. hdl, ldl, gpt についてもいくつか変化が大きいデータがみられた. 質的変数に関しては, 10 名中 5 名が歩行 or 身体活動コードか, 30 分以上運動コードのどちらかについて, 日常の運動の頻度が少なくなるという変化がみられた. さらに, 生活習慣の改善の意識については, 10 名中 4 名がちょうど糖尿病に罹患した年から意識が低くなっていることが確認された. ある個人に関しては糖尿病に罹患したとしに限り, 改善するつもりはないと示し, それまでの年では, 既に改善に取り組んでいるという回答をしていた. 他にも歩行速度が遅くなるというデータが確認された.

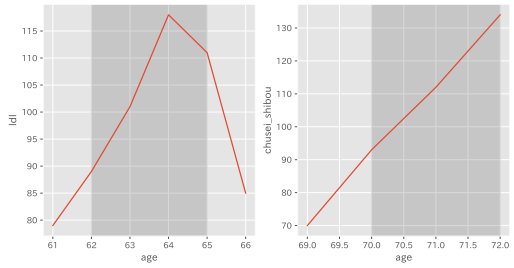

図 A.2　例

A.2.4　時系列情報

本節では, 2.2 節の健康診断データに対して時系列情報の付加をする. ある個人の 2017 年のデータを取り出し, その個人の診療履歴を集計し, 以下の 2 つの方法を用いて時系列情報をデータに付加する.

1.tilt アルゴリズム

平均変化率はある x 年の i 番目の個人の健康診断データのある特徴量 $y_i(x)$ とその直後の $x+1$ の健康診断データについて特徴量 $y_i(x+1)$ の勾配 $\delta y_i = y_i(x+1) - y_i(x)$ の n 人の平均である. 平均変化率積は同様にして積をとったものである. 図 A.3 に tilt アルゴリズムの概略を示す.

2.Linear アルゴリズム

個人 i の特徴量 y に関して, 説明変数を x 年, 目的変数を y として最小二乗法を用いて線形回帰を行い, 回帰係数 a を時系列情報とする. 基準年のデータに付加する. 図**??**に Linear アルゴリズムの概略を示す.
特徴量 y の線形単回帰モデルを

$$y = ax + c$$

とする. ここで説明変数（x）年, a は回帰係数, c は定数項である. 本稿では, この 2 つの方法で定めた特徴量を追加して, 傷病予測モデルを作成し, オリジナルデータとの比較をする.

A.2.5　時系列情報

本節では, 2.2 節の健康診断データに対して時系列情報の付加をする. ある個人の 2017 年のデータを取り出し, その個人の診療履歴を集計し, 以下の 2 つの方法を用いて時系列情報をデータに付加する.

1.tilt アルゴリズム

平均変化率はある x 年の i 番目の個人の健康診断データのある特徴量 $y_i(x)$ とその直後の $x+1$ の健康診断データについて特徴量 $y_i(x+1)$ の勾配 $\delta y_i = y_i(x+1) - y_i(x)$ の n 人の平均である. 平均変化率積は同様にして積をとったものである. 図 A.3 に tilt アルゴリズムの概略を示す.

2.Linear アルゴリズム

個人 i の特徴量 y に関して, 説明変数を x 年, 目的変数を y として最小二乗法を用いて線形回帰を行い, 回帰係数 a を時系列情報とする. 基準年のデータに付加する. 図**??**に Linear アルゴリズムの概略を示す.

特徴量 y の線形単回帰モデルを

$$y = ax + c$$

とする. ここで説明変数（x）年, a は回帰係数, c は定数項である. 本稿では, この 2 つの方法で定めた特徴量を追加して, 傷病予測モデルを作成し, オリジナルデータとの比較をする.

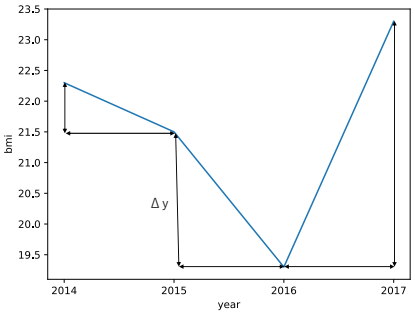

図 A.3　tilt アルゴリズムの概要

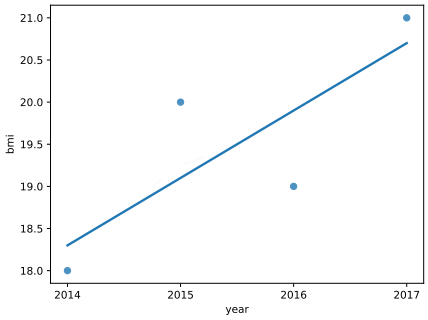

図 A.4　Linear アルゴリズムの概要

A.3　分析

A.3.1　分析方法

本稿では, 次の分析を行う.

1. ロジスティック回帰をし, 時系列情報を考慮したデータの有無が診療結果の統計的有用性にどのように影響するかを調査する.
2. 3 年後までに糖尿病罹患を予測するモデルを作成し, その精度を用いて評価する.
3. 診療履歴を考慮した一意率の調査する.

A.3.2　健康診断と傷病の関係

3 年後までの罹患を目的変数, 健康診断結果を説明変数として, ロジスティック回帰を用いて分析する.
ある被験者 i の 3 年以内の傷病罹患確率 p_i を両者ともに

$$p_i = \frac{1}{1 + e^{-Zi}}$$

で表す. ここで, Z_i は健康診断データから得られる $M = 29(39)$ 種類の説明変数

$$Z_i = \alpha + \beta_1 z_1 + \beta_2 z_2 + \cdots + \beta_M z_M$$

で定められる.
ある特徴量 z_1 について, 他の変数の影響を調整したオッズ比 (adjusted Odds Ratio) は,

$$OR_1 = e^{\beta_1}$$

で与えられる. この分析では, 罹患数が十分に少ない時, オッズ比と相対リスク (Rerative Risk) が等しいことを利用して, 説明変数 z_1 による罹患影響をオッズ比 $OR = Pr(\text{罹患}\,|z_1 = 1)/Pr(\text{非罹患}\,|z_1 = 1)$ で 与える. また, ロジスティック回帰の結果から得られる各説明変数の偏回帰係数に対する標準誤差, Z 値, p 値を分析に用いる. ここ で, ある説明変数を m とすると, z 値は,

$$z\text{ 値}_m = \frac{\text{偏回帰変数}_m}{\text{標準誤差}_m}$$

で表す. また, p 値は有意差 5 %未満であれば, その説明変数 m は目的に対して影響を与えていると判断する.

A.3.3　罹患予測モデル

健康診断データから 3 年後までの糖尿病の罹患予測モデルを作成する. 学習時には罹患者と同数の非罹患者レコードをラ ンダムサンプリングして用いる. 学習アルゴリズムには Multi-layer Parceptron classiffier(MLPclassiffier), Support Vector Machine(SVM) を使用する. 各モデルの評価は, 再現率と適合率の調和平均である F 値を利用する. モデルは python の scikit-learn を用いて実装し, MLPclasiffier のハイパーパラメータにはデフォルト, SVM のハイパーパラメータについてはグリッドサーチを使用する. 2017 年

を基準の年として, 2020 年までに罹患しているかどうか予測する．その際のデータセットを表 A.2 に示す．罹患者と非罹患者について被験者数が大きく異なるので非罹患者から同じ数の非罹患者をサンプリングしてデータの予測を行う．また, サンプリングによる分布の違いを考慮するために, モデル作成を 1000 回行い統計量や分布を比較検証する．表 A.3 にオリジナルデータ, 表 A.4 に tilt アルゴリズムを適用したデータ, 表 A.5 に Linear アルゴリズムを適用したデータの一例を示す.

表 A.2　罹患予測するのに用いるデータセット

	被験者数	属性数	問診結果数
3 年以内に糖尿病に罹患する	209	13	17
3 年以内に糖尿病に罹患しない	209	13	17

A.3.4　一意率について

診療履歴の数に対する安全性への影響を [2] における一意率を用いて観察する．一意率は, すべての個人の集合における，基準年から length 年間における一意な特徴量ベクトルを持つ個人の割合で定める．特徴量ベクトルとは, ある個人 i に対してある特徴量ベクトル x を長さ n の診療履歴のある特徴量 $x_1, x_2, x_3, ..., x_n$ を用いて次のように定義する.

$$\mathbf{x} = (x_1, x_2, x_3, ..., x_n)$$

A.4　分析結果

A.4.1　ロジスティック回帰

表 A.6 にロジスティック回帰を 1000 回行った結果の平均を示す．両者の結果に関して, 全体的にあまり顕著な変化は起きなかった．ただ, 腹囲実測のオッズ比の値が元データが 1.037 なのに対し, 付加後の数値は 1.106 と増加している．また, 拡張期血圧の付加変数のオッズ比が 0.957 と負の傾向を示している．他の付加変数のオッズ比に関しては, 1.0 付近の数値を示している．他には, 服薬 1 血圧コード, 服薬 3 脂質コード, 年齢 (両者ともに, 正または負に大きな傾向を示している．p 値に関しては, 有意か判断する基準値の 0.5 を下回る特徴量は, 両者全体的に少ないが, 比較的付加した方が少し多いといえる．さらに, 先ほどオッズ比が大きいと示した特徴量のうち, 服薬 1 血圧コード, 年齢の二つに関しては, p 値が限りなく低いため, これらの特徴量は有意であることがわかる.

表 A.3　オリジナルデータの一例

属性	値
sex_code	1
bmi	21.5
fukui	78.7
systolic_blood_pressure	119
diastolic_blood_pressure	56
chusei_shibou	90
hdl	70
ldl	142
got	20
gpt	22
gamma_gt	18
kessei_cr	0.75
taijuu_henka_20sai_code	2
undou_shuukan_30pun_code	2
hokou_or_shintai_katsudou_code	2
hokou_sokudo_code	2
tabekata1_hayagui_code	2
tabekata2_shuushinmae_code	1
shokushuukan_code	2
inshu_code	1
suimin_code	1
kitsuen_code	2
seikatsu_shuukan_kaizen_code	3
fukuyaku1_ketsuatsu_code	2
fukuyaku3_shishitsu_code	2
kioureki1_noukekkan_code	2
kioureki2_shinkekkan_code	2
kioureki3_zinfuzen_code	2
hinketsu_code	2
old	36

表 A.4　tilt アルゴリズムを適用したデータの一例

属性	値
sex_code	1
bmi	21.5
fukui	78.7
systolic_blood_pressure	119
diastolic_blood_pressure	56
chusei_shibou	90
hdl	70
ldl	142
got	20
gpt	22
gamma_gt	18
kessei_cr	0.75
taijuu_henka_20sai_code	2
undou_shuukan_30pun_code	2
hokou_or_shintai_katsudou_code	2
hokou_sokudo_code	2
tabekata1_hayagui_code	2
tabekata2_shuushinmae_code	1
shokushuukan_code	2
inshu_code	1
suimin_code	1
kitsuen_code	2
seikatsu_shuukan_kaizen_code	3
fukuyaku1_ketsuatsu_code	2
fukuyaku3_shishitsu_code	2
kioureki1_noukekkan_code	2
kioureki2_shinkekkan_code	2
kioureki3_zinfuzen_code	2
hinketsu_code	2
old	36
bmi_tilt_ave	0.0049
bmi_tilt_product	0.0049
chusei_shibou_tilt_ave	0.087
chusei_shibou_tilt_product	0.087
diastolic_blood_pressure_tilt_ave	-0.021
diastolic_blood_pressure_tilt_product	-0.021
systolic_blood_pressure_tilt_ave	0.03
systolic_blood_pressure_tilt_product	0.03
hdl_tilt_ave	0.055
hdl_tilt_product	0.055
ldl_tilt_ave	0.15
ldl_tilt_product	0.15
got_tilt_ave	-0.0055
got_tilt_product	-0.0055
gpt_tilt_ave	0.0055
gpt_tilt_product	0.0055
gamma_gt_tilt_ave	0.011
gamma_gt_tilt_product	0.011
kessei_cr_tilt_ave	-0.00014
kessei_cr_tilt_product	-0.00014

表 A.5　Linear アルゴリズムを適用したデータの一例

属性	値
sex_code	1
bmi	21.5
fukui	78.7
systolic_blood_pressure	119
diastolic_blood_pressure	56
chusei_shibou	90
hdl	70
ldl	142
got	20
gpt	22
gamma_gt	18
kessei_cr	0.75
taijuu_henka_20sai_code	2
undou_shuukan_30pun_code	2
hokou_or_shintai_katsudou_code	2
hokou_sokudo_code	2
tabekata1_hayagui_code	2
tabekata2_shuushinmae_code	1
shokushuukan_code	2
inshu_code	1
suimin_code	1
kitsuen_code	2
seikatsu_shuukan_kaizen_code	3
fukuyaku1_ketsuatsu_code	2
fukuyaku3_shishitsu_code	2
kioureki1_noukekkan_code	2
kioureki2_shinkekkan_code	2
kioureki3_zinfuzen_code	2
hinketsu_code	2
old	36
bmi_coef	1.8
chusei_shibou_coef	32
diastolic_blood_pressure_coef	-8
systolic_blood_pressure_coef	11
hdl_coef	20
ldl_coef	54
got_coef	-2
gpt_coef	2
gamma_gt_coef	4
kessei_cr_coef	-0.05

A.4.2　ニューラルネットワークによるモデル

表 A.7 に 1000 回, 各アルゴリズムを用いて付加した特徴量を加えて作成した際の F 値の統計量を示す. 図 A.5 には F 値の分布を示す. 最も F 値の平均値, 中央値, 最小値, 最大値が高いのは Linear アルゴリズムを適用したデータであり, その次に, tilt アルゴリズムを適用したデータである. それに加え, 標準偏差も最も小さいのは Linear アルゴリズムを適用したデータであり, その次に, tilt アルゴリズムを適用したデータである.

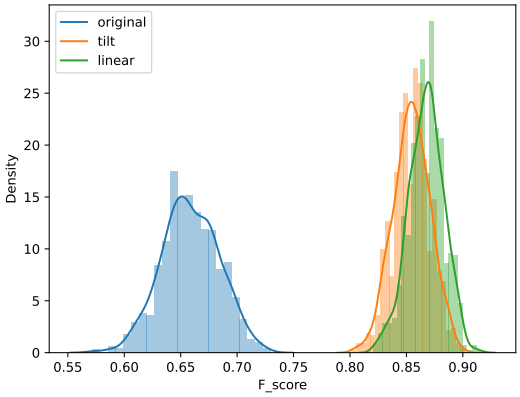

図 A.5　ニューラルネットワークにおける F 値の分布

A.4.3　サポートベクトルマシンによるモデル

表 A.9 に 1000 回, 各アルゴリズムを用いて付加した特徴を加えて作成した際の F 値の統計量を示す. 図 A.7 には F 値の分布を示す. 最も F 値の平均値, 中央値, 最小値, 最大値が高いのは Linear アルゴリズムを適用 したデータであり, その次に, tilt アルゴリズムを適用したデータである. それに加え, 標準偏差も最も小さいのは Linear アルゴリズムを適用したデータであり, その次に, tilt アルゴリズムを適用したデータである.

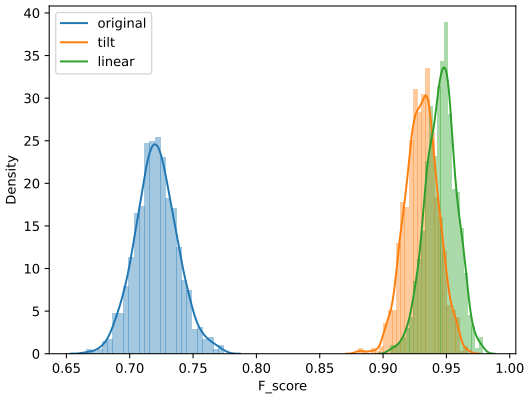

図 A.6　SVM における F 値の分布

A.5　サポートベクトルマシンによるモデル

表 A.9 に 1000 回, 各アルゴリズムを用いて付加した特徴を加えて作成した際の F 値の統計量を示す. 図 A.7 には F 値の分布を示す. 最も F 値の平均値, 中央値, 最小値, 最大値が高いのは Linear アルゴリズムを適用 したデータであり, その次に, tilt アルゴリズムを適用したデータである. それに加え, 標準偏差も最も小さいのは Linear アルゴリズムを適用したデータであり, その次に, tilt アルゴリズムを適用したデータである.

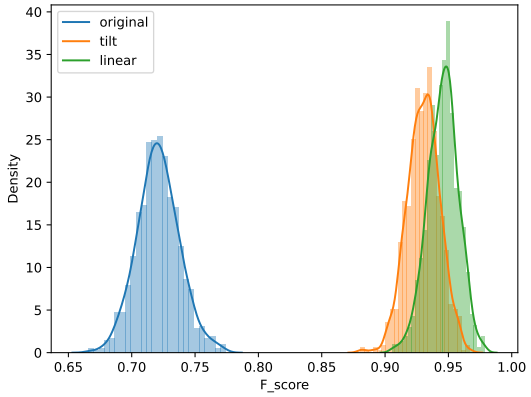

図 A.7　SVM における F 値の分布

A.5.1　診療履歴を考慮した一意率

表 A.10 に履歴期間 length に対応した一意率の平均を示す. 各属性は診療履歴の個数を表している. 診療履歴が長くなると, 一意率が単調に増加していく. また, bmi や血圧などの身体的特徴量に関しては length が 5 以上になると全ての身体的特徴量で 1.0 となり, 問診票などのカテゴリカル変数に関しては一意率が 1.0 となるようなものはなく, length が 2 のときは全てのカテゴリカル特徴量で一意率が 0 となった.

A.6 考察

A.6.1 診療履歴を考慮した特徴量変化について

特徴量変化の調査について, 本調査では, 様々な特徴量の変化が確認された. 特に, 中性脂肪や bmi, 腹囲実測値などの肥満に 関する特徴量が多く変化していることが確認された. この結果の原因として, [4] によると, 肥満は, インスリン の働きを鈍くし, すると膵臓の β 細胞などがインスリンを多く作ろうと働き続け, 体内のインスリン量も増加する. そして, 膵臓が疲れ, 機能 が低下し, 血糖値が上がり hba1c の数値が上がるということがこの結果の原因である. なので, 肥満に関わる特徴量が変化すると糖尿病の有無にかかわるということである. また, 歩行 or 身体コードや, 30 分以上運動コ ード, 生活習慣改善コードは, 運動不足に関わり, 肥満の原因となるため, 間接的に bmi などの特徴量と同じ原因であるといえる. hdl コレステロール, ldl コレステロール, gpt に関しては, 脂肪肝 (中性脂肪が肝臓に蓄積する病気) の原因となる特徴量のため [5], 肥満と同じと考える. さらに, 血清クレアチニンは [6] によると, 血清クレアチニンの低値と 2 型糖尿病は関連があるとされている. そのため, 多くのデータから血清クレアチニンの変化が確認されたと考える. 最後に歩行速度が遅くなっている件については, 高血糖により歩幅が減少することが歩行速度の低下に繋がるので変化がみられたのではないかと考える. [7]

A.6.2 ロジスティック回帰について

ロジスティック回帰について, 本研究では, ロジスティック回帰から糖尿病と各特徴量の相関を調べ, 時系列の考慮の有無を比較した. まず, 腹囲実測値のオッズ比の値に差があることに関しては, 付加後の p 値が 0.362 なので有意な結果を示している. ただし, 腹囲実測値の回帰係数のオッズ比は 0.992, p 値は 0.698 なので偶然このような結果になったのではないかと考える. 腹囲実測の測定は腸管内ガスの影響を受けやすいため, 数値の変動が大きいというのも考えられる. 次に, 拡張期血圧のオッズ比が低いことに関して, 糖尿病にかかりやすい高齢者は, 収縮期血圧が高くなりやすいのに対して拡張期血圧は低くなりやすいというのがオッズ比の値が低くなる理由だと考える. 服薬コードに関しては, 血圧, 脂質の両方がいずれも糖尿病に関わる要素だといえるため大きな値が得られたのだと考える. 最後に, 年齢に関しては, 年齢を重ねるにつれ糖尿病になりやすくなるという事を示している.

A.6.3 罹患予測モデルについて

罹患予測モデルについて, 本稿では診療履歴を tilt アルゴリズムと Linear アルゴリズムを用いて考慮したデータセットを作成し, オリジナルデータと比較することで診療履歴を考慮したデータセットの有用性を評価した. ニューラルネットワークと SVM どちらのアルゴリズムを用いたとしても診療履歴を考慮したデータセットの方が高い精度を出すことがわかった. さらに, 診療履歴を考慮したデータセットのほうが標準偏差も小さく, 安定性が高かった. これは時系列を考慮したデータが特徴量として有用であり, モデルが識別しやすい特徴量であるということを示していると考える. 次に tilt アルゴリズムと Linear アルゴリズムの比較をすると, Linear アルゴリズムの方が全ての代表値で優秀な結果を示した. これは tilt アルゴリズムがかなり不安定なアルゴリズムであるからだと考える. 特に, tilt アルゴリズムの平均変化率積に関しては, 特徴量の変化量が 0 の履歴期間があると, それ以外の診療履歴の値に関わらず, 平均変化率積の値が 0 になってしまい特徴量と

してはかなり不安定である．最後に, そのような不安定な tilt アルゴリズムがオリジナルデータよりも高い精度を出しているのかについて考察する．不安定とはいえど, オリジナルデータよりも詳細な情報を持っているため, 高くなるのは不思議ではないと考える．また, 糖尿病罹患者と糖尿非罹患者はどちらか一方は平均変化率が変化しないことが多くあり, 今回の健康診断データにて平均変化率積が有効な値になったと考える．

A.6.4　診療履歴を考慮した一意率の推移について

一意率について, 本稿では診療履歴の長さによってどのように特徴量の一意率が変化するのかを特徴量ベクトルを定義して調査した．やはり, 診療期間が大きくなればなるほどに一意率は上昇していった．これは特徴量ベクトルの長さが大きくなることによって特徴量ベクトルが一致する確率が小さくなったからだと考える．また, 身体的特徴量とカテゴリカル変数を比較したときに身体的特徴量の方が一意率が高くなる傾向があった．身体的特徴量は連続値であり, カテゴリカル変数は離散値であり値の種類が高々 3 つしかないことからこのような傾向があったと考える．

A.7　終わりに

本研究からは, 様々な結果が得られた．まず, 特徴量変化の調査については, 中性脂肪や腹囲実測値など, 主に肥満の原因となる特徴量の変化が多く確認された．次に, ロジスティック回帰の結果から得られる健康診断と傷病の関係については, 顕著な結果はでなかったが, 付加変数を加えたデータセットのほうが元データセットよりも全体的にやや有意な統計値を得るということがわかった．

作成した罹患予測モデルでは, ニューラルネットワークにてオリジナルデータと tilt アルゴリズムを適用したデータでは平均 F 値の差が 0.196 であった．またオリジナルデータと Linear アルゴリズムを適用したデータでは平均 F 値の差が 0.208 であった．また, サポートベクトルマシンにて同様に比較すると, 平均 F 値の差がそれぞれ 0.209, 0.229 であった．それぞれのモデルで診療履歴を考慮したデータの方が F 値は大きくなった．

一意率については, 履歴期間が大きくなると一意率が増加していくことがわかった．

今後の課題として, tilt アルゴリズムと Linear アルゴリズムは不安定なアルゴリズムであるため, より安定した診療履歴を考慮するアルゴリズムの開発と診療履歴の履歴期間が大きくなることで一意率が上昇し, 匿名性が失われてしまうという問題点を解消することが挙げられる．

表 A.6　ロジスティック回帰結果

	odds_ratio		std_err		zvalue		pvalue	
	original	changed	original	changed	original	changed	original	changed
sex_code	1.003	1.006	0.043	0.047	0.058	0.125	0.746	0.708
bmi	1.006	0.962	0.089	0.100	0.062	-0.385	0.711	0.643
fukui	1.037	1.106	0.091	0.100	0.393	0.988	0.645	0.362
diastolic_blood_pressure	1.040	1.056	0.060	0.076	0.658	0.719	0.526	0.478
systolic_blood_pressure	0.998	0.965	0.060	0.074	-0.047	-0.498	0.728	0.560
chusei_shibou	1.033	1.025	0.041	0.055	0.773	0.449	0.465	0.614
hdl	0.973	0.975	0.044	0.051	-0.616	-0.493	0.548	0.574
ldl	0.986	0.993	0.037	0.042	-0.395	-0.185	0.639	0.676
got	1.008	1.009	0.039	0.051	0.195	0.179	0.747	0.697
gamma_gt	1.033	1.024	0.042	0.059	0.764	0.395	0.471	0.633
kessei_cr	1.004	0.989	0.043	0.051	0.083	-0.213	0.788	0.732
taijuu_henka_20sai_code	0.981	0.996	0.041	0.046	-0.463	-0.099	0.616	0.698
undou_shuukan_30pun_code	1.012	1.006	0.039	0.043	0.312	0.131	0.664	0.723
hokou_or_shintai_katsudou_code	1.006	0.994	0.038	0.041	0.144	-0.139	0.720	0.713
hokou_sokudo_code	1.017	1.013	0.035	0.039	0.476	0.342	0.611	0.701
tabekata1_hayagui_code	1.008	1.004	0.035	0.039	0.226	0.094	0.693	0.710
tabekata2_shuushinmae_code	1.022	1.019	0.037	0.040	0.589	0.472	0.563	0.59
shokushuukan_code	0.984	0.998	0.036	0.039	-0.454	-0.053	0.621	0.77
inshu_code	1.034	1.037	0.038	0.042	0.865	0.862	0.422	0.453
suimin_code	0.984	0.976	0.035	0.038	-0.455	-0.659	0.625	0.539
kitsuen_code	1.002	0.995	0.037	0.040	0.041	-0.142	0.726	0.688
seikatsu_shuukan_kaizen_code	1.009	1.002	0.037	0.041	0.242	0.053	0.680	0.741
fukuyaku1_ketsuatsu_code	0.899	0.906	0.040	0.045	-2.636	-2.193	0.017	0.056
fukuyaku3_shishitsu_code	0.966	0.970	0.038	0.044	-0.932	-0.697	0.393	0.517
kioureki1_noukekkan_code	0.993	0.984	0.035	0.039	-0.215	-0.419	0.695	0.636
kioureki2_shinkekkan_code	0.998	0.994	0.035	0.039	-0.054	-0.155	0.688	0.662
kioureki3_zinfuzen_code	1.016	1.013	0.021	0.025	0.386	0.339	0.528	0.524
hinketsu_code	1.000	0.987	0.037	0.040	-0.016	-0.345	0.737	0.642
age	1.116	1.139	0.039	0.043	2.787	3.036	0.011	0.005
bmi_coef		0.980		0.042		-0.492		0.621
fukui_coef		0.992		0.060		-0.129		0.698
chusei_shibou_coef		1.027		0.050		0.544		0.540
diastolic_blood_pressure_coef		0.957		0.057		-0.774		0.482
systolic_blood_pressure_coef		1.018		0.056		0.295		0.662
hdl_coef		1.026		0.043		0.586		0.566
ldl_coef		0.983		0.041		-0.421		0.631
got_coef		0.996		0.050		-0.084		0.711
gamma_gt_coef		1.006		0.054		0.127		0.655
kessei_cr_coef		1.015		0.047		0.336		0.707

表 A.7　ニューラルネットワークを用いた時の F 値の統計量

	平均	中央値	標準偏差	最小値	最大値
original	0.659	0.658	0.0256	0.572	0.730
tilt	0.855	0.854	0.0165	0.801	0.902
linear	0.867	0.868	0.0156	0.821	0.916

表 A.8　SVM を用いた時の F 値の統計量

	平均	中央値	標準偏差	最小値	最大値
original	0.721	0.721	0.0168	0.666	0.774
tilt	0.930	0.931	0.0128	0.881	0.967
linear	0.950	0.950	0.0120	0.906	0.979

表 A.9　SVM を用いた時の F 値の統計量

	平均	中央値	標準偏差	最小値	最大値
original	0.721	0.721	0.0168	0.666	0.774
tilt	0.930	0.931	0.0128	0.881	0.967
linear	0.950	0.950	0.0120	0.906	0.979

表 A.10　履歴期間 length における一意率の推移

length	2	3	4	5	6	7
unique rate	0.262	0.424	0.437	0.687	0.783	0.795

参考文献

[1] 野田博之, 磯博康 西連地利己, 入江ふじこ 深澤伸子, 鳥山佳則, 大田仁史, 能勢忠男, 住民健診 (基本健康診査) の結果に基づいた脳卒中・虚血性心疾患・全循環器疾患・がん・総死亡の予測, 日本公衛誌 53: 265277, 2006.

[2] 伊藤聡志，池上和輝，菊池浩明匿名加工情報の応用 (1): 健康診断データとレセプトデ ータの分析とプライバシーリスク評価

[3] 伊藤聡志，池上和輝，菊池浩明匿名加工情報の応用 (2): 各種傷病を予測する健康診断 モデル 2020

[4] 門脇孝, 肥満症と糖尿病, 日本内科学会雑誌第 100 巻第 4 号, pp.939-944, 2011.

[5] 奥田昌恵, et al. , 2 型糖尿病における脂肪肝-その頻度及び臨床的特徴-, 糖尿病 50 巻 8 号, pp.631-634, 2007.

[6] 針田信子, et al. , Lower Serum Creatinine Is a New Risk Factor of Type 2 Diabetes, Diabetes Care 誌, pp.424-426, 2009.

[7] 大関直也, 水上昌文, 糖尿病神経障害者の歩行速度特性, 理学療法科学 33(1), pp. 89-93, 2018

付録 B

分担表

作業の分担を B.1 に示す.

表 B.1　ロジスティック回帰結果

	清水	石山
データ前処理	ヘルスケアデータの整形	データの集計
時系列情報の付加	アルゴリズム作成, プログラムの実装	-
ロジスティック回帰	-	健康診断と傷病の関係を調べるプログラムの実装
特徴量変化の調査	-	該当個人データの取得, 調査
罹患推定モデル	罹患推定モデルの作成	-