Ring-LWEベース準同型暗号を用いたプライバシー保護決定木分類

福井 智[1]　王 立華[2]　林 卓也[2]　小澤 誠一[1,a)]

概要：近年，多くの組織や個人が外部のサーバに計算や保管を委託するクラウドコンピューティングを利用しており，機械学習サービスのクラウド上での運用が進んでいる．本研究では，事前に訓練済みの機械学習モデルを保持するモデル所持者と入力データを持つ依頼人，計算資源を提供する外部サーバの 3 者が参加する計算モデルを想定する．我々は Ring-LWE ベース準同型暗号を用いたプライバシー保護決定木分類を提案する．Ring-LWE ベース準同型暗号を用いた効率的なセキュア大小比較と，準同型内積演算を使用することで，依頼人のデータとモデル提供者の決定木モデルのデータの 2 入力を双方暗号化した状態で安全に計算を外部委託可能なプロトコルを構築した．提案プロトコルがオンラインで公開されているデータに対して実時間で動作可能であることを示した．

キーワード：プライバシー保護データマイニング，機械学習，準同型暗号，決定木

Privacy-Preserving Decision Tree Classification Using Ring-LWE-Based Homomorphic Encryption

Satoshi Fukui[1]　Lihua Wang[2]　Takuya Hayashi[2]　Seiichi Ozawa[1,a)]

Abstract: As the number of cloud computing users has been soaring, it is solicited to establish a secure computing platform where people can employ machine learning algorithms while preserving privacy of data. In this paper, we propose a privacy-preserving decision tree classification protocol using ring-LWE-based homomorphic encryption. It applies to cloud computing system with three-party: a client who has sensitive data, a model holder who has a pre-trained decision tree, and an outsourced server that supplies computing resource. To protect data privacy, input data and tree construction are encrypted by a client and a model holder, respectively, before being sent to an outsourced server. We demonstrate that the proposed privacy-preserving decision tree classification protocol works within a practical time for several public data sets.

Keywords: Privacy-Preserving Data Mining, Machine Learning, Homomorphic Encryption, Decision Tree

1. 序論

IoT の発達によりあらゆるデータを取得できる昨今，収集した膨大なデータの利活用が進んでいる．それに伴い外部のサーバに計算や保管を委託可能なクラウドコンピューティングサービスの需要が高まっている．自身で計算資源を持たずともクラウドサーバを利用することで大規模データの管理や，機械学習を用いたデータ分析に必要な重い計算処理が可能となる．一方で機械学習アルゴリズムを用いてデータ分類を行う際，計算処理が必要なので分析対象のデータを明かさなければならない．例えば，銀行の不正送金検知を想定する．顧客の送金データをクラウドサーバに送信してデータ分析を行うと機密情報の漏洩につながる．

[1] 神戸大学大学院工学研究科電気電子工学専攻
Department of Electrical and Electronic Engineering, Graduate School of Engineering, Kobe University
[2] 国立研究開発法人　情報通信研究機構
National Institute of Information and CommunicationsTechnology
[a)] ozawasei@kobe-u.ac.jp

また，データ分析に用いる機械学習モデルの公開は知的財産の損失になるため，クラウドサーバ管理者や外部者への漏洩を防ぐことが必要である．このように機密情報・個人情報のプライバシー保護，知的財産損失の問題がクラウドサーバ利用の障壁となっている．

本研究では，機械学習アルゴリズムの 1 つである決定木を扱う．決定木は入力データに対してテスト (例えば，気温 $>$ 20 度 かどうか) を行い分類を行う単純なアルゴリズムであるが，他の機械学習アルゴリズムと比べて結果の解釈が容易であるという利点があり，病気の診断や，金融リスク評価などに広く使用されている．

本研究では分析対象となるデータを所持する依頼人，トレーニング済み決定木モデルを持つモデル所持者，クラウドサーバの 3 者が参加する状況を想定する．この状況下で，依頼人のデータとモデル所持者の持つ決定木モデルの情報を明かさずにデータ分類を行うプロトコルを提案する．

関連研究

Wu ら [10] は加法準同型暗号と紛失通信を用いたプライバシー保護決定木プロトコルを提案した．決定木の構造を秘匿するために，彼らは決定木を完全二分木でランダムな木に変形している．そのためサーバの計算量と通信量は木の深さに対して指数的に増加する．この方法ではスパースな決定木において計算量，通信量が増加する可能性がある．Tai ら [8] はこれに対し，線形関数で決定木分類計算を行う手法を用いて，指数的に計算量と通信量が増加するのを避けるプロトコルを提案した．これにより決定木がスパースなまま評価でき，計算量と通信量は決定ノード数に依存する．

[8] は整数比較に加法準同型暗号を使用した DGK プロトコル [2] を用いている．整数比較プロトコルでの計算には乗法が含まれるため，DGK プロトコルでは一方を平文，一方を暗号文として処理を行なっており，ビット毎に暗号化する必要がある．これに対して，Saha ら [7] は ring-LWE ベース Somewhat 準同型暗号を用いた整数比較プロトコルを提案しており，16bit 及び 32bit の整数比較において DGK プロトコルの約 146 倍の速度で比較を行うことを示している．また，Wang ら [9] は Saha らのプロトコルを改善しており，Saha らのプロトコルより約 2 倍の速さで比較を行うことができる．

本研究の貢献

本研究は，ring-LWE ベース準同型暗号を用いて，3 者間で決定木分類を行うプロトコルを構築した．本研究の貢献は以下の通りである．

- 既存の決定木分類プロトコルに採用されている比較プロトコル [2] より効率の良い比較プロトコル [9] を用いる．
- 入力データと決定木モデルの情報，双方を暗号化状態で決定木分類を可能にし，実用性を高める．
- 準同型内積演算により，決定木モデルの情報を秘匿して，決定木分類の計算を代理計算サーバに委託できる．
- UCI Machine Learning Repository[4] にて公開されている実データセットに対して実用的な時間でプロトコルが動作可能であることを示した．

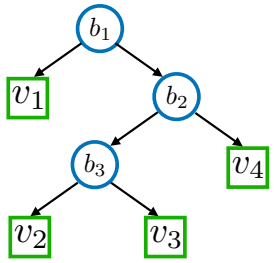

図 1　決定木の例
Fig. 1　Decision tree example

2. 準備

2.1 決定木

決定木は効率性と単純さにより広く使用される機械学習アルゴリズムである．**図 1** はその例である．決定木 $\mathcal{T} : \mathbb{Z}^N \to \mathbb{Z}$ は，特徴ベクトル $\boldsymbol{x} = (x_1, \ldots, x_N)$ と呼ばれる属性データを入力とし，出力 $\mathcal{T}(\boldsymbol{x})$ は $\boldsymbol{x}$ が属するクラスである．通常，特徴ベクトル空間は $\mathbb{R}^N$ であるが，本研究では暗号化された属性データが入力であるため $\mathbb{Z}^N$ に変更している．決定木は決定ノードと葉ノードの 2 種類のノードで構成される．決定ノードはブール値 $b_i = \mathbb{1}\{x_{\lambda_i} > t_i\}$ を出力する．ただし，$\lambda_i \in [N]$ は特徴ベクトルのインデックス，t_i は閾値とする．葉ノードは出力値 v_k を保持する．本研究で用いる決定木は二分木を想定する．すなわち，各ノードは 0 ～ 2 個の子を持つ．二分木において，決定ノードを m 個とすると，葉ノードは $m+1$ 個である．決定木による分類は木の根ノードから開始し，各決定ノードで特徴ベクトルのある属性値と閾値との大小比較を行う．比較結果に基づいて，左または右の子ノードに進む．この処理を葉ノードに到達するまで繰り返す．出力値 $\mathcal{T}(\boldsymbol{x})$ は葉ノードが持つ v_k となる．

2.2 線形関数による決定木分類 [8]

決定ノード D_i が出力するブール値 $b_i = 1(0)$ は次に進むノードが左 (右) の子に進むことを表す．決定ノード v_i とその左・右の子を繋ぐそれぞれの辺 $\mathsf{e}_{i,0}, \mathsf{e}_{i,1}$ に対して辺コスト $\mathsf{ec}_{i,0} := 1 - b_i, \mathsf{ec}_{i,1} := b_i$ を定義する．決定木の根ノードから各葉ノードには唯一のパスがあり，あるパス P_k に含まれる辺コストの和をパスコスト pc_k と定義する．辺コストは決定ノードの比較結果 b_i によって決まり，比較結

果に従って次に進むノードへの辺コストが0，もう一方のノードへの辺コストは1となる．したがって，入力に対してパスコスト $\mathsf{pc}_k = 0$ である葉ノードが唯一決まり，その葉ノードに到達する．

図1の各葉ノードに対するパスコストは，

$$\begin{aligned}\mathsf{pc}_1 &= 1 - b_1,\\ \mathsf{pc}_2 &= b_1 + (1 - b_2) + (1 - b_3),\\ \mathsf{pc}_3 &= b_1 + (1 - b_2) + b_3,\\ \mathsf{pc}_4 &= b_1 + b_2\end{aligned} \tag{1}$$

となる．

2.3 Ring-LWE ベース準同型暗号

本研究では，準同型暗号として，Ring-LWE ベースの公開鍵準同型暗号ライブラリ SEAL v3.3 [6] を用いる．SEAL は Fan ら [3] で提案した Somewhat 準同型暗号方式を実装しており，加法準同型性と乗法準同型性を持つため，平文を Packing することによって，効率的な内積準同型計算ができる．本節では，概念的に加法と乗法準同型計算と内積計算実現するための Packing 方法を紹介する．暗号化方式の詳細は [6] と [3] にご参照になり，本論文では省略．

表記法

暗号方式が基づく特別な多項式環 $\mathcal{R}$ に関するパラメータは下記の通り表記する．

- n：多項式 $x^n + 1$ を定める 2 の冪乗の整数. 多項式環 $\mathcal{R} := \mathbb{Z}[x]/(x^n + 1)$ を定義
- $q : q = q_1 \times \cdots \times q_k (q_i$ は素数) で構成される整数． 暗号文空間を表す多項式環 $\mathcal{R}_q := \mathbb{Z}_q[x]/(x^n + 1)$ を定義
- $p : p < q$ を満たす整数で，暗号方式の平文空間 $\mathcal{R}_p := \mathbb{Z}_p[x]/(x^n + 1)$ を定義する．
- σ：秘密鍵空間 $\mathcal{R}_{(0,\sigma^2)}$ を定める離散ガウス分布の標準偏差．$\mathcal{R}_{(0,\sigma^2)}$ の要素は環 $\mathcal{R}$ 上の多項式であり，各係数は独立に分散 σ^2 の離散ガウス分布からサンプリングされる．

Somewhat 準同型暗号方式では

ParamGen：システムパラメータ (n, q, p, σ) を出力，
鍵生成 KeyGen：公開鍵 pk と秘密鍵 sk を出力，
暗号化 $\mathsf{Enc}(pk, \cdot)$：平文 m を入力，暗号文 c を出力，
復号 $\mathsf{Dec}(sk, \cdot)$：暗号文 c を入力，平文 m を出力

と 4 つの基本アルゴリズムで構成する．また，加法準同型計算と乗法準同型計算アルゴリズムは Add と Mul で定義し，それぞれの復号アルゴリズムは DecA と DecM で表す．つまり，二つの平文 m と m' に対し，暗号文はそれぞれ $c = \mathsf{Enc}(sk, m)$, $c' = \mathsf{Enc}(sk, m')$ とすると，m と m' の和と積はそれぞれ下記の様に暗号化したまま計算できる．Add，DecA，Mul，DecM の 4 つで準同型演算および演算後の暗号文の復号を行う．

$$\begin{aligned}&\mathsf{Add}(c, c') = c_{add} \in \mathcal{R}_q,\\ &\mathsf{DecA}(sk, c_{add}) = m + m' \in \mathcal{R}_p;\end{aligned} \tag{2}$$

$$\begin{aligned}&\mathsf{Mul}(c, c') = c_{mul} \in \mathcal{R}_q,\\ &\mathsf{DecM}(sk, c_{mul}) = mm' \in \mathcal{R}_p.\end{aligned} \tag{3}$$

以降, $\mathsf{Enc}(pk, \cdot) := [\![\cdot]\!], \mathsf{Add}(c, c') := c \oplus c', \mathsf{Mul}(c, c') = c \otimes c'$ と表記する．また，c と c' の差は式 (2) を用いて実現可能であり，$\mathsf{Sub}(\mathsf{c}, \mathsf{c}') = \mathsf{c} \ominus \mathsf{c}'$ と表記する．

Ring-LWE ベース内積準同型演算 [11]

Yasuda ら [11] は以下の式 (4) と (5) の 2 つの Packing 法を用いることで効率的な準同型内積演算を提案した．

長さ ℓ の整数ベクトル $U = [u_0, u_1, \ldots, u_{\ell-1}]$ に対して，以下の 2 つの多項式を定める．

$$\mathsf{poly}_1(U) = \sum_{i=0}^{\ell-1} u_i x^i, \tag{4}$$

$$\mathsf{poly}_t(U) = -\sum_{i=0}^{\ell-1} u_i x^{n-i}. \tag{5}$$

任意の長さ l の整数ベクトル $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \ldots, \alpha_{\ell-1}], \boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_{\ell-1}]$ をそれぞれ式 (4), (5) で Packing し乗算を行うと，

$$\mathsf{poly}_1(\boldsymbol{\alpha})\mathsf{poly}_t(\boldsymbol{\beta}) = \sum_{i=0}^{\ell-1} \alpha_i \beta_i + \text{other items}$$

となり，定数項が内積 $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle$ の計算結果となる．したがって，2 つの暗号文 $[\![\mathsf{poly}_1(\boldsymbol{\alpha})]\!], [\![\mathsf{poly}_t(\boldsymbol{\beta})]\!]$ に対して式 (3) を用いて準同型乗算を行うことで，暗号化したまま内積計算が可能．

2.4 Wang らのセキュア整数比較 [9]

ℓ-bit 整数比較 [2]

ℓ-bit の整数 $u = \sum_{i=0}^{\ell-1} u_i 2^{\ell-(i+1)}$ の 2 進ベクトルを $U = [u_0, u_1, \ldots, u_{\ell-1}]$ で与え，$1 \le d \le \ell$ で U の d-bit 部分ベクトルを $U_d = [u_0, \ldots, u_{d-1}, 0, \ldots, 0]$ とする．ここで，$u^{\ell-1}$ は整数 u の最下位ビット (LSB) である．

Alice と Bob が a, b の 2 つの ℓ-bit 整数を持つ状況を想定する．両整数の 2 進ベクトルを $A = [a_0, a_1, \ldots, a_{\ell-1}]$, $B = [b_0, b_1, \ldots, b_{\ell-1}]$, d-bit 部分ベクトルを $A_d = [a_0, \ldots, a_{d-1}, 0, \ldots, 0]$, $B_d = [b_0, \ldots, b_{d-1}, 0, \ldots, 0]$ と表す．$0 \le i \le \ell - 1$ と $1 \le j \le \ell - 1$ において，以下の式によって比較計算が実現可能である．

$$\begin{aligned}w_j &= \sum_{k=0}^{j-1} |a_k - b_k| = \sum_{k=0}^{j-1} (a_k - b_k)^2\\ &= \langle A_j - B_j, A_j - B_j \rangle,\\ v_i &= a_i - b_i + 1,\\ c_i &= v_i + w_i.\end{aligned} \tag{6}$$

もし，式 (6) 中の c_i のいずれかの i について $c_i = 0$ であるならば比較結果として $a < b$, そうでなければ $a \geq b$ を得る．

Packing 法

ℓ-bit の整数 u の 2 進ベクトル $U = (u_0, u_1, \ldots, u_{\ell-1})$ に対して，式 (4) と下記の式 (7) の多項式を定める．

$$\mathsf{poly}_2(U) = \sum_{d=1}^{\ell-1} \sum_{j=0}^{d-1} u_i x^{\ell d - j}. \tag{7}$$

任意の 2 つの ℓ-bit の整数 a, b の 2 進ベクトル $A = (a_0, a_1, \ldots, a_{\ell-1})$, $B = (b_0, b_1, \ldots, b_{\ell-1})$ に対して上記の Packing 法を用いることで式 (6) の c_i $(i = 1, \ldots, \ell)$ は，

$$\begin{aligned}\mathsf{poly}(c) = (\mathsf{poly}_1(A) - \mathsf{poly}_1(B))(\mathsf{poly}_2(A) \\ -\mathsf{poly}_2(B) + \mathsf{poly}(I)) + \widetilde{\mathsf{poly}}(\mathbf{1})\end{aligned} \tag{8}$$

で計算可能．c_i は $\mathsf{poly}(c)$ の $x^{i\ell}$ $(i = 1, \ldots, \ell)$ の係数に対応する．ただし， $\mathsf{poly}(I) = \sum_{i=1}^{\ell} x^{(\ell-1)(i-1)}$, $\widetilde{\mathsf{poly}}(\mathbf{1}) = \mathsf{poly}_1(\mathbf{1})\mathsf{poly}(I)$ とし，これらは事前計算可能である．

Wang らのセキュア整数比較プロトコル

式 (4), (7) で定義した Packing 法を用いた Wang らの整数比較プロトコル [9] を説明する．このプロトコルには 3 名の参加者が存在する．Alice と Bob はそれぞれ ℓ-bit の整数 a, b を所持しており，サーバを通してデータを明かすことなく a と b の大小比較を行う．以下にその手順を示す．

1. Alice は秘密鍵と公開鍵を生成し，公開鍵を Bob, サーバに送信．
2. Alice, Bob は，

$$[\![a]\!]_i := \mathsf{Enc}(pk, \mathsf{poly}_i(A))$$

と

$$[\![b]\!]_i := \mathsf{Enc}(pk, \mathsf{poly}_i(B))$$

を $i = 1, 2$ についてそれぞれ計算し，サーバに送信．
3. サーバは，

$$\begin{aligned}[\![c]\!] := ([\![a]\!]_1 \ominus [\![b]\!]_1) \otimes ([\![a]\!]_2 \ominus [\![b]\!]_2 \oplus \mathsf{poly}(I)) \\ \oplus \widetilde{\mathsf{poly}}(\mathbf{1})\end{aligned} \tag{9}$$

を計算．
4. サーバは $\mathcal{R}_p$ から一様ランダムにサンプリングした多項式 $r \leftarrow \mathcal{R}_p$ で $[\![c]\!]$ をマスクする．

$$[\![c']\!] := [\![c]\!] \oplus r \tag{10}$$

$[\![c']\!]$ を Alice に送信．
5. Alice は $[\![c']\!]$ を復号した

$$c' = \mathsf{Dec}(sk, [\![c']\!])$$

をサーバに送信．
6. サーバは以下の計算によりマスクを取り除く．

$$c = c' - r = \sum_{i=0}^{\ell-1} c_i x^{i\ell} + \text{other items}, \tag{11}$$

$i = 0, \ldots, \ell - 1$ について c_i の $i\ell$ 次の係数のいずれかが 0 であれば $a < b$, そうでなければ $a \geq b$ を比較結果として得る．

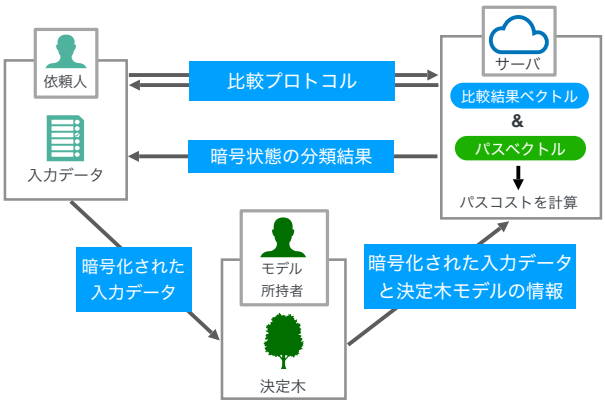

図 2　代理計算モデル
Fig. 2　Our computation model

3. プライバシー保護決定木分類

我々は [9] の整数比較プロトコルを使用して，[8] のプライバシー保護決定木プロトコルをベースに，3 者でプライバシー保護決定木分類を行うプロトコルを提案する．

3.1 代理計算モデル

本研究の代理計算モデルを図 2 に示す．具体的には，決定木分類モデルを持つある組織や企業がクラウドサーバにその情報を暗号化して送信し，計算を外部委託する状況を想定している. 分析対象となる特徴ベクトルを持つ依頼人，訓練済みの決定木モデルを所有しているモデル所持者，モデル所持者の代わりに計算処理を行うサーバの 3 者が参加する．依頼人は特徴ベクトルを暗号化してモデル所持者に送信する．モデル所持者は決定木の閾値，パスコストの計算に必要な情報を暗号化する．モデル所持者の保持する決定木モデルの各決定ノードの閾値が依頼人の持つ特徴ベクトルのどの次元と比較するかという情報 (λ_i) を秘匿するために依頼人の暗号化された特徴ベクトルはモデル所持者を中継してサーバに送信される．サーバは送信された情報をもとにモデル所持者に代わり決定木分類に必要な計算処理を行い暗号状態の分類結果を依頼人に送信する. 依頼人は復号して分類結果を得る．これにより企業や組織が自身でサーバを保持していない場合でも，プライバシーを保護した状態で決定木分類が可能となる．

3.2 内積演算によるパスコスト計算

本研究では，2.2 節で説明したパスコストを暗号化してサーバに送信することで決定木モデルの情報を秘匿し，パスコストの計算をサーバに委託可能にした．具体的には，パスコストを以下の 2 つのベクトルに変形する．

$$\mathsf{pc_k} = \langle [1, b_1, \cdots, b_m], [P_0, P_1, \cdots, P_m] \rangle := \langle \boldsymbol{B}, \boldsymbol{P} \rangle$$

$\boldsymbol{B}$ は 1 と比較結果 b_i を並べたベクトル．$\boldsymbol{P}$ はその係数ベクトルにあたる．以降 $\boldsymbol{B}$ を比較結果ベクトル，$\boldsymbol{P}$ をパスベクトルと記述する．

上記の方法を用いて式 (1) のパスコストを以下の通り変形できる．

$$\begin{aligned}
\mathsf{pc}_1 &= \langle \boldsymbol{B}, [1, -1, 0, 0] \rangle, \\
\mathsf{pc}_2 &= \langle \boldsymbol{B}, [2, 1, -1, -1] \rangle, \\
\mathsf{pc}_3 &= \langle \boldsymbol{B}, [1, 1, -1, 1] \rangle, \\
\mathsf{pc}_4 &= \langle \boldsymbol{B}, [0, 1, 1, 0] \rangle
\end{aligned}$$

ただし，$\boldsymbol{B} = [1, b_1, b_2, b_3]$ である．

パスコストの計算を 2 つのベクトルの内積計算に置き換えることが可能である．したがって 2.3 節の準同型内積演算によってパスベクトルを暗号化した状態で安全にパスコストを計算可能である．

3.3 提案プロトコル

1. (依頼人)

公開鍵，秘密鍵を生成し，公開鍵をモデル所持者とサーバに送信する．

2. (依頼人)

特徴ベクトルの各要素を式 (4), (7) を用いて Packing して暗号化した

$$\begin{aligned}
[\![x_i]\!]_1 &= \mathsf{Enc}(pk, \mathsf{poly}_1(x_i)), \\
[\![x_i]\!]_2 &= \mathsf{Enc}(pk, \mathsf{poly}_2(x_i))
\end{aligned}$$

を $i = 1, \ldots, N$ についてモデル所持者に送信する．

3. (モデル所持者)

$j = 1, \ldots, m$ において閾値 t_j を式 (4), (7) を用いて Packing して暗号化した

$$\begin{aligned}
[\![t_j]\!]_1 &= \mathsf{Enc}(pk, \mathsf{poly}_1(t_j)), \\
[\![t_j]\!]_2 &= \mathsf{Enc}(pk, \mathsf{poly}_2(t_j))
\end{aligned}$$

を生成する．また，$k = 1, \ldots, m+1$ において

$$\begin{aligned}
[\![\boldsymbol{P}'_k]\!]_1 &= \mathsf{Enc}(pk, \mathsf{poly}_1(R_k \cdot \boldsymbol{P}_k)), \\
[\![\boldsymbol{V}_k]\!]_1 &= \mathsf{Enc}(pk, \mathsf{poly}_1(\mathbf{V}_k))
\end{aligned}$$

を生成する．ただし，$R_k, R'_k \in \mathcal{R}^*_p$, $\boldsymbol{V}_k := \boldsymbol{P}_k \cdot R'_k + [v_k, 0, \ldots, 0]$ で定義し，$\boldsymbol{V}_k$ を分類結果ベクトルと呼ぶ．

閾値とその比較対象の特徴ベクトルの要素の暗号文のペア

$$([\![x_{\lambda_j}]\!]_1, [\![x_{\lambda_j}]\!]_2), ([\![t_j]\!]_1, [\![t_j]\!]_2)$$

とパスベクトルと分類結果ベクトルの暗号文ペア

$$[\![\boldsymbol{P}'_k]\!]_1, [\![\boldsymbol{V}_k]\!]_1$$

をそれぞれ $j = 1, \ldots, m$, $k = 1, \ldots, m+1$ についてサーバに送信する．

4. (サーバ)

式 (9) において，$a = t_j, b = x_{\lambda_j}$ として $[\![c_j]\!]$ を計算する．

式 (10) により $\mathcal{R}_p$ から一様ランダムにサンプリングした多項式 $r_j \leftarrow \mathcal{R}_p$ で $[\![c_j]\!]$ をマスクした $[\![c'_j]\!]$ を依頼人に送信する．

5. (依頼人)

$[\![c'_j]\!]$ を復号した

$$c'_j = \mathsf{Dec}(sk, [\![c'_i]\!])$$

をサーバに返却する．

6. (サーバ)

式 (11) により c_j を得る．

c_j の $i\ell$ 番目 $(i = 0, \ldots, \ell - 1)$ のいずれかの係数が 0 であれば $b_j = 1$，それ以外なら $b_j = 0$ とする．これにより，比較結果ベクトル $\boldsymbol{B} = [1, b_1, \ldots, b_m]$ を得る．

7. (サーバ)

比較結果ベクトル $\boldsymbol{B}$ を式 (5) を用いて Packing し，$\mathsf{poly}_t(\boldsymbol{B})$ を得る．以下の計算を行う．

$$\begin{aligned}
[\![\tilde{\boldsymbol{P}}_k]\!] &= [\![\boldsymbol{P}'_k]\!]_1 \otimes \mathsf{poly}_t(\boldsymbol{B}) \\
[\![\tilde{\boldsymbol{V}}_k]\!] &= [\![\boldsymbol{V}_k]\!]_1 \otimes \mathsf{poly}_t(\boldsymbol{B})
\end{aligned}$$

ランダム順列 $\mathcal{P}$ を生成し，順番を入れ替えた，$([\![\tilde{\boldsymbol{P}}_{\mathcal{P}(1)}]\!], [\![\tilde{\boldsymbol{V}}_{\mathcal{P}(1)}]\!]), \ldots, ([\![\tilde{\boldsymbol{P}}_{\mathcal{P}(m+1)}]\!], [\![\tilde{\boldsymbol{V}}_{\mathcal{P}(m+1)}]\!])$ を依頼人に送信する．

8. (依頼人)

$k' = 1, \ldots, m+1$ において $[\![\tilde{\boldsymbol{P}}_{k'}]\!]$ を復号し，

$$\mathsf{Dec}(sk, \tilde{\boldsymbol{P}}_{k'}) = 0 + \text{other items}$$

であれば，$[\![\tilde{\boldsymbol{V}}_{k'}]\!]$ を復号し，

$$\mathsf{Dec}(sk, [\![\tilde{\boldsymbol{V}}_{k'}]\!]) = v_{k'} + \text{other items}$$

から，最終的な出力 $v_{k'}$ を得る．

3.4 データの安全性

提案手法では，依頼人とモデル所持者は honest であると想定する．すなわち，依頼人とモデル所持者は悪意がなく，バグのあるデータを送らない．一方でサーバは honest-but-curious で，つまり依頼された計算は正確に行うが，機会があればデータを覗き見すると想定する．また，依頼人，

表 1　実験に用いた決定木のデータ
Table 1　Decision tree used for experiments

	Iris	Breast	Heart	Nursery	Spam
N	4	30	13	8	57
d	3	6	9	10	24
m	4	8	35	49	110

モデル所持者，サーバはどの 2 者も結託していないことを想定する．

提案手法によってサーバに漏れる情報は以下の通りである．

- 特徴ベクトルの次元数 N
- 決定木モデルの深さ d
- 決定木モデルの決定ノード数 m
- 決定木モデルの葉ノード数 $m+1$
- 依頼人の特徴ベクトルと閾値の比較結果 b_i

依頼人とサーバは結託しておらず依頼人がサーバに秘密鍵を渡すことはない．サーバは決定木の深さと決定ノード数からモデル所持者の決定木モデルの構造を再現できない．モデル所持者とサーバは結託していないので，大小比較の結果 b_i と閾値 t_j から依頼人の特徴ベクトルが推測されることはない．したがって，依頼人の特徴ベクトルと，モデル所持者の決定木モデルを再現するだけの情報が漏洩しないので，提案手法の安全性は妥当であると考える．

4. 実験

4.1 実験準備

実験には UCI Machine Learning Repository にて公開されている 5 つのデータセット：Iris : Iris Data Set, Heart : Heart Disease Data Set，Breast : Breast Cancer Wisconsin Data Set，Nursery : Nursery Data Set，Spam : Spambase Data Set を使用した．Python のオープンソース機械学習ライブラリである scikit-learn を用いてそれぞれのデータセットを使用してトレーニングを行い，決定木のパスベクトルと閾値を求めた．トレーニング済みの決定木のデータを**表 1** に示す．N, d, m はそれぞれデータセットの次元数，決定木の深さ，決定木の内部ノード数を表す．本研究は Wang らの整数比較プロトコル [9] を使用した．プロトコルの入力が整数でなければならないため，各データセットの特徴量ごとに適切な倍数を掛け，小数部分は切り捨てを行う前処理をして入力データセットの数値を $[0, 2^{13})$ の範囲の整数にした．

4.2 性能評価

本節では 3.3 節のプロトコルの速度計測と，Tai らのプライバシー保護決定木プロトコル [8] との通信・暗号化回数の比較から性能評価を行う．

実装には暗号ライブラリ SEAL v3.3 [6] を用いた．提案プロトコルが正しく動作し，現在一般的に推奨される安全性レベルの 128 ビットセキュリティを持つように [1] にしたがい，以下のパラメータを使用した．

$$n = 2048,\ \log_2 q = 54,\ p = 40961.$$

これらは以下の不等式を満たす．

$$q > 8p^2\sigma^4 n^2 + 4p\sigma n^{2/3}.$$

ただし，$\sigma = 3.2$ は SEAL のデフォルトパラメータを用いている．また，入力の整数の最大ビット長 ℓ に対して，$\ell^2 < n$ の条件を満たすように n を設定した．計測は Core i7-7700K(4.20 GHz) のシングルスレッドで行った．計測結果を**表 2** に示す．それぞれ 10 回計測を行なった平均値を表示している．また，総時間には，鍵生成の時間が含まれるため，表の数値の総和より大きい値となる．表 2 の結果から，最も木構造が大きい Spam Data Set に対して，数百ミリ秒かかる．これは十分実用的な速度であると考える．

表 3 に提案手法と Tai ら [8] のプロトコルの通信・暗号化回数を示す．ただし，通信回数は暗号文を送信した回数とする．Tai らのプロトコルは 2-party モデルであり，提案手法の計算モデルにおけるサーバとモデル所持者を同一人物とみなしたモデルである．今回のデータセットは $\ell = 13$ であるので依頼人の暗号化回数は提案手法の方が少ない．通信回数の少なさに関しては，ℓ, m, N の大きさによる．

5. 結論

本稿では ring-LWE ベース準同型暗号を用いることで，計算を安全に外部委託できるプライバシー保護決定木分類プロトコルを提案した．Wang らの比較プロトコル [9] を使用することで依頼人の特徴ベクトル，決定木の閾値の双方を暗号化し，Yasuda ら [11] の準同型内積演算によりパスコストの計算を行うことでモデル所持者の決定木の構造を秘匿した．速度評価では，提案プロトコルをデータセットに対して適用した場合，評価を行った中で最も決定木の構造が大きいデータセットに対して数百ミリ秒の計算時間で処理可能であることを示した．

提案手法ではサーバに比較結果が漏れてしまう．今後の研究として，Lu らのプロトコル [5] のようにサーバに比較結果を漏らすことなく計算を行いかつ，実用的な速度を達成するプロトコルの構築が挙げられる．安全性と高速さを兼ね備えた実用にふさわしいプロトコルの構築によりクラウドサーバを利用したデータ利活用の促進につながるであろう．

謝辞

本研究の成果は JST CREST 研究領域「イノベーション創発に資する人工知能基盤技術の創出と統合化」研究課題「プライバシー保護データ解析技術の社会実装」JST

表 2　データセットに対する計算時間（秒）
Table 2　Run time for the data sets (seconds)

	依頼人		モデル所持者	サーバ		
Data Set	Enc	Dec	Enc	比較	パスコスト	総時間
Iris	0.0018	0.0009	0.0018	0.0046	0.0014	0.0123
Breast	0.0155	0.0027	0.0037	0.0109	0.0019	0.0364
Heart	0.0063	0.0012	0.0182	0.0492	0.0082	0.0993
Nursery	0.0036	0.0187	0.0254	0.0695	0.0112	0.1358
Spam	0.0290	0.0496	0.0576	0.1577	0.0246	0.3406

表 3　通信・暗号化回数の比較
Table 3　Comparing the number of communication and encryption

	提案手法	Tai ら [8]
通信回数	$2N + 10m + 4$	$\ell N + 2m + 2$
依頼人暗号化回数	$2N$	ℓN
モデル保持者暗号化回数	$4m + 2$	0

CREST JPMJCR19F6 の助成を受けて得られたものです.

参考文献

[1] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., Vaikuntanathan, V.,"Homomorphic Encryption Security Standard", HomomorphicEncryption.org (2018).

[2] Damgård, I., Geisler, M., Krøigaard, M., "A correction to efficient and secure comparison for on-line auctions", *IJACT*, pp. 323-324, 2009.

[3] Fan, J., Vercauteren, F., "Somewhat practical fully homomorphic encryption", *IACR Cryptology*, ePrint 2012/144, 2014.

[4] Lichman, M., "UCI machine learning repository", http://archive.ics.uci.edu/ml

[5] Lu, W. J., Zhou, J. J., Sakuma, J., "Non-interactive and output expressive private comparison from homomorphic encryption", In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 67-74, ACM, 2018.

[6] Microsoft SEAL 3.3, https://github.com/Microsoft/SEAL

[7] Saha, T. K., Koshiba, T., "An efficient privacy-preserving comparison protocol", In *International Conference on Network-Based Information Systems*, pp. 553-565, Springer, 2017.

[8] Tai, R. K., Ma, J. P., Zhao, Y., Chow, S. S., "Privacy-preserving decision trees evaluation via linear functions", In *European Symposium on Research in Computer Security*, ESORICS2017, PartⅡ, LNCS10493, pp. 494-512, Springer, 2017.

[9] Wang, L., Hayashi, T., Saha, T. K., Aono, Y., Koshiba, T., Moriai, S., "An efficiently secure comparison scheme using homomorphic encryption," In *Symposium on Cryptography and Information Security*, 2019.

[10] Wu, D. J., Feng, T., Naehrig, M., Lauter, K., "Privately evaluating decision trees and random forests", In *Proceedings on Privacy Enhancing Technologies*, pp. 335-355, 2016.

[11] Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., Koshiba, T., "Practical packing method in somewhat homomorphic encryption", In *Data Privacy Management and Autonomous Spontaneous Security* , DPM2013 and SETOP2013, LNCS8247, pp. 34-50, Springer, 2013.