2B5-OS-15b-4

垂直分割モデルにおけるプライバシ保護ロジスティックス回帰分析

Privacy-preserving Logistic Regression Analysis for Vertically Partitioned Data

呉　双[*1]　川崎　将平[*1]　菊池　浩明[*2]　佐久間　淳[*1*3]
Shuang Wu　Shohei Kawasaki　Hiroaki Kikuchi　Jun Sakuma

[*1]筑波大学 大学院システム情報工学研究科 コンピュータサイエンス専攻
Dept. of Computer Science, Graduate school of SIE, University of Tsukuba

[*2] 明治大学総合数理学部
School of Interdisciplinary Mathematical Sciences, Meiji University

[*3]科学技術振興機構 CREST
Japan Science and Technology Agency, CREST

Logistic regression model is one of the most widely used statistical analysis methods in various fields. In this work, we propose a privacy-preserving logistic regression using stochastic gradient descent (SGD) under cryptographic notion when data are vertically partitioned among different parties. Our method makes use of polynomial approximation to handle the computation of the logistic function. In addition, we adopt a protocol to securely remove the factor from encrypted values. This is used to avoid overflow after a number of SGD updates. Thus, the feasibility of our method is ensured for large scale data sets.

1. はじめに

機械学習でプライバシ保護を実現するために, ランダム化と暗号プロトコルを用いる方法が多く用いられる. ランダム化ではデータにノイズを加えるために精度の劣化を引き起こし得るが, 計算処理の効率性を保証する. Fienberg らはランダム化を利用して, 水平分割されたデータベースに対する安全なロジスティック回帰を行う方法を提案した [1]. しかし, 回帰係数を求める際に用いられる安全な総和計算と安全な行列共有には多くのコストがかかる. さらに, この手法ではパーティーの保持するデータベースの秘密は保護されるが, データベース中の個人の秘密は必ずしも保護されない.

ランダム化と異なり, 暗号プロトコルに基づく手法は対象となるアルゴリズム毎に設計され, 暗号処理を行うために計算効率は良くないが, データにノイズを加えないのでより良い精度を保証する. 加えて, ランダム化における統計的安全性に基づくプライバシ保護に代わり, 暗号プロトコルでは計算量的安全性に基づくプライバシ保護が可能である. しかし, 暗号上では非線形関数を扱えないために, ロジスティックモデルで用いるロジスティック関数 (シグモイド関数) が評価できず, 暗号プロトコル上でロジスティックモデルの設計を行うことは困難である. 近年, Hall らは暗号上で安全にロジスティック回帰を行う方法を提案している [3]. Hall らの提案では, 累積分布関数 (CDF) の経験近似とテイラー近似によってロジスティック関数を近似している. ロジスティック分布の累積分布関数を使う場合ではロジスティック関数の近似は良い近似精度を得られるが, 経験近似のサンプル数 L が比較的小さい場合でも、この近似を評価するための計算量が大きくなる点に問題がある. テイラー近似を用いる方法では, 更新の度に近似値を再計算する必要があること, 更新処理に用いられる安全な逆行列計算も高い計算コストが必要である.

本稿は, 垂直分割データに対する確率的勾配法を用いた暗号的に安全なロジスティック回帰を提案する. 本研究で用いるプライバシモデルは [3] と同様であり, 暗号化された値に対して特定の演算が可能な準同型暗号を用いることで, それぞれのパーティーのデータのプライバシを保護する. 暗号上でプライバシ保護ロジスティック回帰を達成するために解決しなければならない問題として, 以下が挙げられる. (1) ロジスティック関数が非線形性を持つために, 直接暗号上で計算することができない. (2) ロジスティック回帰の学習における標準的な手法であるニュートン法は, 準同型暗号上で計算的に扱いにくいヘッセ行列の逆行列計算が必要である. [3] で提案された近似手法は, 秘密逆行列計算と秘密比較に依存するために, 本研究の多項式近似に比べ, より計算コストが高いと考えられる. 我々のアプローチと [3] との比較を表 1 に示す. 本研究では, ロジスティック関数の計算を扱うために多項式フィッティングを用い, ニュートン法ではなく一階微分のみを計算する確率的勾配法を用いる. さらに, 新たに提案する近似関数の変換と, 秘密近似除算プロトコルの採用により, オーバーフロー問題の効率的に解決し大規模なデータセットを扱うことを可能にした. そして, すべての処理は準同型暗号上で計算可能である.

表 1: N と d はそれぞれ事例と事例の次元を表す. L は経験近似のサンプル数を表し, P はパーティー数, s は Euler の方法のステップ数, K は多項式の次数を表す.

	Complexity	Primitives
[3] protocol 1	$O(P^2(Nd^2 + d^3 \log d))$ multiplication, $O(NL)$ encryptions	matrix inversion Greater Than
[3] protocol 2	$O(N(s + d) + d^2)$	Euler's method Hessian lower bound
Ours	$O(d + K)$ encryptions	factor removal

2. 準備

2.1 ロジスティック回帰

ロジスティック回帰モデルは, 従属変数 y といくつかの独立変数 $\boldsymbol{x} = (x_1, x_2, \ldots, x_d)$ の関連度合いを説明するために設計されたモデルである. このモデルは, 入力事例のラベルを予測するために多く用いられている. このロジスティック回帰モデ

連絡先: 呉　双, 筑波大学システム情報工学研究科, 〒 305–8572 茨城県つくば市天王台, 029-853-3826, anita_ ws[at] mdl.cs.tsukuba.ac.jp

ルの予測関数は, 以下のように定義される.

$$f(\boldsymbol{x}_i;\boldsymbol{w}) = \sigma(\boldsymbol{w}^T\boldsymbol{x}) = 1/(1+\exp(-\boldsymbol{w}^T\boldsymbol{x})),$$

ここで $\sigma(z) = 1/(1+\exp(-z))$ はロジスティック関数である. ロジスティック回帰モデルは次のように定義される.

$$\Pr(y|\boldsymbol{x},\boldsymbol{w}) = \left[\sigma(\boldsymbol{w}^T\boldsymbol{x})\right]^y \left[1-\sigma(\boldsymbol{w}^T\boldsymbol{x})\right]^{1-y}.$$

データセット $D = \{(\boldsymbol{x}_i, y_i)|\boldsymbol{x}_i \in R^d, y_i \in \{0,1\}\}_{i=1}^N$ が与えられたとき, 重みベクトル $\boldsymbol{w}$ は以下の最適化問題の解として得られる.

$$\arg\min_{\boldsymbol{w}} \lambda R(\boldsymbol{w}) - \sum_{i=1}^{N} \log \Pr(y_i|\boldsymbol{x}_i,\boldsymbol{w}), \tag{1}$$

ここで $-\sum_{i=1}^N \log \Pr(\mathrm{y_i}|\mathrm{x_i},\mathrm{w})$ はロジット損失であり, $R(\boldsymbol{w})$ は正則化項を表す. $\lambda \geqslant 0$ は正則化パラメータと呼ばれ, 与えられたデータへの適合度合いを制御する.

式 (1) の最適化問題を解く標準的な方法は, 二階微分を用いるニュートン・ラフソン法である. この方法は暗号上では計算上に扱いにくい, 逆行列計算を行う必要がある. そのため, 本研究では L2 正則化項 $R(\boldsymbol{w}) = \|\boldsymbol{w}\|_2 = \sum_{j=1}^d w_j^2$ を持つロジスティック回帰モデルを確率的勾配法を用いて解く. 確率的勾配法とは, 各更新毎にランダムに抽出した一つの事例 $(\boldsymbol{x}_t, y_t)$ の勾配を基に重みベクトルを以下のように更新する方法である.

$$\begin{aligned} w_{t+1}^j &\leftarrow (1-\lambda\eta_t)w_t^j + \eta_t \nabla_{w^j} \log \Pr(y_i|\boldsymbol{x}_i,\boldsymbol{w}) \\ &= (1-\lambda\eta_t)w_t^j + \eta_t(y_t - \sigma(\boldsymbol{w}\cdot\boldsymbol{x}))x_t^j. \end{aligned} \tag{2}$$

2.2 加法準同型暗号

提案プロトコルの安全性は, 公開鍵暗号システムにおける加法準同型暗号に基づく. 与えられた公開鍵と秘密鍵のペアを (pk, sk), $r \in \mathbb{Z}_n$ を乱数としたとき, メッセージ $m \in \mathbb{Z}_n$ を公開鍵 pk で暗号化したものを $c = E_{pk}(m;r)$ と表す. 一度暗号化した平文 m は, 暗号文 c の公開鍵 pk に対応する秘密鍵 sk を用いて $m = D_{sk}(c)$ による復号で得ることができる. 加法準同型暗号の重要な特徴はその加法的準同型性であり, この性質のために復号を行うことなく暗号文同士の加算が可能である. 二つの平文 m_1, $m_2 \in \mathbb{Z}_n$ が与えられたとき, 暗号化された値同士の安全な加算は以下で与えられる.

$$E_{pk}(m_1;r_1)\cdot E_{pk}(m_2;r_2) = E_{pk}(m_1+m_2;r_1+r_2). \tag{3}$$

$m_2 = 0$ とすれば, この性質はメッセージ m_1 の再ランダム化を可能にする. さらに, $v \in \mathbb{Z}_n$ を定数としたとき, 乗算 vm の計算は式 (3) により達成される.

$$E_{pk}(m;r)^v = E_{pk}(vm;rv). \tag{4}$$

本研究では, この加法準同型性をもつ暗号システムである Paillier 暗号 [4] を用いる. 簡潔さのために以降では乱数を省略し, メッセージ m の暗号文を $E_{pk}(m)$ と表記する.

3. プライバシ保護ロジスティック回帰

3.1 問題定義

異なるパーティーがそれぞれ保持しているデータから構築した共用データベースがあり, 個人, 企業, 研究機関などのパーティーがこのデータベース上でロジスティック回帰による解析を行いたいとする. しかし 各パーティーは. 所有権や守秘義務などにより, 保有している秘密データを共有もしくは共用データベース上のデータと結合したくない. 本研究では, 実際に各パーティーが持つ秘密データを結合することなく, 共用データベース上でロジスティック回帰解析を行うことを目的とする.

以下のような共用データベース $\mathbf{X} = \{\boldsymbol{x}_i|\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^N$, $\boldsymbol{y} = \{y_i|y_i \in \{0,1\}\}_{i=1}^N$ を考える.

$$X: \begin{pmatrix} \overbrace{\begin{matrix} x_{1,1}^A & \cdots & x_{1,d_1}^A \\ x_{2,1}^A & \cdots & x_{2,d_1}^A \\ \vdots & & \vdots \\ x_{i,1}^A & \cdots & x_{i,d_1}^A \\ \vdots & & \vdots \\ x_{N,1}^A & \cdots & x_{N,d_1}^A \end{matrix}}^{X_A} & \overbrace{\begin{matrix} x_{1,d_1+1}^B & \cdots & x_{1,d}^B \\ x_{2,d_1+1}^B & \cdots & x_{2,d}^B \\ \vdots & & \vdots \\ x_{i,d_1+1}^B & \cdots & x_{i,d}^B \\ \vdots & & \vdots \\ x_{N,d_1+1}^B & \cdots & x_{N,d}^B \end{matrix}}^{X_B} \end{pmatrix}$$

$$\mathbf{y}: [y_1, y_2, \cdots, y_N]^T$$

それぞれのベクトル $\boldsymbol{x}_i$ は $\boldsymbol{y}$ の要素 y_i と対応しており, y_i は $\boldsymbol{x}_i$ のクラスラベルを表している.

パーティー A およびパーティー B の 2 つのパーティーについて考える. パーティー A は, $\mathbf{x}_i^A \in \mathbb{R}^{d_1}$ を保持しており, $\boldsymbol{x}_i^A$ をパーティー B から隠したい情報であるとする. 一方, パーティー B は, $\boldsymbol{x}_i^B \in \mathbb{R}^{d_2}$) を保持しているとする. ここで, $d = d_1 + d_2$ である. このとき, パーティー A とパーティー B の訓練データは, $\boldsymbol{x}_i^A$ と $(\boldsymbol{x}_i^B, y_i)$ を結合したものである. 上記のような訓練データの分割は, 同一データを属性によって分離し各パーティーへ分割することによる, データの垂直分離を行っていると考えることができる. 本稿で取り扱う問題は, パーティー A とパーティー B が互いの秘密データを実際に結合せずに, ロジスティックモデル (分類器) を構築することである.

3.2 多項式による近似

SGD における重みベクトル $\boldsymbol{w}$ は式 (2) によって更新されるため, ヘッセ行列の逆行列を計算する必要はないが, 更新過程において 2 つの問題が残っている. 一つは, 式 (2) の更新過程においてロジスティック関数の評価をする必要があるが, 準同型暗号上ではこの計算ができないことである. そこで, ロジスティック関数 $\sigma(z)$ を K 次の多項式で近似し, 多項式を秘密計算によって評価するための秘密多項式計算 (secure polynomial computation: SPC) プロトコル [7] を用いることでこの問題を解決する. 2 つ目は, ロジスティック関数の入力と出力や学習の更新パラメータ (ステップサイズ) は実数 $\mathbb{R}$ の領域をとるが, 準同型暗号上では整数しか扱うことができないことである. 任意の固定精度の実数は整数に変換することができるが, ロジスティック関数の出力を整数にすることができない. この問題を解決するために, ロジスティック関数の多項式近似と相似した整数値を返す関数を用いる. この関数を用いることによって, 更新式におけるすべての値を整数に変換することができる.

$\tilde{\sigma}(z) = \sum_{k=0}^K \alpha_k z^k$ を, ロジスティック関数 $\sigma(z)$ の多項式近似であるとする. α_k は実数であるため, この近似多項式は直接準同型暗号上で扱うことができない. その代わりに, 下記の式で定義するロジスティック関数 $\sigma(z)$ を多項式近似し, 定数 M により拡大した関数 $\tilde{\sigma}_M(z)$ を用いる.

$$\tilde{\sigma}_M(z) = M\sum_{k=0}^{K} \alpha_k (\frac{1}{M}z)^k = M\sum_{k=0}^{K} \beta_k z^k,$$

ここで M は, $M^{K+1}\beta_i = M^{-k+K+1}\alpha_k$ $(k = 0, 1, \ldots, K)$ が整数になるようにするための拡大定数である. この多項式近似では, 入力と出力の空間を M 倍に拡大することによって, すべての多項式の係数が整数になるようにしている.

3.3 秘密更新

正則化パラメータ λ とステップサイズパラメータ η_t に用いる, 別の拡大定数 M' を導入する. 定数 M' は, $M'\eta_t$ や $M'(1-\lambda\eta_t)$ がどちらも整数になるようにするための拡大定数である. 式 (2) において $\tilde{\sigma}_M(z)$ を $\sigma(z)$, $M'\eta_t$ を η_t, $M'(1-\lambda\eta_t)$ を $(1-\eta_t\lambda)$ に置き換え, 式 (2) の左辺と右辺をそれぞれ M^{K+1} 倍することによって以下の式が得られる.

$$\begin{aligned} w_{t+1}^{j'} &= M'M^{K+1}(1-\lambda\eta_t)w_t^j \\ &\quad + M'M^K\eta_t(My_t - \tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t))x_t^j \\ &= M'M^{K+1}w_{t+1}^j \end{aligned} \tag{5}$$

このように M と M' を適切に設計することで, 式 (5) におけるすべての項は整数となり, 準同型暗号上で扱うことができる. 表記を簡潔にするためにベクトル $\boldsymbol{w}$ のすべての要素をそれぞれ暗号化したベクトルを $E_{pk}(\boldsymbol{w})$ で表したとき, 式 (5) の両辺を暗号化することによって, 準同型暗号上における更新式は以下の式で与えられる.

$$\begin{aligned} E_{pk}(\boldsymbol{w}'_{t+1}) &= [E_{pk}(M\boldsymbol{w}_t)]^{M'M^K(1-\lambda\eta_t)} \cdot [E_{pk}(y_t)]^{M'M^{K+1}\eta_t\boldsymbol{x}_t} \\ &\quad \cdot [E_{pk}(M^K\tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t))]^{-M'\eta_t\boldsymbol{x}_t}, \end{aligned} \tag{6}$$

$M^K\tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ は本稿のプライバシーモデルにおいて SPC プロトコルを用いて評価できるため, 式 (6) による SGD の更新式は安全に評価することができる.

3.4 プライバシ保護ロジスティック回帰プロトコル

Protocol1 に, プライバシ保護ロジスティック回帰 (privacy-preserving logistic regression : PPLR) プロトコルを示す. 式 (2) において j 番目のパラメータ w_t^j の更新における秘密情報は, w_t^j, $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$, y_t, x_t^j である. 本プロトコルでは, パーティー A の秘密データ $\boldsymbol{x}^A$ に関するモデルパラメータ $\boldsymbol{w}^A$ と, パーティー B の秘密データ $\boldsymbol{x}^B$ に関するモデルパラメータ $\boldsymbol{w}^B$ は, それぞれのパーティーが保持しており, パラメータの更新は各パーティーが別々に行う. パーティー A の視点から見たとき, 自身の保持している情報は参照できるため, モデルパラメータ $\boldsymbol{w}^A$ を更新するためにはパーティー B が保持している $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$ と y_t が必要である. 従って, $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$ と y_t を得ることができれば, パーティー A はパーティー B から独立に $\boldsymbol{w}^A$ の更新を行うことができる. これは, パーティー B についても同じことがいえる[*1]. このようにパーティー B とパーティー A の更新プロトコルは (y_t の扱いを除いて) 対称であるため, 以下ではパーティー A のプロトコルについて議論する.

近似関数の計算を行うために SPC プロトコルを, プライバシを保護した上で除算をおこなうために秘密因数除去 (secure factor removal : SFR) プロトコル [2] を適用する. SPC プロトコルと SFR をプロトコルを, 以下のように表記する.

$$\begin{aligned} \mathrm{SPC}(\emptyset, E_{pk}(z)) &\longrightarrow (\emptyset, , E_{pk}(\tilde{f}(z))) \\ \mathrm{SFR}(E_{pk}(x), \emptyset) &\longrightarrow (s, \emptyset) \end{aligned}$$

ここで, $s \approx E_{pk}(\lfloor\frac{x}{y}\rfloor)$ である.

Protocol 1 プライバシ保護ロジスティック回帰

- **Public input:** Coefficient $\beta_k (1 \leqslant k \leqslant K)$, M, $M' \in \mathbb{Z}_n$
- **Input of A:** $\boldsymbol{x}_i^A \in \mathbb{Z}^{d_1}$
- **Input of B:** $\boldsymbol{x}_B = \{(\boldsymbol{x}_i^B, y_i) | \boldsymbol{x}_i^B \in \mathbb{Z}^{d_2}, y_i \in \{0,1\}\}_{i=1}^N$
- **Output of A:** $E_{pk_B}(M\boldsymbol{w}^A)$
- **Output of B:** $E_{pk_A}(M\boldsymbol{w}^B)$

1: **setup phase**. B generates cryptographic key pair (pk_B, sk_B), share the public key with A. A initializes his partial parameter $\boldsymbol{w}_t^A$.
2: **phase 1**. scalar-product $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$ computation:
(a) B computes $E_{pk_B}(M\boldsymbol{w}_t^B\cdot\boldsymbol{x}_t^B)$ and $E_{pk_B}(y_t)$. Sends them to A.
(b) A computes: $E_{pk_B}(M\boldsymbol{w}\cdot\boldsymbol{x}) = E_{pk_B}(M\boldsymbol{w}_t^B\cdot\boldsymbol{x}_t^B)\cdot E_{pk_B}(M\boldsymbol{w}_t^A\cdot\boldsymbol{x}_t^A)$
3: **phase 2**. update $\boldsymbol{w}$:
(a) A and B evaluates $\tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ by SPC:
$\big(E_{pk_B}(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t), \emptyset\big) \longrightarrow \big(E_{pk_B}(M^K\tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)), \emptyset\big)$
(b) A updates $E_{pk_B}(\boldsymbol{w}_t^A)$ with eq. (6):
obtains $E_{pk_B}(M'M^{K+1}\boldsymbol{w}_{t+1}^A)$
4: **phase 3**. A removes factor $M'M^K$ SFR:
$\big(E_{pk_B}(M'M^{K+1}\boldsymbol{w}_{t+1}^A, \emptyset)\big) \longrightarrow \big(E_{pk_B}(M\boldsymbol{w}_{t+1}^A), \emptyset\big)$
5: If convergence conditions are satisfied, terminate the protocol. Otherwise, $t = t+1$ and jump to step 2.

phase 1(a) ではパーティー B がパーティー A に $E_{pk_B}(M\boldsymbol{w}_t^B\cdot\boldsymbol{x}_t^B)$ と $E_{pk_B}(y_t)$ を送信し, phase 1(b) でパーティー A は受け取ったデータと自身の保持しているデータを組み合わせることでパラメータとデータの内積 $E_{pk_B}(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ を得る. phase 2(a) では SPC プロトコルを用いることによって $E_{pk_B}(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ から $\tilde{\sigma}_M(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ を評価し, この値を用いることで式 (6) による更新をパーティー A 単独で実行することが可能となっている. この更新処理ではパラメータを $M'M^K$ 倍にするため, phase 3 で SFR プロトコルを用いることによってこれを取り除いている. ステップ 5 では収束判定を行っているが, SGD では N 回の更新で終了させることがよく行われる. この収束判定には, 秘密収束判定プロトコル [5] を用いることもできる.

3.5 計算量解析

プロトコルの多くの計算時間を占める暗号化処理の回数に対する, PPLR 全体の時間計算量と通信計算量を導出する. phase 1 は 2 つのが共同して内積 $E_{pk_B}(M\boldsymbol{w}_t\cdot\boldsymbol{x}_t)$ を計算しており, $(d+1)$ 回の暗号化処理が必要である. phase 2 では SPC プロトコルによる近似関数の評価を行っており, それに $(K+1)$ 回の暗号化処理を行っている. phase 3 では 2 つのパーティーがそれぞれ SFR プロトコルによって定数倍 $M'M^K$ を除去しているため, $2d$ 回の暗号化処理を行っている. 従ってこれらをまとめると, T をアルゴリズムが収束するまでに必要な更新の回数としたとき, PPLR の時間計算量は $O(T(d+K))$ となる. SPC と SFR で行われる通信の計算量はそれぞれ $K+2$, $3d$ であるため, PPLR の通信計算量は $O(d+K)$ となる.

4. 実験

SPECT (SPT), SPECTF (SPTF), Haberman (HM), breast-cancer-wisconsin (BCW) [6], Mammographic_mass

*1 正確には, プライバシモデルにおいてパーティー B が y_t を保持していることを仮定しているため, パーティー B が更新を行う際に は $\boldsymbol{w}_t\cdot\boldsymbol{x}_t$ のみしか必要がない.

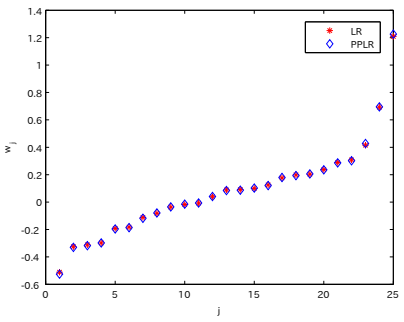

図 1: 赤の星印は通常のロジスティック回帰法 LR, 青の菱形は PPLR によって得られるパラメータ w_j を表している.

表 2: LR と PPLR の予測精度

	LR		PPLR		
accuracy (%)	SGD	BGD	9th	7th	5th
SPT	83.58	84.29	82.83	82.08	77.61
SPTF	76.12	79.44	76.12	76.12	76.12
HM	74.07	76.01	74.07	74.07	74.07
BCW	92.86	96.29	90.00	85.71	68.57
Mammo	70.47	69.73	70.98	71.50	72.02
German	71.00	71.80	70.00	70.00	70.00

表 3: PPLR の各段階で一つの事例を処理するための時間 (秒).

phase 1	SPT	SPTF	HM	BCW	Mammo	German
9th	0.05	0.05	0.05	0.04	0.05	0.04
7th	0.05	0.06	0.05	0.04	0.05	0.05
5th	0.05	0.05	0.05	0.05	0.05	0.05
phase 2						
9th	3.34	3.44	3.20	3.22	3.30	3.37
7th	1.96	2.05	1.95	1.85	1.93	1.92
5th	1.01	1.09	1.07	1.01	1.01	1.05
phase 3						
9th	4.32	8.60	0.64	1.84	0.85	4.81
7th	4.29	8.01	0.66	1.82	0.86	4.71
5th	4.10	7.99	0.66	1.86	0.83	4.69

(Mammo), German [8] における PPLR の予測精度と実行時間を示すことで, PPLR の実現可能性を示す.

表 2 ではプライバシ保護を行わない通常のロジスティック回帰と PPLR の予測精度の比較を行っており, PPLR が正当に予測できることを示している. 一般に, 近似多項式の次元を大きくするほど予測精度が良い. 表 3 に, プロトコルの効率性を示すために, プロトコルの各段階において一つの事例を処理する際の実際の計算時間と通信時間を示す. phase 1 における内積の計算は定数時間であり二回の暗号化しか必要ないため, すべてのデータセットで例外無く 0.05 秒付近の計算時間となっている. phase 2 の SPC を含めた秘密更新の計算時間は多項式の次元に関係しているため, 明らかに 9 次元の多項式の計算時間が一番長く, 5 次元のときの約 3 倍となっている. phase 3 の因数除去の計算時間は属性数に依存するため, 例えば, 23 属性の SPECT のデータが 4.25 秒の時間がかかっているのに対し, 45 属性の SPECTF では 8.20 秒の時間がかかっている.

図 4. に, German データにおいてプライバシ保護を行わない通常のロジスティック回帰による学習で得られたパラメータ $\boldsymbol{w}$ と, PPLR プロトコルで得られたパラメータを図示する. 図により, PPLR プロトコルによって得られるパラメータが通常のロジスティック回帰の解と同等の解が得られていることを示している.

5. 終わりに

本稿では, 複数のパーティーがそれぞれ異なる属性値を保持する垂直分割モデルにおいて, 互いに自身が持つ情報を共有せずに確率的勾配法に基づきロジスティックス回帰を学習する暗号理論的に安全なロジスティックス回帰分析プロトコルを提案した. ロジスティックス回帰は非線形なシグモイド関数の評価や逆行列演算を含むため, 暗号プロトコルとしての実現が困難である. 提案法では, (1) オンラインロジスティックス回帰による逐次更新, (2) シグモイド関数の多項式近似, の二つの工夫を導入し, ロジスティックス回帰分析が暗号理論的に安全に実現可能であることを示した. また実験により, 提案法で学習した解は, 多項式の次数が十分に高い場合, 近似を導入しない場合とほぼ同等であることを示した.

謝辞

本研究は, JST CREST「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」領域におけるプロジェクト「自己情報コントロール機構を持つプライバシ保護データ収集・解析基盤の構築と個別化医療・ゲノム疫学への展開」の助成を受けた.

参考文献

[1] S. E. Fienberg, W. J. Fulp, A. B. Slavkovic, and T. A. Wrobel. “ secure ” log-linear and logistic regression analysis of distributed databases. In *Privacy in Statistical Databases*, pages 277–290. Springer, 2006.

[2] R. Hall, S. E. Fienberg, and Y. Nardi. Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, 27(4):669, 2011.

[3] R. Hall, Y. Nardi, and S. Fienberg. Achieving both valid and secure logistic regression analysis on aggregated data from different private sources. *arXiv preprint arXiv:1111.7277*, 2011.

[4] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology-EUROCRYPT'99*, pages 223–238. Springer, 1999.

[5] J. Sakuma, S. Kobayashi, and R. N. Wright. Privacy-preserving reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 864–871. ACM, 2008.

[6] W. N. Street, O. L. Mangasarian, and W. H. Wolberg. An inductive learning approach to prognostic prediction. In *ICML*, pages 522–530. Citeseer, 1995.

[7] S. Wu, J. Kawamoto, H. Kikuchi, and J. Sakuma. Privacy-preserving online logistic regression based on homomorphic encryption. In *IEICE Tech. Rep*, volume 113, pages 67–74. IBISML, 10 2013.

[8] I. Yeh, K.-J. Yang, T.-M. Ting, et al. Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with Applications*, 36(3):5866–5871, 2009.