協調学習スキームを導入したプライバシー保護XGBoost

山本 楓己[1,a)]　王 立華[2,b)]　小澤 誠一[1,c)]

概要：複数の医療機関や金融機関などがもつ機微なデータを互いに公開せず, 分類や異常検知などの AI システムを構築できるプライバシー保護機械学習が注目を集めている. 本研究では, 高性能かつ説明性に優れたアンサンブル決定木モデルである XGBoost に協調学習スキームを導入した, プライバシー保護 XGBoost を提案する. Yang らは損失関数の勾配情報を集約することによって, Zhao らは複数のデータ所有者がモデルを順番に更新することによって, 協調学習を XGBoost に適用した. これに対し, 複数組織が中央サーバと共有するデータの安全性と AI モデルの性能を両立する改良手法を提案する. 提案手法が実際に情報を秘匿したままデータを利活用できているかを検証するため, 安全性の分析とモデルの性能評価を行った. その結果から, 提案手法は高い安全性を保ちながら有用なモデルを学習可能であることが示された.

キーワード：機械学習, 協調学習, プライバシー保護, ビッグデータ解析, アンサンブル分類木

Privacy-Preserving XGBoost Introducing Federated Learning Scheme

Fuki Yamamoto[1,a)]　Lihua Wang[2,b)]　Seiichi Ozawa[1,c)]

Abstract: Privacy-preserving machine learning is attracting attention because it can build AI systems for classification and anomaly detection without exposing sensitive data owned by multiple organizations related to fields such as medical and finance. In this study, we propose a privacy-preserving XGBoost that introduces a federated learning into XGBoost, which is an ensemble decision tree model with high performance and interpretability. To achieve this, Yang et al. proposed a method of aggregating the gradient of the loss function, and Zhao et al. proposed a method in which each data owner updates a common model in order. In contrast, we propose improved methods with both the security of data shared by multiple organizations with a central server and the performance of AI models. We evaluate security and model performance to verify that data owned by data owners can be utilized without revealing each other. The result showed high security and model performance.

Keywords: Machine Learning, Federated Learning, Privacy Preserving, Big Data Analysis, Ensemble Tree Classifier.

1. はじめに

近年の情報技術の発展により, 膨大かつ多様なデータを解析することが可能になった. しかし, 一般にこのようなデータは個人の情報や営業上の秘密など機微な情報を含むことが多く, 第三者提供などを通して利活用する際には, 関連法に基づいた適切な匿名加工やデータ収集を行う必要がある. また, 法規制に準拠した場合でも, 提供データから個人の機微な情報が漏洩するリスクは依然として残ることから, 特に複数組織が有するパーソナルデータの利活用は困難な課題として残されてきた. このような情報漏洩のリスクを低減するための手法として, 複数組織間で安全性を保証した状態でのデータ解析を可能とするプライバシー保護データ解析技術 [1] が注目されている. 本研究では, 互いの

[1] 神戸大学
Kobe University
[2] 国立研究開発法人 情報通信研究機構
National Institute of Information and Communications Technology
[a)] tamafuki929@gmail.com
[b)] lh-wang@nict.go.jp
[c)] ozawasei@kobe-u.ac.jp

情報を明かさないまま, 複数組織間で共通の機械学習モデルを学習することを目標とする.

プライバシー保護データ解析技術として, 暗号状態で特定の演算を可能とする準同型暗号 [2] や, 出力にノイズを付加する差分プライバシー [3] などがある. 準同型暗号には計算コストの増大や演算に対する制限, 差分プライバシーには秘匿性と有用性のトレードオフといったデメリットが存在する. 従来手法では, 上記の手法を用いて加工を施したデータを一箇所へ集約することで機械学習モデルのプライバシー保護学習を実現していた. しかし, 複数組織が有するパーソナルデータを利活用したい場合, たとえ暗号化してデータを共有したとしても, 個人情報保護法で言うところの第三者提供に当たるとの解釈がなされており, 依然としてデータ利活用のハードルは高いままである. 協調学習 [4] は従来の手法と異なり, データを一箇所に集めることなく機械学習モデルの学習を可能とする学習スキームである. 具体的には, 各組織が独立に機械学習モデルの学習を行い, モデルの更新結果のみを一箇所に集約する. その後, 各組織は集約された更新結果を用いてそれぞれの機械学習モデルを更新することで共通のモデルを学習する. 協調学習のスキームで集約されるモデルの更新情報は, 一般に統計データとして扱われ, 個人情報保護法でいうところの第三者提供には当たらないとの見解が出されることもある. このようなことから, 複数組織が有する個人情報を漏洩することなく，相互に利活用するスキームとして期待されている.

協調学習スキームの導入が試みられてきた機械学習モデルの一つに勾配ブースティング決定木 (GBDT) [5] が存在する. GBDT は, 複数の決定木を予測値と真の値の誤差に基づき逐次的に学習する機械学習の手法の一つである. その性能の高さから注目を集めている. Chen らは, GBDT に様々な工夫を加えることで, より高速かつスケーラブルな GBDT モデルである XGBoost を提案した [6]. 具体的には, スパースなデータに対するアルゴリズムや, 近似された分割点を用いるアルゴリズムの実装などの工夫がなされている. また, GBDT は複数の決定木によって構成されるため, 予測の過程がブラックボックスとなる深層学習などと比較して出力に対する高い説明性を持つ. Yang ら [7] は勾配情報を中央サーバーに集約することで XGBoost に協調学習を適用した. しかし, この手法は各データ所有者が持つ統計情報を中央サーバーや他のデータ所有者に対して明かしてしまっている. また, Zhao ら [8] はノイズを付加した共通のモデルを複数データ所有者が順番に学習することで, XGBoost に協調学習と ϵ 差分プライバシーを導入した. 差分プライバシーは機械学習モデルに対する弱い秘匿性を保証するが, 一般にそのためのノイズの付加はモデルの性能を犠牲にすることが知られている.

本研究では, 互いの情報を秘匿しつつ, データ所有者間で共通の XGBoost を学習するための二つの手法を提案する.

一つ目の提案では, $Yang$ らの手法 [7] を基に以下の工夫を加えることで, より各データ所有者のプライバシーに配慮した学習スキームへ改善した.

- 中央サーバーに対して情報を秘匿するために準同型暗号を用いる.
- 各データ所有者に対して情報を秘匿するために復元抽出を用いて勾配情報の集約を行う.

二つ目の提案では, Zhao らの手法 [8] を基に以下の工夫を加えることで, よりモデルの性能に重点を置いた学習スキームを開発した.

- 性能を劣化させる可能性がある差分プライバシーを導入せず, 精度に影響を与えない方法で安全性を向上
- 複数のモデルから最適なモデルを選択するアルゴリズムによる予測精度の改善

2. 準備

2.1 準同型暗号

準同型暗号 [9][2] は, 暗号状態で特定の演算を可能とする暗号方式である. ここで, 平文空間 M における加法を+, 暗号空間 C における加法を $\oplus$ とする. 本稿では加法準同型計算のみを用いるため $\forall m_1,\ m_2 \in M$ に関して, 以下の式 (1) を満たす. pk と sk はそれぞれ公開鍵と秘密鍵を表す.

$$m_1 + m_2 = \mathrm{Dec}_{sk}(\mathrm{Enc}_{pk}(m_1) \oplus \mathrm{Enc}_{pk}(m_2)) \tag{1}$$

2.2 XGBoost

XGBoost [6] はスパースな入力に対する高速アルゴリズムや, 近似された分割候補点を用いる損失のない高速化アルゴリズムなどが実装された最新の GBDT である. GBDT は複数の決定木で構成され, その更新は分割点と葉の重みの決定によって行われる. XGBoost では入力特徴値を用いる代わりに損失関数の勾配情報を用いて更新を行う. 勾配情報 $g_i,\ h_i$ は以下の式 (2)(3) のように計算される.

$$g_i = \partial_{\hat{y}_i^{(k-1)}} l\left(y_i, \hat{y}_i^{(k-1)}\right) \tag{2}$$

$$h_i = \partial^2_{\hat{y}_i^{(k-1)}} l\left(y_i, \hat{y}_i^{(k-1)}\right) \tag{3}$$

ここで, $l\left(y_i, \hat{y}_i\right)$ は損失関数 [6] であり, 現在のモデルの予測と真のラベルの誤差から計算される. 従来の GBDT では, 勾配情報を不純度として用いて分割点を決定し, 誤差を最小化するように葉の重みを決定する. しかし, XGBoost では, 以下の式 (4) に示されたコスト関数 $\mathcal{L}^{(k)}\left(f_k\right)$ を最小化するように分割点を決定し, 葉の重み $\hat{\omega}_j$ は式 (5) のように二次近似したコスト関数から解析的に決定される.

$$\mathcal{L}^{(k)}\left(f_k\right) = \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}_i\right) + g_i f_k\left(x_i\right) + \frac{h_i}{2} f_k^2\left(x_i\right)\right] + \Omega\left(f_k\right) \tag{4}$$

$$\hat{\omega}_j = -\frac{\Sigma_{i\in I_j} g_i}{\Sigma_{i\in I_j} h_i + \lambda} \tag{5}$$

$$score = \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda} \tag{6}$$

近似された葉の重み $\hat{\omega}_j$ から, 式 (4) の損失関数最小化は式 (6) に示された score の最大化に帰着され, すべての分割候補点から score を最大化する点を分割点として決定する.

2.3 関連研究

2.3.1 Yang らの協調学習 [7]

Yang らは複数データ所有者と中央サーバーによって構成されるスキームにおいて, 損失関数の勾配情報をモデルの更新情報として中央サーバーに集約することで協調学習を達成した. 勾配情報は共通の形の行列へ変換した上で中央サーバーに送信される. この手法の安全性はオリジナルデータの代わりに勾配情報を集約するという点のみで保証されている. しかし, 中央サーバは各データ所有者の勾配情報を集約するが, そのときに求められる勾配情報行列で決定木が学習可能となる. そのため, このような勾配情報行列を中央サーバや各データ所有者に送ることは望ましいとは言えない.

2.3.2 Zhao らの協調学習 [8]

Zhao らは複数データ所有者のみで構成されたスキームにおいて, 共通のモデルを順番に学習することで GBDT モデルの協調学習を達成した. 具体的には, 各データ所有者は自身が持つデータセットでモデルを更新し, 更新したモデルを次のデータ所有者に送信する. 各データ所有者は送信されたモデルと自身のデータセットのみでモデルを更新するため, 通信の必要がある情報は更新済のモデルのみである. ただし, 差分プライバシーを満たすために付加するノイズの影響から, モデルの予測精度が犠牲になる点が問題として残っている.

3. 提案手法

本研究では, XGBoost に協調学習を導入するための二つの手法を提案する. それぞれを, 準同型暗号を用いた協調学習, 逐次分散学習を用いた協調学習と呼ぶ. どちらの学習スキームも, 複数データ所有者 $U = \{u_1, u_2, u_3, ..., u_D\}$ と, 中央サーバー S によって構成される. また, 本稿を通して表 1 に定義した記号群を用いる. 本章では, これらの提案手法に関する説明を順に行う.

3.1 準同型暗号を用いたプライバシー保護 XGBoost

勾配情報行列は決定木を構築可能であるため, $u \in U$ が計算した勾配情報行列は $u \in U$ の統計情報であると考えられる. Yang らの手法では, S は全ての $u \in U$ の勾配情報行列を得ることができる. また, $D = 2$ の場合, $u \in U$ も他者の勾配情報行列を得ることができる. これらを解決するために, 秘密計算を用いた *All Aggregation* と, 更に復元抽出を用いて集約を行う *Random Aggregation* の二つの集約手法を提案する. 提案手法の概要を**図 1** に示す. 本章では, Yang らの既存手法である勾配情報の行列への変換手法と, 提案手法である勾配情報行列の集約手法について記述する.

表 1 本稿で用いる記号の説明

記号	説明
S	中央サーバー
D	データ所有者数
U	全データ所有者集合
T_i	i 回目の更新を終えたモデル
$iter$	モデルの目標更新回数
$\mathrm{Enc}_{pk}(\cdot)$	公開鍵 pk を用いた暗号化
$\mathrm{Dec}_{sk}(\cdot)$	秘密鍵 sk を用いた復号

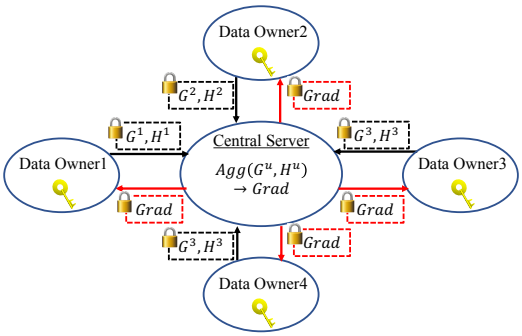

図 1 準同型暗号を用いたプライバシー保護 XGBoost の概略図. $\mathbf{G}^u$, $\mathbf{H}^u$ は勾配情報行列を, $Grad$ は勾配情報行列を集約した結果を表す.

3.1.1 勾配情報の行列への変換

Yang らは勾配情報を集約するために, 全ての $u \in U$ の勾配情報を同じ形の勾配情報行列に変換する手法を提案した. 具体的には, $u \in U$ の勾配情報は $(split \times d)$ の行列 $\mathbf{G}^u$, $\mathbf{H}^u$ へ変換される. ここで, $split$ は分割候補数を, d はデータセットの特徴の数をそれぞれ表す. 具体的には次のように計算される. 最初に, $u \in U$ は自身のデータの特徴値を $1 \sim split$ の値へ等間隔で離散化する. 次に, 勾配情報行列の各要素を以下の式 (7)(8) に従って計算する. ここで, $G_{v,k}, H_{v,k}$ は勾配情報行列の v 行 k 列目の要素を, $s_{v,k}$ は特徴 k の v 番目に大きな離散化特徴値を, $x_{i,k}$ は元データの特徴 k の i 番目のサンプルに該当する特徴値をそれぞれ表す. Yang らの手法と異なり, これらは準同型暗号によって暗号化された後に S に送信される.

$$G_{v,k} = \Sigma_{i\in\{i|s_{v,k}\geq x_{i,k}\geq s_{v-1,k}\}} g_i \tag{7}$$

$$H_{v,k} = \Sigma_{i\in\{i|s_{v,k}\geq x_{i,k}\geq s_{v-1,k}\}} h_i \tag{8}$$

3.1.2 勾配情報行列の集約手法

S が勾配情報行列を集約する手法として最も単純な考えは総和を計算することである. $\mathrm{Enc}_{pk}(\mathbf{G}^u)$, $\mathrm{Enc}_{pk}(\mathbf{H}^u)(u \in U)$ は加法準同型性を持つため, S は以下の式 (9)(10) に従っ

Algorithm 1:
準同型暗号を用いたプライバシー保護 XGBoost

for $i \leftarrow 1$ **to** *iter* **do**
  $u \in U$ は T_{i-1} から $\mathbf{G}^u$, $\mathbf{H}^u$ を計算し暗号化する.
  $u \in U$ は $\mathrm{Enc}_{pk}(\mathbf{G}^u)$, $\mathrm{Enc}_{pk}(\mathbf{H}^u)$ を S に送信する.
  if *All Aggregation* を使用 **then**
    S は $\mathrm{Enc}_{pk}(\mathbf{G}^{all})$, $\mathrm{Enc}_{pk}(\mathbf{H}^{all})$ を計算する.
    S は全ての $u \in U$ に計算結果を送信する.
  end
  if *Random Aggregation* を使用 **then**
    S は $\mathrm{Enc}_{pk}(\mathbf{G}^{rand})$, $\mathrm{Enc}_{pk}(\mathbf{H}^{rand})$ を計算する.
    S は全ての $u \in U$ に計算結果を送信する.
  end
  $u \in U$ は復号結果から T_i を学習する.
end

て暗号状態で総和を計算することができる. この手法を本稿では *All Aggregation* と呼ぶ.

$$\mathrm{Enc}_{pk}(\mathbf{G}^{all}) = \sum_{u \in U} \mathrm{Enc}_{pk}(\mathbf{G}^u) \tag{9}$$

$$\mathrm{Enc}_{pk}(\mathbf{H}^{all}) = \sum_{u \in U} \mathrm{Enc}_{pk}(\mathbf{H}^u) \tag{10}$$

All Aggregation は秘密計算により S に対する秘匿性を達成したが, $D = 2$ の場合や U の間で共謀関係が存在するとき, $u \in U$ は $\mathbf{G}^{all}$, $\mathbf{H}^{all}$ から他者の勾配情報行列を得る事ができる. これを解決するため, $\mathbf{G}^{all}$, $\mathbf{H}^{all}$ の代わりに, 無作為な復元抽出を用いて集約を行う $\mathbf{G}^{rand}$, $\mathbf{H}^{rand}$ を用いる *Random Aggregation* を提案する. これによって, $D = 2$ の場合や U の間に共謀関係が存在する場合でも, 他者の $\mathbf{G}^u$, $\mathbf{H}^u$ を得ることを困難にする. $\mathbf{G}^{rand}$, $\mathbf{H}^{rand}$ は以下の式 (11)(12) に示すように, S が送信された勾配情報行列の中から無作為に復元抽出を行なった集合 U' に対して和をとることで計算される.

$$\mathrm{Enc}_{pk}(\mathbf{G}^{rand}) = \sum_{u \in U'} \mathrm{Enc}_{pk}(\mathbf{G}^u) \tag{11}$$

$$\mathrm{Enc}_{pk}(\mathbf{H}^{rand}) = \sum_{u \in U'} \mathrm{Enc}_{pk}(\mathbf{H}^u) \tag{12}$$

3.1.3 学習アルゴリズム

準同型暗号を用いたプライバシー保護 XGBoost の学習アルゴリズムを Algorithm 1 に示す. 最初に全ての $u \in U$ が前回までのモデルから勾配情報を計算する. 次に, それを用いて $\mathbf{G}^u$, $\mathbf{H}^u$ を計算し, 暗号化してから S へ送信する. S は暗号状態で勾配情報行列を集約し, その結果を全ての $u \in U$ に送信する. 最後に, $u \in U$ はこれを復号した結果からモデルを更新する.

3.2 逐次分散学習を用いたプライバシー保護 XGBoost

Zhao らは共通の GBDT モデルを各データ所有者が逐次的に更新することで協調学習を達成した. 本稿では, この手法を逐次分散学習と呼ぶ. また, Zhao らは差分プライバシーのためのノイズ付加により, モデルの性能を犠牲にしていた. しかし, 本研究では, モデルの情報はデータ所有者間のみで共有することを想定している. そのため, 差分プライバシーを保証する必要性はあまりなく, モデルの性能に重点を置くべきであると考えた. 以上から, Zhao らの逐次分散学習において, 性能を犠牲にする差分プライバシーを導入せずともプライバシー保護を可能とする, より実用的な学習スキームを提案する. 以降, この提案手法を逐次分散学習を用いたプライバシー保護 XGBoost と呼ぶ.

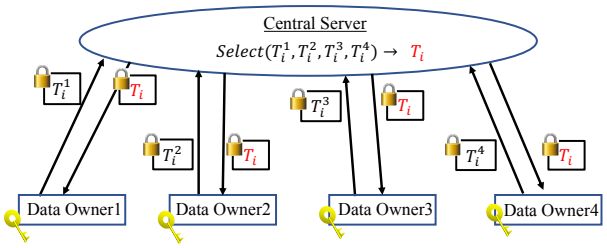

図 2 逐次分散学習を用いたプライバシー保護 XGBoost の概略図. *Select* は選択アルゴリズム, T_i^k は i 回目の更新をデータ所有者 k が行ったモデル, T_i は i 回目のモデルとして決定されたモデルを表す.

3.2.1 学習アルゴリズム

Zhao らのスキームは複数データ所有者のみで構成され, あらかじめ定められた順序でモデルの更新を行う. そのため, 各更新を行ったデータ所有者を特定することができる. ここで, GBDT を構成するそれぞれの決定木は単一のデータ所有者から学習され, 一般に決定木は統計情報であると考えられることから, 各データ所有者は学習済みモデルから他者の統計情報を得ることができる. 本研究では差分プライバシーを適用しないことを考慮すると, これらの情報が漏洩することは好ましくないと考えられる.

以上の問題を解決するため, 中央サーバー S と複数データ所有者 U によって構成される協調学習スキームを提案する. 逐次分散学習を用いたプライバシー保護 XGBoost の概略を図 3.2.1 に表す. ここで, T_i^k は i 回目の更新を u_k によって行われたモデルを表す. Zhao らの手法と同様に, $u \in U$ は自身のデータセットのみを用いてモデルを更新する. Zhao らの手法と異なるのは, 通信の有無による更新を行ったデータ所有者の特定を防ぐために, $u \in U$ は順番に更新行うのではなく毎回全員が更新を行う. これらのモデルは暗号化した状態で S に集約され, 後述するアルゴリズムによって一つのモデルが選択される.

具体的な学習アルゴリズムを Algorithm 2 に示す. 最初に, $u \in U$ は自身のデータから T_1^u を学習し, 暗号化した後にそれを S に送信する. S は後述する選択アルゴリズムを用いて, 複数の暗号化されたモデルの中から $\mathrm{Enc}_{pk}(T_1)$ を決定し, 全ての $u \in U$ に送信する. 最後に, $u \in U$ は T_1 と

Algorithm 2:
逐次分散学習を用いたプライバシー保護 XGBoost

for $i \leftarrow 1$ **to** iter **do**
  if $i = 1$ **then**
    $u \in U$ は自身のデータを用いて T_i^u を学習する.
  end
  else
    $u \in U$ は T_{i-1} から T_i^u を学習する.
    $u \in U$ は T_i^u を暗号化する.
    $u \in U$ は $\mathrm{Enc}_{pk}(T_i^u)$ を S に送信する.
  end
  S は選択アルゴリズムに従い $\mathrm{Enc}_{pk}(T_i)$ を選択する.
  S は $\mathrm{Enc}_{pk}(T_i)$ を $u \in U$ に送信する.
end

Algorithm 3:
Gradient Select を用いた場合の学習アルゴリズム

for $i \leftarrow 1$ **to** iter **do**
  $u \in U$ は T_{i-1} から T_i^u を学習し暗号化する.
  $u \in U$ は G_{ave} を計算する.
  $u \in U$ は $\mathrm{Enc}_{pk}(T_i^u)$, G_{ave} を S に送信する.
  if $i <= D$ **then**
    $u \in U$ は $\mathrm{Enc}_{pk}(T_i^u)$ を S に送信する.
    S は *Random Select* に従い $\mathrm{Enc}_{pk}(T_i)$ を選択する.
  end
  else
    S は G_{ave} の降順に U から部分集合 U' を得る.
    S は U' の中から無作為に $\mathrm{Enc}_{pk}(T_i)$ を選択する.
  end
  S は $\mathrm{Enc}_{pk}(T_i)$ を $u \in U$ に送信する.
end

自身のデータから T_2^u を学習する. 以上の流れを繰り返すことでモデルの学習を行う.

3.2.2 均一性とランダム性を考慮したデータ所有者の選択アルゴリズム

全ての $T_i^u (u \in U)$ の中から T_i を選択するための最も単純なアルゴリズムは無作為に選択することである. しかし, この場合選択される $u \in U$ に偏りが生まれる可能性があり, 学習されるモデルの性能に悪影響を及ぼすと考えられる. 従って, $u \in U$ を均一に選択する必要がある. ここで, Zhao らの手法のように, あらかじめ定められた順序で選択してしまうと各データ所有者は同じ回数おきに順番が回ってくる. この場合, 同じデータ所有者が学習した木の集合を簡単に得ることができるため好ましくない. 以上から, 無作為性と均一性を両立するため, 一巡毎に無作為に並べ替えた順序で選択するアルゴリズム *Random Select* を提案する. 具体的には, 最初に S が U の要素を無作為に並べ替えた順序を作成する. 次に, 一巡するまでその順序で $u \in U$ を選択する. 一巡した後, 再度無作為に並べ替えた順序を作成し, 上記の流れを繰り返す.

3.2.3 誤分類率に基づくデータ所有者の選択アルゴリズム

S が $u \in U$ を均等に選択すると U の各要素で学習の進行が異なる場合, 収束が遅くなったりモデルの性能に悪影響を及ぼす恐れがある. それゆえ, g の絶対値である $|g|$ を用いた選択アルゴリズム *Gradient Select* を提案する.

g は損失関数の勾配であるので, $|g|$ は予測値と真の値の誤差の大きさを表す. それゆえ, $u \in U$ が持つデータの $|g|$ の合計は, $u \in U$ が持つデータセットに対する誤分類率の大きさを表す. 一般に誤分類されているデータに重点を置いた方がより有益な学習を行うことができると考えられるため, $|g|$ が大きなデータ所有者を優先的に選択する方法を考える. ここで, ラベルごとのデータ数が不均衡であるデータセットの場合, $|g|$ の合計は多数派のラベルのデータに支配されてしまう. また, $|g|$ の合計は所有するデータの量に依存するため, U の各要素で所有するデータのサンプル数が異なる場合に比較が困難になる. 以上の問題を考慮して, 次式 (13) で示すように, $|g|$ のラベルごとの平均値の和 G_{ave} を選択指標として用いる.

$$G_{ave} = \frac{G_{pos}}{pos} + \frac{G_{neg}}{neg} \tag{13}$$

ここで pos, neg はそれぞれ, 正例データと負例データのサンプル数を表す. また, G_{pos}, G_{neg} はそれぞれ, 正例データの $|g|$ の合計, 負例データの $|g|$ の合計を表す.

具体的な学習アルゴリズムを Algotithm 3 に示す. 最初に *Random Select* によって学習を行う. その時, $\forall u \in U$ は $\mathrm{Enc}_{pk}(T_1^u)$ だけでなく G_{ave} も S に送信する. *Random Select* を終えた後, S は全ての $u \in U$ の G_{ave} を持っていることになる. その後, S は G_{ave} の降順に U から x 個を選択した集合 U' を得る. S はその集合 U' から無作為に $u \in U$ を選択し T_i を決定する.

4. 実験

本章では提案手法の検証を行う. 提案手法のスキームでは, U の各要素が同じ特徴空間とラベル空間を持つ独立したデータセットを所有する必要がある. そのため, オープンデータセットを水平方向に D 個に分割し, それぞれを $u \in U$ が所有するデータとして実験を行なった.

実装環境は Ubuntu 18.04, RAM 64[Gb], プログラム言語は Python と C++を用いた. 準同型暗号による秘密計算の実装はライブラリ [10] を用いた. ここで, 暗号パラメータは分割候補数 *split* に応じて変更した. 具体的には, $split = 10, 100, 1000$ では $n = 8192$, $split = 10000$ では $n = 32768$ となった. q は 128-bit セキュリティを満たすように設定した.

表 2　実験で使用するデータセットの情報と XGBoost を学習した時の予測精度.

データセット名	データ数	特徴数	予測精度
arcene [11]	118	9999	0.835
biodeg [12]	631	40	0.855
credit [13]	179363	29	0.851
german [14]	598	23	0.751

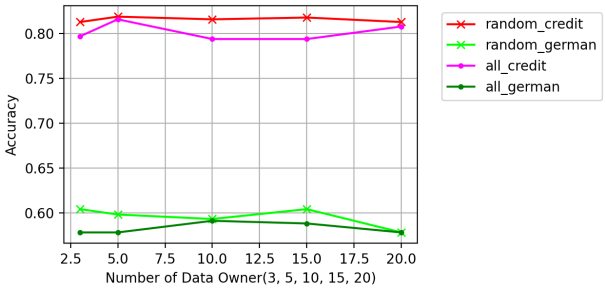

図 3　準同型暗号を用いたプライバシー保護 XGBoost におけるデータ所有者数の変化が予測精度に与える影響.

表 3　準同型暗号を用いたプライバシー保護 XGBoost における分割候補数がモデルの性能に与える影響.

分割候補数 (*split*)	10	100	1000	10000
all (credit)	0.079	0.799	0.818	0.813
random (credit)	0.079	0.808	0.819	0.813
all (german)	0.578	0.588	0.588	0.587
random (german)	0.585	0.598	0.591	0.588

4.1　データセット

表 2 に実験で用いた 4 種類の二値分類のデータセットの情報を示す．ここで，予測精度はデータセット全体を用いて XGBoost を学習した場合の予測精度を示している．また，本章では簡略化のため，Arcene Data Set [11] を'arcene'，QSAR biodegradation Data Set [12] を'biodeg'，Credit Card Fraud Detection [13] を'credit'，German Credit Data [14] を'german' とデータセット名に略称を用いる．モデルの評価指標は基本的に正解率を用いるが，'credit' のデータセットのみクラス間の不均衡を考慮して F1-score を用いている．

4.2　準同型暗号を用いたプライバシー保護 XGBoost に関する実験結果

4.2.1　データ所有者数の変化が与える影響

D を変化させた時のモデル性能への影響を検証した．結果は図 3 に示す．ここで，'all' は *All Aggregation* を，'random' は *Random Aggregation* を用いて集約した結果を示している．実験結果から，どちらの集約手法でも，D が増加してもモデルの性能が低下しないことが分かった．また，*Random Aggregation* を用いたモデルはいずれのデータセットでも高い性能を見せた．

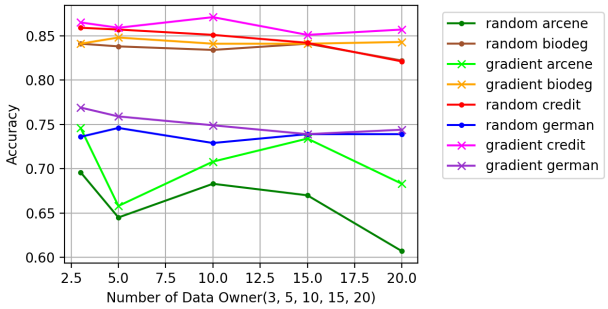

図 4　逐次分散学習を用いたプライバシー保護 XGBoost におけるデータ所有者数が予測精度に与える影響

4.2.2　分割候補数の変化が与える影響

分割候補数の変化がモデルの性能や学習時間に与える影響を検証した．結果は表 3 と表 4 に示す．表 4 には暗号化に伴う影響の検証結果も示した．

表 3 からわかるように，分割候補数を 10 としたとき予測精度は極端に低く，分割候補数を増やすと精度は向上する．しかし，分割候補数が 1000 の時と 10000 の時では予測精度はほとんど差がない．従って，分割候補数がある程度大きいことがモデルの性能に影響することが分かる．これは，分割候補数が少ない場合，離散化によって元データから多くの情報が失われるためと考えられる．

表 4 からわかるように，分割候補数の増加に応じて更新に必要な時間も増大していることが読み取れる．これは，分割候補数の増大により，分割点の決定に要する時間が増大するためであると考えられる．また，暗号化を行うことで S と U の処理時間が増大した．これは，暗号化処理の追加に伴い，S と U の両方の計算コストが増大するためと考えられる．

4.3　逐次分散学習を用いたプライバシー保護 XGBoost に関する実験結果

4.3.1　データ所有者数の変化が及ぼす影響

次に，逐次分散学習を用いたプライバシー保護 XGBoost における D の変化や選択アルゴリズムの違いがモデルの性能に与える影響を検証した．結果は図 4 に示す．ここで，'random' は *Random Select* を，'gradient' は *Gradient Select* を用いた場合を示す．*Random Select* ではデータ所有者数が増加した時に僅かに予測精度の悪化が見られたが，*Gradient Select* ではそれは見られなかった．更に，*Gradient Select* を用いた場合は，どのデータセットにおいても *Random Select* を用いた場合より高い予想精度が得られた．

4.4　提案手法間の比較

全てのデータセットの全ての D において，逐次分散学習を用いたプライバシー保護 XGBoost の方が高い予測精度を持つモデルを学習できた．これは，勾配情報行列が離散

表 4　準同型暗号を用いたプライバシー保護 XGBoost における分割候補数の変化が更新時間に及ぼす影響．各更新にかかる時間を (S による処理時間)+(U による処理時間)[ms] と表示した．

split	10	100	1000	10000
all (plain)	0.015 + 3.274	0.026 + 4.068	0.126 + 11.65	1.904 + 0.089
random (plain)	0.011 + 3.164	0.018 + 3.950	0.094 + 11.69	1.137 + 0.091
all (crypto)	1630 + 516	1636 + 515	1707 + 522	7836 + 2272
random (crypto)	849 + 514	952 + 515	1095 + 522	5467 + 2275

化された分割候補点を表現することから, 逐次分散学習を行う方がより正確な分割候補点を用いて学習できるためと考えられる．

5. 安全性分析

本章では提案手法の安全性についての検証を行う．それにあたり, 攻撃者を以下のように仮定する．

- Honest-but-curious な S
- Honest-but-curious な U
- 攻撃方法は Passive Attack のみを想定する

ここで, Passive Attack はスキームの構成要素の間に共謀関係がなく, それぞれの攻撃者は正しくスキームに従った結果得られる情報を元に攻撃することを表す．

5.1 準同型暗号を用いたプライバシー保護 XGBoost に対する安全性評価

各攻撃者は以下の情報を取得することを目的にすることとする．ここで, $\mathbf{G}^u$, $\mathbf{H}^u$ はここから決定木を学習可能であることから, $u \in U$ が持つ統計情報であるとみなすことができる．したがって, 他者に漏洩することは好ましくない．ここで, ラベルはモデルが分類を行う個人の情報を意味する．例えば, 不正検知であれば不正か正常の 2 種類のラベルが考えられる．

- S
  - $\mathbf{G}^u$, $\mathbf{H}^u(\forall u \in U)$
  - $\forall u \in U$ のラベルごとのデータ数
  - 個人のラベルや特徴値
- U
  - 自分以外の $u \in U$ の $\mathbf{G}^u$, $\mathbf{H}^u$
  - 自分以外の $u \in U$ のラベルごとのデータ数
  - 自分以外の $u \in U$ に属する個人のラベルや特徴値

また, 各攻撃者は以下の情報を得ることが想定される．ここで, 通信は安全な通信路を用いて行うこととし, 通信内容は傍受されないものとする．

- S : $\mathrm{Enc}_{pk}(\mathbf{G}^u)$, $\mathrm{Enc}_{pk}(\mathbf{H}^u)(\forall u \in U)$
- U : ($\mathbf{G}^{all}$, $\mathbf{H}^{all}$), または ($\mathbf{G}^{rand}$, $\mathbf{H}^{rand}$)

5.1.1 中央サーバーに対する安全性

S は暗号化された情報のみを得るため, 鍵の情報がない限り有益な情報を得ることはできない．

5.1.2 データ所有者に対する安全性

最初に, 自分以外の $u \in U$ の $\mathbf{G}^u$, $\mathbf{H}^u$ を得ることができるかを考察する．スキーム内に共謀関係は存在しない仮定であるため, どちらの集約手法を用いても $D > 2$ であれば目的の情報を得ることはできない．$D = 2$ で *All Aggregation* を用いた場合, $\mathbf{G}^{all}$, $\mathbf{H}^{all}$ から自身の $\mathbf{G}^u$, $\mathbf{H}^u$ を引けば目的の情報を得ることができる．しかし, $D = 2$ で *Random Aggregation* を用いた場合, $\mathbf{G}^{rand}$, $\mathbf{H}^{rand}$ は無作為に復元抽出したものの和であるため, 目的の情報を一意に得ることはできない．また, ある $u \in U$ を除いた U の間で共謀関係を許した場合も, *Random Aggregation* は目的の情報を一意に得ることを防ぐ．

自分以外の $u \in U$ のラベルごとのデータ数に関しては, 前述したように $D > 2$ であれば他者の $\mathbf{G}^u$, $\mathbf{H}^u$ を得ることはできないことから得ることはできない．

次に, 個人のラベルや特徴値が漏洩する危険性について考察する．それにあたり, 最初に個人の勾配情報から漏れる情報について考察する．二値分類の場合, 予測値を y_{pred}, 真のラベルを y とした時, 勾配情報 g, h は以下の式 (14)(15) で表される．

$$g = y_{pred} - y \tag{14}$$

$$h = y_{pred}(1 - y_{pred}) \tag{15}$$

ここで, y_{pred} は確率であるため $0 \leq y_{pred} \leq 1$ である．ここから, $y = 0$ ならば $g \geq 0$, $y = 1$ ならば $g \leq 0$ となるため, g の符号は y を表すことがわかる．

しかし, 勾配情報行列の各要素は複数のデータの g の和 $\sum g$ である．そのため, 極端なケースを除けば個人の y を得ることはできない．極端なケースとして, 単一のデータから成る勾配情報行列の要素がある場合を考えると, それは個人の y を表すことになる．つまり, データセット内で一意の離散化特徴値を持つ個人が存在し, それを特定可能な場合に個人の y が漏洩する．また, 学習されたモデルの葉の重みもまた $\sum g$ によって符号が決まるため同様の問題がある．この場合, データセット内で木の閾値条件を満たす唯一の個人が存在し, それを特定可能な場合に個人の y が漏洩する．

5.2 逐次分散学習を用いたプライバシー保護 XGBoost に対する安全性評価

各攻撃者は以下の情報を取得することを目的とする．ここで, 学習済みモデルは U のみで共有することを目的とするので S には知られたくない．また, T_i^u は $u \in U$ の統計情報と考えられるため他者に漏洩するのは好ましくない．

- S
  - モデルの情報
  - 個人のラベルや特徴値
  - $u \in U$ のラベルごとのデータ数
- U
  - 自分以外の $u \in U$ が学習した T_i^u
  - 自分以外の $u \in U$ のラベルごとのデータ数
  - 自分以外の $u \in U$ に属する個人のラベルや特徴値

また, 各攻撃者が得られる情報は以下のものが想定される．ここで, 通信は安全な通信路を用いて行うこととし, 通信内容は傍受されないものとする．

- S: $\mathrm{Enc}_{pk}(T_i^u)$ $(u \in U,\ i = \{0, 1, 2, ...\})$, G_{ave}
- $u \in U$: T_i $(i = 0, 1, 2, ...)$

5.2.1 中央サーバーに対する安全性

S が得る情報のうち, 暗号化されている情報からは鍵がない限り有益な情報を得ることはできない．また, G_{ave} は, 勾配情報の絶対値のラベルごとの平均値の和である．従って, ここから個人の y を推定することや, ラベルごとのデータ数を得ることは困難である．

5.2.2 データ所有者に対する安全性

$\forall u \in U$ は $T_i (i = 0, 1, 2, ...)$ のみを得る．最初に自分以外の $u \in U$ が学習した T_i^u を得られるかを考察する．各 T_i はそれぞれ S によって選択されるため, S 以外はそれぞれが誰に更新されたモデルであるかを識別できない．また, S への通信は毎回全ての $u \in U$ が行うため, その有無から誰が更新を行ったか特定することもできない．したがって, $D > 2$ の場合, 識別不可能性から自分以外の $u \in U$ が学習した T_i^u を得ることはできない．これは, $D = 2$ の場合, 自身の学習した木を除けば目的の情報が得られるからである．また, $D > 2$ の場合, 同様に識別不可能性から自分以外の $u \in U$ のラベルごとのサンプル数も得ることはできない．

次に, 個人の情報が漏洩する危険性について考察する．前述したように, 学習済みモデルの葉の重みは該当するデータの $\sum g$ から決定される．したがって, 極端なケースで個人の y が漏洩する可能性がある．

6. 結論

本研究では, それぞれ準同型暗号と逐次分散学習を用いた二つのプライバシー保護 XGBoost の手法を提案した．しかし, どちらの手法も極端なケースでは, 葉の重みなどから個人の機微な情報が漏洩する危険性を持つということが判明した．これらの問題はモデルの学習時に何らかの制約をかけるなどして解決する必要がある．今後は, 上記の極端なケースに対する安全性の向上や暗号鍵の共有手段の実装, G_{ave} をより安全に活用する手段の実装などにより, より実用的なスキームへと洗練していくことを課題とする．

謝辞

本稿の執筆にあたり数々の有益なご教授を賜りました, 大森敏明准教授, 国立研究開発法人情報通信研究機構の方々に御礼申しあげます．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」の研究課題「プライバシー保護データ解析技術の社会実装」(JPMJCR19F6) により得られたものです．

参考文献

[1] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y. and Theodoridis, Y.: State-of-the-art in privacy preserving data mining, Vol. 33, No. 1, ACM New York, NY, USA, pp. 50–57 (2004).

[2] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes, *International conference on the theory and applications of cryptographic techniques*, Springer, pp. 223–238 (1999).

[3] Dwork, C., McSherry, F., Nissim, K. and Smith, A.: Calibrating noise to sensitivity in private data analysis, *Theory of cryptography conference*, Springer, pp. 265–284 (2006).

[4] Yang, Q., Liu, Y., Chen, T. and Tong, Y.: Federated machine learning: Concept and applications, *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM New York, NY, USA, pp. 1–19 (2019).

[5] Friedman, J. H.: Greedy function approximation: a gradient boosting machine, *Annals of statistics*, JSTOR, pp. 1189–1232 (2001).

[6] Chen, T. and Guestrin, C.: Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794 (2016).

[7] Yang, M., Song, L., Xu, J., Li, C. and Tan, G.: The tradeoff between privacy and accuracy in anomaly detection using federated XGBoost, *arXiv preprint arXiv:1907.07157* (2019).

[8] Zhao, L., Ni, L., Hu, S., Chen, Y., Zhou, P., Xiao, F. and Wu, L.: Inprivate digging: Enabling tree-based distributed data mining with differential privacy, *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, IEEE, pp. 2087–2095 (2018).

[9] Gentry, C.: Fully homomorphic encryption using ideal lattices, *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 169–178 (2009).

[10] Microsoft Research, Redmond, W.: Microsoft SEAL (release 3.5), https://github.com/Microsoft/SEAL (2020).

[11] UCI: Arcene Data Set, https://archive.ics.uci.edu/ml/datasets/Arcene.

[12] UCI: QSAR biodegradation Data Set, https://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation.

[13] Kaggle: Credit Card Fraud Detection, https://www.kaggle.com/mlg-ulb/creditcardfraud.

[14] UCI: German Credit Data, https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).