明治大学総合数理学部

2023 年度

卒　業　研　究

# AI モデルの説明可能性 Shapley 値からの属性推定リスクの評価とその対策

学位請求者　先端メディアサイエンス学科

當麻僚太郎

# 目次

# 第 1 章

# はじめに

近年，機械学習モデルは金融や雇用などの重要な領域で活用されることが増えている．[1, 3, 4] 多くのモデルはニューラルネットワークやアンサンブルモデルなどの複雑な構造を持つため，入力に対する挙動がブラックボックスである．そのため，モデルの公平性や透明性を保証し，モデルの出力に対して説明を与えるための説明可能性技術 eXplainable AI（XAI）が注目されている [1, 2].

機械学習モデルを用いた商品サービスを提供する基盤である Machine Learning as a Service（以下，MLaaS）プラットフォームでは，様々な説明可能性技術を用いた説明を提供している．特に Shapley 値 [13] を基にした説明は，Amazon Web Services [5] や Microsoft Azure [6] などの主要な MLaaS プラットフォームで提供されている．例えば，図 1.1 では Amazon SageMaker Studio [7] を用いて各入力ベクトルに対する Shapley 値ベクトルと特徴量ごとに Shapley 値の絶対値を平均した大域的な説明を得ている．

しかし，2022 年に Luo ら [8] は Shapley 値に基づく説明から本来秘匿されているモデルへの入力属性を推論出来ることを示した．Luo ら [8] は，最小勾配法による属性推定アルゴリズム $\psi$ を提案している．一方，説明モデル $f$ に多くの機械学習アルゴリズムがあるように，$\psi$ にも多くの可能性がある．特に，モデル $f$ と $\psi$

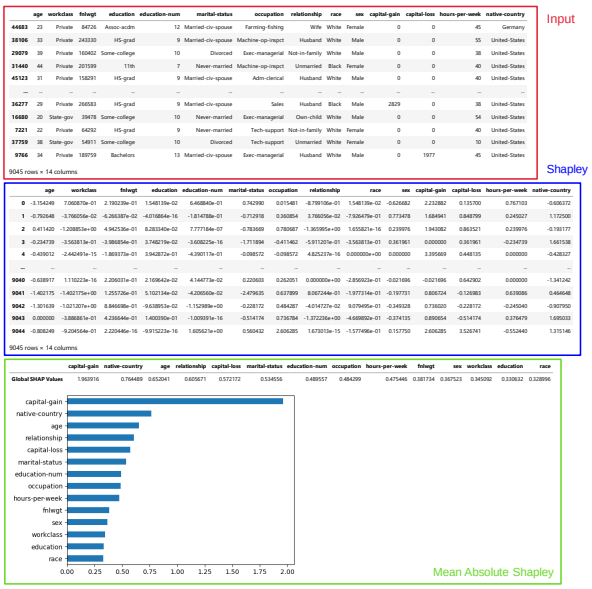



**図 1.1:** Amazon SageMaker を用いた Shapley 値の計算例

の間には相関があり，属性推定リスクの評価は自明ではない．

そこで，本研究では，Luo ら [8] の手法を基にして，各説明変数と目的変数間の相関や攻撃者が採用するアルゴリズム $\psi$ の違いに対してどのように属性推定リスクが変化するかを明らかにする．特に，$f$ と $\psi$ が線形回帰モデルのときには，Shapley 値から正確にプライベートな特徴ベクトルの推定が可能であることを示す．また，Shapley 値と同様の局所的な説明手法である LIME[14] に対するプライバシーリスクを調査し，Luo ら [8] の攻撃手法が Shapley 値以外の説明可能性技術に対して有効であることを示す．この提案方式を，3 つのデータセット Adult [10]，Bank Marketing [11], Credit Card Client [12] について適用した結果を報告する．

# 第 2 章

# 基本定義

## 2.1 Shapley 値

### 2.1.1 Shapley 値の定義

Shapley 値 [13] は協力ゲーム理論において連携プレイヤー間で利益を分配するための協調作業を定量化するために，1953 年に Shapley に提案された指標である．本研究では，$n$ 特徴の入力 $\boldsymbol{x} = (x_1, \ldots, x_n)$ に対するモデルの出力 $f(\boldsymbol{x})$ の局所的な説明として Shapley 値ベクトル $\boldsymbol{s} = (s_1, \ldots, s_n)$ を与える．

特徴量のインデックス集合を $N = \{1, 2, \ldots, n\}$，$N$ の部分集合を $S$，Shapley 値を計算するために参照するデータのサンプルを $\boldsymbol{x}^0$ とする．$S$ に対応する入力サンプルを $\boldsymbol{x}_{[S]} = ((x_{[S]})_1, \ldots, (x_{[S]})_n)$ とする．ここで，$i = 1, \ldots, n$ について，

$$(x_{[S]})_i = \begin{cases} x_i & \text{if } i \in S, \\ x_i^0 & \text{otherwise.} \end{cases} \tag{2.1}$$

例えば，$\boldsymbol{x} = (\mathbf{2}, \mathbf{5}, \mathbf{1}, \mathbf{3})$，$\boldsymbol{x}^0 = (0, 3, 2, 1)$，$S = \{2, 3\}$ としたとき，$\boldsymbol{x}_{[S]} = [0, \mathbf{5}, \mathbf{1}, 1]$ である．このとき，特徴量に対する Shapley 値 $s_i$ は，

$$s_i = \frac{1}{n!} \sum_{S \subseteq N \setminus \{i\}} |S|!(n - |S| - 1)! \left( f(\boldsymbol{x}_{[S \cup \{i\}]}) - f(\boldsymbol{x}_{[S]}) \right) \tag{2.2}$$

で定められる．$\boldsymbol{s} = \phi(\boldsymbol{x}; \boldsymbol{x}^0, f) = (s_1, \ldots, s_n)$ を Shapley 値を与える写像とする．

### 2.1.2 Shapley sampling values

式 (2.2) の時間計算量は $O(2^n)$ であるため，多くの MLaaS プラットフォームでは近似的に Shapley 値を計算するサンプリング手法を採用している．本研究では，$n!$ 個の順列のうちランダムに $v = 50$ 個サンプリングして Shapley 値 $s_i$ を計算している．

### 2.1.3 Shapley 値の計算例

入力サンプル $\boldsymbol{x} = (1.5, \text{True}, A)$ に対し，参照サンプル $\boldsymbol{x}^0 = (-0.4, \text{False}, B)$ を用いて Shapley 値を計算する例を示す．ここで，攻撃対象のモデル $f$ は表 2.1 の通りに出力する．

**表 2.1:** 入力データに対するモデル $f$ の出力

| | $x_1$ | $x_2$ | $x_3$ | $f(x)$ |
|---|---|---|---|---|
| $\boldsymbol{x}$ | 1.5 | True | $A$ | 0.8 |
| $\boldsymbol{x}^0$ | −0.4 | False | $B$ | 0.6 |
| $\boldsymbol{x}_{[\{1\}]}$ | 1.5 | False | $B$ | 0.9 |
| $\boldsymbol{x}_{[\{2\}]}$ | −0.4 | True | $B$ | 0.8 |
| $\boldsymbol{x}_{[\{3\}]}$ | −0.4 | False | $A$ | 0.2 |
| $\boldsymbol{x}_{[\{1,2\}]}$ | 1.5 | True | $B$ | 1.0 |
| $\boldsymbol{x}_{[\{1,3\}]}$ | 1.5 | False | $A$ | 0.6 |
| $\boldsymbol{x}_{[\{2,3\}]}$ | −0.4 | True | $A$ | 0.4 |

このとき，Shapley 値 $\boldsymbol{s} = (s_1, s_2, s_3)$ は式 (2.2) によりそれぞれ計算される．

$$
\begin{aligned}
s_1 = &\frac{0!(3-0-1)!}{3!}(f(\boldsymbol{x}_{[\{1\}]}) - f(\boldsymbol{x}_{[\{\}]})) \\
&+ \frac{1!(3-1-1)!}{3!}(f(\boldsymbol{x}_{[\{1,2\}]}) - f(\boldsymbol{x}_{[\{2\}]})) \\
&+ \frac{1!(3-1-1)!}{3!}(f(\boldsymbol{x}_{[\{1,3\}]}) - f(\boldsymbol{x}_{[\{3\}]})) \\
&+ \frac{2!(3-2-1)!}{3!}(f(\boldsymbol{x}_{[\{1,2,3\}]}) - f(\boldsymbol{x}_{[\{2,3\}]})) \\
= &\frac{1}{3}(0.9 - 0.6) + \frac{1}{6}(1.0 - 0.8) \\
&+ \frac{1}{6}(0.6 - 0.2) + \frac{1}{3}(0.8 - 0.4) \\
= &\frac{2}{6} \approx 0.33
\end{aligned}
\tag{2.3}
$$

$s_2$，$s_3$ も式 (2.3) と同様にして，$\boldsymbol{s} = (0.33, 0.18, -0.32)$ と計算される．$\boldsymbol{s}$ の総和はおよそ 0.2 であり，これは $f(\boldsymbol{x}) - f(\boldsymbol{x}^0)$ に等しい．正の Shapley 値 $s_1$，$s_2$ に対応する属性 $x_1$，$x_2$ はモデル $f$ の出力を増加させるように働き，負の Shapley 値 $s_3$ に対応する属性 $x_3$ はモデル $f$ の出力を減少させるように働くことを示す．

また，Shapley 値の絶対値の大きさ $|s_1|$，$|s_2|$，$|s_3|$ は，各属性の変化量に対してどの程度モデルの出力が変動したかを説明している．例えば，属性 $x_1$ が −0.4 から 1.5 に変化したことによるモデル $f$ の出力の変化量は $|s_1| = 0.33$ であるのに対し，属性 $x_2$ が False から True に変化したことによるモデル $f$ の出力の変化量は $|s_2| = 0.18$ であるため，属性 $x_2$ より属性 $x_1$ の方が $f$ の出力の変化に大きく寄与していたと説明できる．

## 2.2 LIME

### 2.2.1 LIME の定義

Local Interpretable Model-agnostic Explanations (LIME) は，Ribeiro ら [14] によって提案された，Shapley 値と同様に入力サンプルごとに説明を生成する手法である．$n$ 特徴の入力 $\boldsymbol{x} = (x_1, \ldots, x_n)$ が与えられたとき，その周辺のデータに対するモデル $f$ のふるまいを，線形モデルや決定木，ルールベースなどの解釈が容易なモデル $g$ で近似する．本研究では説明モデル $g$ に線形モデル $g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ を採用しているものとし，その係

数ベクトル $\boldsymbol{w}$ が説明ベクトルとして与えられるものとする．

説明モデル $g$ を学習するための損失関数 $\mathcal{L}$ は

$$\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) = \sum_{\boldsymbol{z}, \boldsymbol{z}' \in Z} \pi_{\boldsymbol{x}}(\boldsymbol{z})(f(\boldsymbol{z}) - g(\boldsymbol{z}'))^2$$

と定義される．ここで，$\pi_{\boldsymbol{x}}(\boldsymbol{z}) = e^{-D(\boldsymbol{x},\boldsymbol{z})^2/\sigma^2}$ は距離 $D(\boldsymbol{x}, \boldsymbol{z})$ に応じた重みであり，$\sigma$ はそのパラメタである．$\boldsymbol{z}$ は $\boldsymbol{x}$ と同じ $n$ 次元ベクトルであり，$\boldsymbol{z}'$ は $\boldsymbol{z}$ の一部の特徴量を抜き出したベクトルである．$Z$ は $\boldsymbol{z}$ と $\boldsymbol{z}'$ の組の集合である．

モデル $g$ の取りうる集合を $G$，$\boldsymbol{w}$ の非ゼロ要素の数を $\Omega(g)$ としたとき，説明モデル $g$ は目的関数 $\xi(\boldsymbol{x}) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_{\boldsymbol{x}}) + \Omega(g)$ を最小化する $g^*$ が学習される．

### 2.2.2 LIME の計算例

Shapley 値の計算例と同様に，入力サンプル $\boldsymbol{x} = (1.5, \text{True}, A)$ に対して LIME を計算する例を示す．表 2.2 は，説明モデル $g$ を学習するために用いるデータ $\boldsymbol{z}$，$\boldsymbol{z}'$ である．

**表 2.2:** 説明モデル $g$ の学習に用いるデータ $\boldsymbol{z}$

| | $z_1$ | $z_2$ | $z_3$ | $z'_1$ | $z'_2$ | $z'_3$ | $f(\boldsymbol{z})$ |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{z}^1$ | 1.5 | True | $A$ | 1.5 | 1 | 1 | 0.8 |
| $\boldsymbol{z}^2$ | −0.4 | False | $B$ | −0.4 | 0 | 0 | 0.6 |
| $\boldsymbol{z}^3$ | 0.1 | False | $A$ | 0.1 | 0 | 1 | 0.3 |
| $\boldsymbol{z}^4$ | 0.8 | True | $C$ | 0.8 | 1 | 0 | 0.9 |
| $\boldsymbol{z}^5$ | −1.1 | True | $A$ | −1.1 | 1 | 1 | 0.2 |

ここで，距離を求める関数を

$$D(\boldsymbol{x}, \boldsymbol{z}) = \sqrt{(x_1 - z_1)^2 + |x_2 = z_2|^2 + |x_3 = z_3|^2}$$

とし，$\pi_{\boldsymbol{x}}(\boldsymbol{z})$ のパラメタ $\sigma = 2$ とする．説明モデル $g$ を線形モデル $g(\boldsymbol{z}') = w_1 z'_1 + w_2 z'_2 + w_3 z'_3 + b$ とすると，損失関数は

$$\begin{aligned}
\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) &= \pi_{\boldsymbol{x}}(\boldsymbol{z}^1)(f(\boldsymbol{z}^1) - g(\boldsymbol{z}'^1))^2 \\
&\quad + \pi_{\boldsymbol{x}}(\boldsymbol{z}^2)(f(\boldsymbol{z}^2) - g(\boldsymbol{z}'^2))^2 \\
&\quad + \pi_{\boldsymbol{x}}(\boldsymbol{z}^3)(f(\boldsymbol{z}^3) - g(\boldsymbol{z}'^3))^2 \\
&\quad + \pi_{\boldsymbol{x}}(\boldsymbol{z}^4)(f(\boldsymbol{z}^4) - g(\boldsymbol{z}'^4))^2 \\
&\quad + \pi_{\boldsymbol{x}}(\boldsymbol{z}^5)(f(\boldsymbol{z}^5) - g(\boldsymbol{z}'^5))^2 \\
&= (1.5w_1 + w_2 + w_3 + b - 0.8)^2 \\
&\quad + 0.2(-0.4w_1 + b - 0.6)^2 \\
&\quad + 0.5(0.1w_1 + w_3 + b - 0.3)^2 \\
&\quad + 0.7(0.8w_1 + w_2 + b - 0.9)^2 \\
&\quad + 0.2(-1.1w_1 + w_2 + w_3 + b - 0.2)^2
\end{aligned} \tag{2.4}$$

である．ここで，$w_1$，$w_2$，$w_3$，$b$ で $\mathcal{L}(f, g, \pi_{\boldsymbol{x}})$ を偏微分して，

$$\frac{\partial}{\partial w_1}\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) = 5.94w_1 + 3.66w_2 + 2.66w_3 + 3.6b - 3.24 = 0$$

$$\frac{\partial}{\partial w_2}\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) = 3.68w_1 + 3.8w_2 + 2.4w_3 + 3.8b - 2.94 = 0$$

$$\frac{\partial}{\partial w_3}\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) = 2.66w_1 + 2.4w_2 + 3.4w_3 + 3.4b - 1.98 = 0$$

$$\frac{\partial}{\partial b}\mathcal{L}(f, g, \pi_{\boldsymbol{x}}) = 3.62w_1 + 3.8w_2 + 3.4w_3 + 5.2b - 3.48 = 0$$

このとき，$w_1 \approx 0.23, w_2 \approx 0.13, w_3 \approx -0.30, b \approx 0.61$ であり，説明ベクトルとして $\boldsymbol{w} = (0.23, 0.13, -0.30)$ が得られる．

## 2.3 Shapley 値と LIME の説明可能性

Shapley 値と LIME の計算例において説明ベクトルはそれぞれ $\boldsymbol{s} = (s_1, s_2, s_3) = (0.33, 0.18, -0.32)$，$\boldsymbol{w} = (w_1, w_2, w_3) = (0.23, 0.13, -0.30)$ であった．それぞれを図 2.1 に示す．$s_1 > s_2 > 0 > s_3$，$w_1 > w_2 > 0 > w_3$ であり，似た説明が得られた．

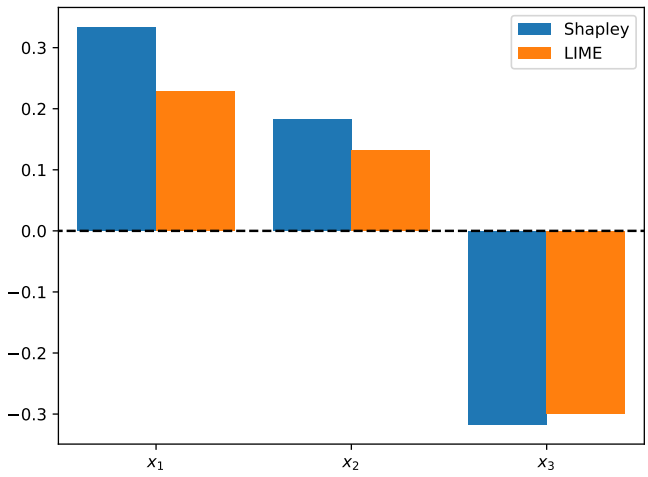



**図 2.1:** 入力サンプル $\boldsymbol{x} = (1.5, \text{True}, A)$ に対する Shapley 値と LIME による説明ベクトル

しかし，Shapley 値の絶対値 $|s_1|$，$|s_3|$ によると特徴量 $x_1$，$x_3$ はモデル $f$ の出力に同程度の影響を与えているが，LIME による説明では $|w_1|$ と $|w_3|$ は異なっている．Shapley 値と LIME の計算例におけるモデルは $f = \frac{1}{1+\exp(-x_1-x_2-x_3)}$ としている．ここで，True，False を 1，0 に，$A$，$B$，$C$ を $-1$，0，1 にエンコードしている．入力サンプルは $\boldsymbol{x} = (1.5, \text{True}, A)$ であるため，$|x_1| > |x_3| > |x_2|$ の順に重要度が並ぶような説明が最も理想的である．従って，この計算例においては，LIME より Shapley 値の方がより良い説明を得ている．

また，Shapley 値と LIME は，入力サンプル $\boldsymbol{x}$ とモデル $f$ に対して計算するときの説明の安定性が異なる．Shapley 値による説明ベクトル $\boldsymbol{s}$ は，参照サンプル $\boldsymbol{x}^0$ によって定まる．そのため，複数の参照サンプルから

算出した Shapley 値を平均して $\boldsymbol{s}$ を計算することで安定した説明を得られる．一方で，LIME による説明ベクトルは説明モデル $g$ を学習するための $\boldsymbol{z}$，$\boldsymbol{z}'$ や $\pi_{\boldsymbol{x}}(\boldsymbol{z})$ のパラメタ $\sigma$ によって変化する．$\boldsymbol{z}$，$\boldsymbol{z}'$ は数を増やすことで説明モデル $g$ の学習を安定させられるが，$\sigma$ は入力 $\boldsymbol{x}$ に対する学習データ $\boldsymbol{z}$ の距離に応じた重みパラメタであり，通常は固定して計算が行われる．最適な $\sigma$ を決定することは困難であるため，LIME による説明は事前に決定された $\sigma$ によって定まる．決められた $\sigma$ の正当性は明らかでなく，LIME による説明の正当性を示すことができない．従って，Shapley 値の方がより安定した説明である．

# 第 3 章

# 基本原理

## 3.1 Feature Inference Attack on Shapley Values [8]

### 3.1.1 システムモデル

Luo ら [8] は，サービス事業者が機密の学習データセット $\mathcal{X}_{train}$ に基づいてブラックボックスモデル $f$ を訓練し，MLaaS プラットフォーム上に展開するシステムモデルを仮定している．その実験概要図を図 3.1 に示す．

ユーザはプライベートな入力サンプル $\boldsymbol{x}$ を送信し，モデルの出力 $\hat{\boldsymbol{y}} = (y_1, \ldots, y_c)$ と $n$ 個の説明値のベクトル $\boldsymbol{s} = (s_1, \ldots, s_n)$ を得る．ただし，$c$ は正解ラベルの数である．$c > 2$ のとき対応する説明ベクトルは本来 $c$ 個分得られるが，ここでは $c = 1$ とする．

### 3.1.2 攻撃者

攻撃者は $\mathcal{X}_{train}$ と同じ分布に従う補助データセット $\mathcal{X}_{aux}$ を持っていると仮定する．全ての $\boldsymbol{x}_{aux} \in \mathcal{X}_{aux}$ をモデル $f$ に送信し，対応する説明データ $\mathcal{S}_{aux}$ を得る．そして，$\psi : \mathcal{S}_{aux} \to \mathcal{X}_{aux}$ が誤差 $L(\psi(\mathcal{S}_{aux}), \mathcal{X}_{aux})$ を最小化するように訓練する．プライベートな入力 $\boldsymbol{x}$ の推測値は，与えられた Shapley 値 $\boldsymbol{s}$ を用いて $\hat{\boldsymbol{x}} = \psi(\boldsymbol{s})$ とする．攻撃者の属性推論を図 3.2 に示す．このアルゴリズムを Algorithm 1 に示す．

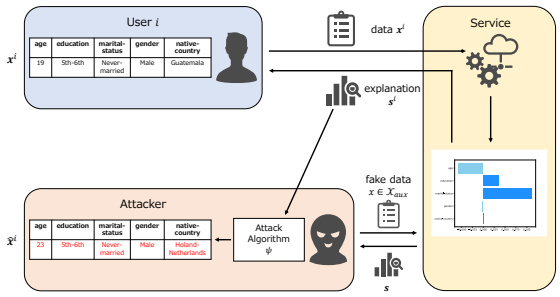



**図 3.1:** 全体概要図

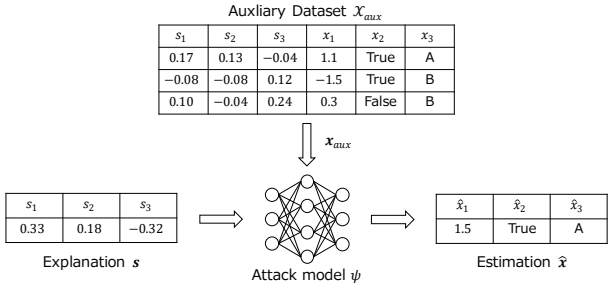



**図 3.2:** 属性推論攻撃の概要図

**Algorithm 1** 補助データセットを用いた推定 [8]

**Input:** ブラックボックスモデル $f$，補助データセット $\mathcal{X}_{aux}$，学習率 $\alpha$，攻撃対象の Shapley 値ベクトル $\boldsymbol{s}$
**Output:** 推論されたプライベートな入力 $\hat{\boldsymbol{x}}$

$\mathcal{S}_{aux} \leftarrow \phi(\mathcal{X}_{aux}; f)$
$\theta_\psi \leftarrow \mathcal{N}(0, 1)$
**for** each epoch **do**
  **for** each batch **do**
    $loss \leftarrow 0$
    $B \leftarrow$ randomly select a batch of samples
    **for** $j \in 1, \ldots, |B|$ **do**
      $(\hat{\boldsymbol{x}}_{aux})^j \leftarrow \psi((\boldsymbol{s}_{aux})^j; \theta_\psi)$
      $loss \leftarrow loss + L((\hat{\boldsymbol{x}}_{aux})^j, (\boldsymbol{x}_{aux})^j)$
    **end for**
    $\theta_\psi{}' \leftarrow \theta_\psi - \alpha \nabla_{\theta_\psi} loss$
  **end for**
**end for**
$\hat{\boldsymbol{x}} \leftarrow \psi(\boldsymbol{s}; \theta_\psi)$
**return** $\hat{\boldsymbol{x}}$

### 3.1.3 属性推論攻撃の例

10 行 5 列のサンプルデータに対して属性推論攻撃を行う例を示す．データの生成は，標準正規分布に従う独立な 3 つの乱数列 $n_1$，$n_2$，$n_3$ を用いて，$x_1 = n_1$，$x_2 = n_2$，$x_3 = n_1 n_2$，$x_4 = n_2 n_3$，$y = x_1 - x_3 x_4$ とする．生成したデータを表 3.1 に示す．

データの 1〜5 行目を $\mathcal{X}_{test}$，6〜10 行目を $\mathcal{X}_{train}$ とする．Shapley 値は $\mathcal{X}_{train}$ の各行を参照サンプルとし，それぞれ求めた Shapley 値の平均，すなわち $\boldsymbol{s} = \frac{1}{5}\sum_{j=6}^{10} \phi(\boldsymbol{x}, \boldsymbol{x}^j)$ とする．

例として，$\mathcal{X}_{train}$ をフィッティングした線形回帰モデル $f$ について，$\mathcal{X}_{test}$ に対する Shapley 値 $\mathcal{S}_{test}$ を表 3.2 に示す．モデル $f$ と推定アルゴリズム $\psi$ の組み合わせに対する MAE を表 3.3 に示す．$f$ と $\psi$ がどちらも線形モデルのとき，誤差 0 で正確に入力特徴を推論出来た．

**表 3.1:** サンプルデータ

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y$ |
|---|---|---|---|---|---|
| $\mathcal{X}_{test}$ | 1.8 | 0.1 | 0.3 | −0.4 | 1.9 |
| | 0.4 | 1.5 | 0.6 | 1.0 | −0.2 |
| | 1.0 | 0.8 | 0.7 | 0.7 | 0.5 |
| | 2.2 | 0.1 | 0.3 | −0.1 | 2.2 |
| | 1.9 | 0.4 | 0.8 | 1.0 | 1.1 |
| $\mathcal{X}_{train}$（$\mathcal{X}_{aux}$） | −1.0 | 0.3 | −0.3 | −0.5 | −1.2 |
| | 1.0 | 1.5 | 1.4 | 0.1 | 0.9 |
| | −0.2 | −0.2 | 0.0 | 0.0 | −0.2 |
| | −0.1 | 0.3 | 0.0 | 0.5 | −0.1 |
| | 0.4 | −0.9 | −0.4 | −1.3 | −0.1 |

**表 3.2:** Shapley 値 $\mathcal{S}_{test}$

| | $s_1$ | $s_2$ | $s_3$ | $s_4$ |
|---|---|---|---|---|
| $\boldsymbol{x}^1$ | 1.30 | 0.02 | 0.06 | −0.04 |
| $\boldsymbol{x}^2$ | 0.28 | −0.29 | 0.18 | 0.34 |
| $\boldsymbol{x}^3$ | 0.72 | −0.13 | 0.21 | 0.26 |
| $\boldsymbol{x}^4$ | 1.59 | 0.02 | 0.06 | 0.04 |
| $\boldsymbol{x}^5$ | 1.37 | −0.04 | 0.25 | 0.34 |

**表 3.3:** モデル $f$ と推定アルゴリズム $\psi$ の組み合わせに対する MAE

| | | MAE | | | | |
|---|---|---|---|---|---|---|
| $f$ | $\psi$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | 平均 |
| 線形回帰 | 線形回帰 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 線形回帰 | 決定木 | 0.82 | 1.24 | 0.74 | 1.18 | 1.00 |
| 決定木 | 線形回帰 | 0.69 | 0.52 | 0.41 | 0.53 | 0.54 |
| 決定木 | 決定木 | 0.68 | 1.16 | 0.82 | 0.54 | 0.80 |
| 平均 | | 0.55 | 0.73 | 0.49 | 0.59 | |

## 3.2 線形モデルに対する説明可能性と属性推定リスク

**補題 1.** $f$ を線形回帰による説明モデルとする．任意の $i \in N$，$S \subseteq N \setminus \{i\}$，参照ベクトル $(x_1^0, x_2^0, \ldots, x_n^0)$ について，

$$f(\boldsymbol{x}_{[S\cup\{i\}]}) - f(\boldsymbol{x}_{[S]}) = \beta_i(x_i - x_i^0) \tag{3.1}$$

である．

**証明)** 説明モデル $f$ を $f(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$ と表すと，線形モデルのため，

$$\begin{aligned} f(\boldsymbol{x}_{[S\cup\{i\}]}) - f(\boldsymbol{x}_{[S]}) &= \beta_0 + \sum_{k\in S\cup\{i\}} \beta_k x_k + \sum_{k\in N\setminus(S\cup\{i\})} \beta_k x_k^0 \\ &\quad - \beta_0 + \sum_{k\in S} \beta_k x_k + \sum_{k\in N\setminus S} \beta_k x_k^0 \\ &= \beta_i (x_i - x_i^0) \end{aligned}$$

□

**命題 2.** $f$ を線形モデルによる説明モデル，$\psi$ を線形モデルによる推定アルゴリズムとする．$n < |\mathcal{X}_{aux}|$ のとき，$\psi$ による推定の MAE = 0 である．

**証明)** 補題 1 より，$s_i$ について

$$\begin{aligned} s_i &= \frac{1}{n!} \sum_{S\subseteq N\setminus\{i\}} |S|!(n-|S|-1)! f(\boldsymbol{x}_{[S\cup\{i\}]}) - f(\boldsymbol{x}_{[S]}) \\ &= \frac{1}{n!} \sum_{S\subseteq N\setminus\{i\}} |S|!(n-|S|-1)! \beta_i (x_i - x_i^0) \\ &= \lambda_i (x_i - x_i^0) \end{aligned}$$

ここで $\lambda_i = \frac{1}{n!} \sum_{S\subseteq N\setminus\{i\}} |S|!(n-|S|-1)!\beta_i$ をまとめた項である．したがって，$\psi$ による推定モデルは，

$$\begin{aligned} \hat{x}_i &= \alpha_0 + \alpha_1 s_1 + \cdots + \alpha_n s_n \\ &= \alpha_0 + \alpha_1(\lambda_1(x_1 - x_1^0)) + \cdots + \alpha_n(\lambda_n(x_n - x_n^0)) \\ &= \alpha_0 - \sum_{k=1}^{n} \alpha_k \lambda_k x_k^0 + \alpha_1\lambda_1 x_1 + \cdots + \alpha_n \lambda_n x_n \\ &= \gamma_0 + \gamma_1 x_1 + \cdots + \gamma_n x_n \end{aligned}$$

と，$x_1, \ldots, x_n$ の線形式で与えられる．ただし，ここで $\gamma_i = \alpha_i\lambda_i$，$\gamma_0 = \alpha_0 - \sum_{k=1}^{n} \alpha_k\lambda_k x_k^0$ をまとめた項である．

$\mathcal{X}_{aux}$ が十分に大きく，$n+1$ 以上の行数があるならば，最小二乗法により，誤差なく $\gamma_1, \ldots, \gamma_n$ が算出される． □

結果のところ，線形式によるモデルの説明データを提供すると，属性推定リスクが上がることを意味している．

しかし，LIME による説明ベクトルから属性推定を誤差なく行うことは出来ない．LIME の説明モデル $g(\boldsymbol{z}') = \boldsymbol{w}^T\boldsymbol{z}' + b$ は，$f$ が線形モデルであり $\boldsymbol{z} = \boldsymbol{z}'$ となる単純な例において，$g$ の学習のための $\boldsymbol{z}$，$\boldsymbol{z}'$ が十分にあるとき，$g(\boldsymbol{z}') = \boldsymbol{w}^T\boldsymbol{z}' + b$ は $f(x) = \boldsymbol{\beta}^T\boldsymbol{x} + \beta_0$ に近似される．このとき，全ての入力 $\boldsymbol{x}$ について同じ説明ベクトル $\boldsymbol{w}$ が得られるため，$\boldsymbol{w}$ から $\boldsymbol{x}$ を誤差なく推定することは出来ない．また，LIME は $f$ が線形かつ説明モデル $g$ が線形のとき，有用な説明を提供することが出来ない．

# 第 4 章

# 提案方式

想定する攻撃は先行研究と同様に，図 3.1 とする．実験に用いるデータセットを表 4.1 に示す．

この設定下において，説明モデル $f$ や攻撃者が採用する最適化アルゴリズム，各説明変数と目的変数間の相関などに対する属性推定リスクを明らかにすることを目的とする．また，LIME に対する属性推定リスクを調べる．

## 4.1 評価指標

属性推定リスクの評価に用いる指標は，MAE と攻撃成功率 SR の 2 つである．

### 4.1.1 MAE

MAE（Mean Absolute Error），すなわち $\ell_1$loss は誤差の絶対値の平均を取る．$m$ 行 $n$ 列のデータセット $\boldsymbol{x}$ に対する推定データ $\hat{\boldsymbol{x}}$ の MAE は

$$\ell_1(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \frac{1}{mn} \sum_{j=1}^{m} \sum_{i=1}^{n} |\hat{x}_i^j - x_i^j| \tag{4.1}$$

で与えられる．

### 4.1.2 攻撃成功率 SR

SR（Success Rate）は攻撃によって正しく推定された入力特徴量の割合を表す．質的変数に対しては推定カテゴリが一致している時，量的変数に対しては推定値と真の値との誤差の絶対値が閾値以下である時成功と判定する．$\boldsymbol{x}$ と推定 $\hat{\boldsymbol{x}}$ の SR は

$$SR(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \frac{success(\hat{\boldsymbol{x}}, \boldsymbol{x})}{mn} \tag{4.2}$$

**表 4.1:** 使用データセット

| データセット | レコード数 | クラス | 特徴量 |
|---|---|---|---|
| Adult [10] | 48, 842 | 2 | 14 |
| Bank Marketing [11] | 45, 211 | 2 | 16 |
| Credit Card [12] | 30, 000 | 2 | 24 |

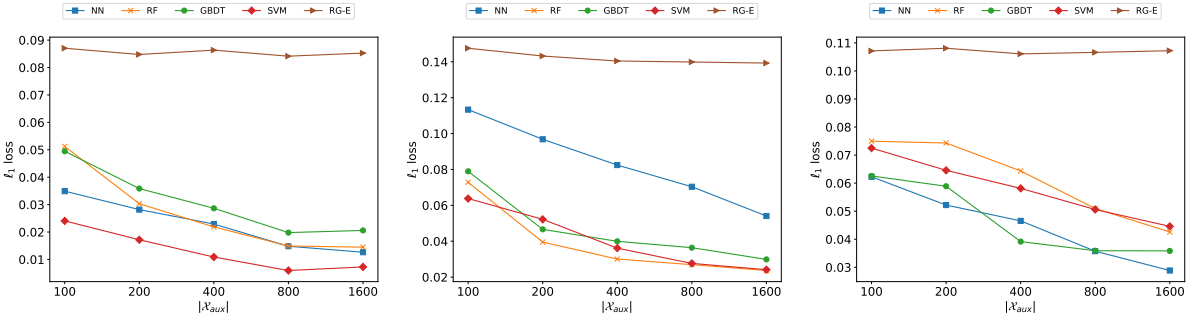
で与えられる．ここで，推定に成功した入力特徴の個数を $success(\hat{\boldsymbol{x}}, \boldsymbol{x})$ とする．

## 4.2 実験方法

### 4.2.1 実験 1

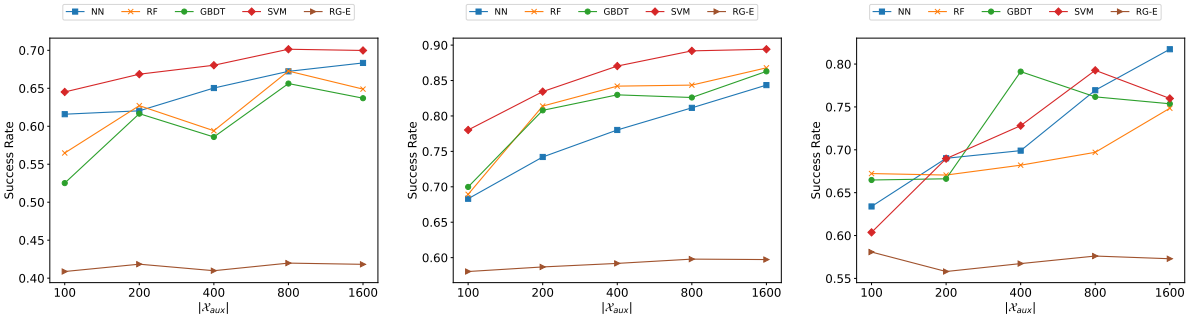
Shapley 値と LIME 値のブラックボックスモデル $f$ に対する属性推定リスクを調べる．攻撃対象となるブラックボックスモデル $f$ はニューラルネットワーク（NN），ランダムフォレスト（RF），勾配ブースティング木（GBDT），カーネル SVM（SVM）の 4 種類である．NN は Pytorch[9] で実装し，$n$ 次元の入力層と $c$ 次元の出力層を持ち，ニューロン数 $2n$ の隠れ層を 2 つ持つ．活性化関数は出力層のみ softmax でそれ以外は ReLU を用いる．RF，SVM，GBDT は sklearn で実装した．RF と GBDT の木の数と最大の深さは，それぞれ $(100, 5)$，$(100, 3)$ とする．その他のパラメータは全てデフォルトの値とする．RG-E は $\mathcal{X}_{aux}$ に基づく経験分布からランダムに予測したときの結果である．

攻撃者が持つデータセット $\mathcal{X}_{aux}$ の行数を変化させたときに，どのように MAE と SR が変化するかを調べる．Shapley 値に対する属性推論の結果を図 4.1，4.2 に，LIME に対する属性推論の結果を図 4.3，4.4 に示す．



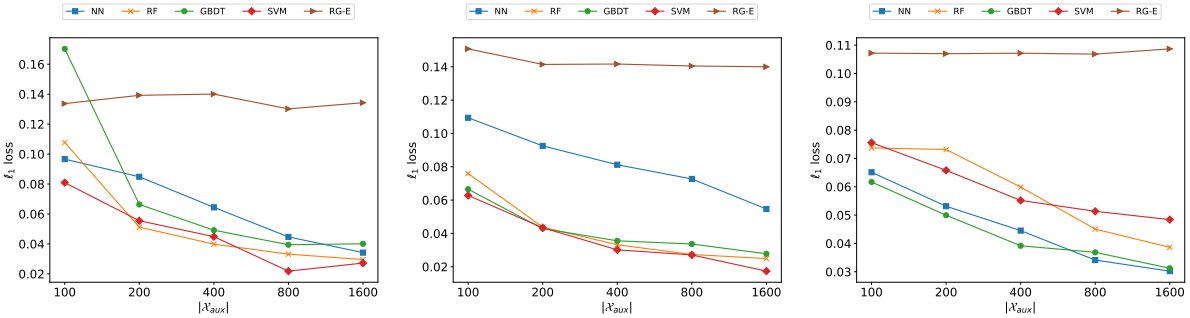

**(a)** Adult **(b)** Bank Marketing **(c)** Credit Card

**図 4.1:** 補助データセットの大きさ $|\mathcal{X}_{aux}|$ と各モデル $f$ についての Shapley 値からの属性推論攻撃の MAE



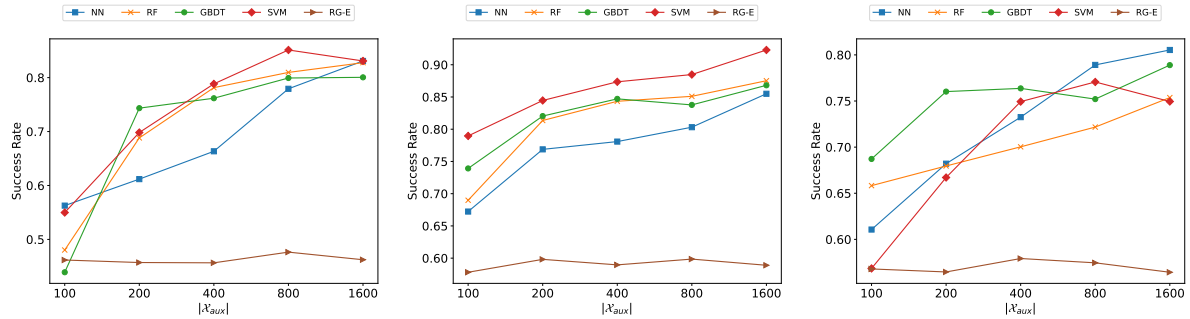

**(a)** Adult **(b)** Bank Marketing **(c)** Credit Card

**図 4.2:** 補助データセットの大きさ $|\mathcal{X}_{aux}|$ と各モデル $f$ についての Shapley 値からの属性推論攻撃の SR




**(a)** Adult **(b)** Bank Marketing **(c)** Credit Card

**図 4.3:** 補助データセットの大きさ $|\mathcal{X}_{aux}|$ と各モデル $f$ についての LIME からの属性推論攻撃の MAE




**(a)** Adult **(b)** Bank Marketing **(c)** Credit Card

**図 4.4:** 補助データセットの大きさ $|\mathcal{X}_{aux}|$ と各モデル $f$ についての LIME からの属性推論攻撃の SR

### 4.2.2 実験 2

線形モデル $f$ に対する Shapley 値からの攻撃が成功することを確認する実験を行う．Adult データセット [10] から $f$ をフィッティングし，それに対して攻撃アルゴリズムを用いて属性推論を行う線形回帰モデル $\psi$ を学習する．その結果として，$\mathcal{X}_{test}$ の各列に対する SR を表 4.2 に示す．ただし，$|\mathcal{X}_{aux}| = 1600$ とする．

**表 4.2:** $\mathcal{X}_{test}$ の各列に対する SR

| 列 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SR | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

### 4.2.3 実験 3

攻撃者が採用する最適化アルゴリズムを変化させたときの属性推定リスクを調査する．攻撃者の採用する推定アルゴリズムとは，攻撃者が Algorithm 1 において攻撃モデル $\psi$ のパラメータ $\theta_\psi$ を更新する式 (4.3)

$$\theta_\psi \leftarrow \theta_\psi - \alpha \nabla_{\theta_\psi} loss \tag{4.3}$$

の変種を意味する．本研究では，勾配降下法ベースの最適化アルゴリズムとして，SGD[15]，Momentum[16]，RMSprop[17]，Adam[18] の 4 種類を調べる．推定モデル $\psi$ は先行研究と同じく，特徴量の数 $n$ に対して隠れ層のニューロン数 $4n$，出力層のニューロン数 $n$ のニューラルネットワークとし，活性化関数は全てで sigmoid 関数を用いる．実装は Pytorch で行い，SGD の学習率 $\eta = 0.01$，Momentum（すなわち SGD のうち *momentum* $\neq 0$ のもの）の学習率 $\eta = 0.01$，*momentum* $= 0.9$ と指定したもの以外は全てデフォルトのパラメータを用いる．

実験の結果を図 4.5 に示す．

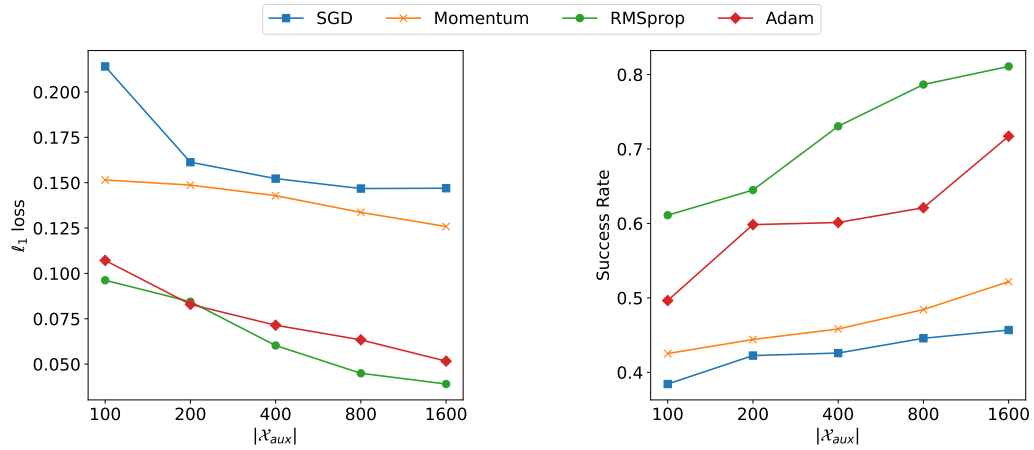



**図 4.5:** 攻撃者が採用する最適化アルゴリズムに対する，補助データセットの大きさを変化させたときの MAE と SR の変化

### 4.2.4 実験 4

データセットの各説明変数と目的変数との間の相関係数に対して，属性推定を行ったときの MAE の関係を明らかにする．相関係数の計算は，目的変数が質的変数であるため，説明変数が量的変数のときは相関比，質的変数のときは Cramer の連関係数 [**?**] で与える．説明モデル $f$ には NN を，攻撃者の属性推論アルゴリズム $\psi$ には RMSprop を用いる．各属性ごとの MAE を補助データセットやランダムデータセットの大きさを変化させて計算し，それらの平均を取ったものをその属性に対する MAE を評価する．実験の結果を図 4.6 に示す．

## 4.3 実験結果と考察

### 4.3.1 結果 1

図 4.1，4.2 において，データセットの行数 $|\mathcal{X}_{aux}|$ が増えるにしたがって $\ell_1$loss が下がり SR が上がる，すなわち，属性推定リスクが上がった．ただし，SR の値の大きさはデータセットによって異なり，属性推定リスクはデータに依存することが示唆された。

図 4.3，4.4 において，参照データの大きさ $|\mathcal{X}_{aux}|$ に対する推定リスクの傾向は，Shapley 値と同様であった．ほとんどの場合についてランダムな予測より小さい誤差で属性推定されたが，Adult データセット，$|\mathcal{X}_{aux}| = 100$，$f$ が GBDT の場合のみ Shapley 値と異なり推定誤差がランダムな予測を上回った．

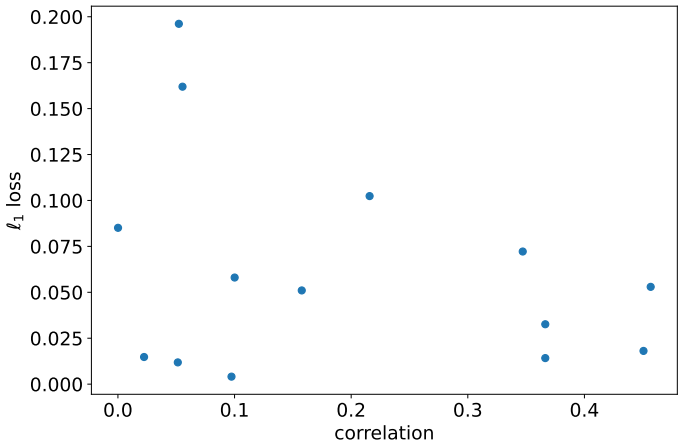



**図 4.6:** 各説明変数と目的変数間の相関係数に対する攻撃者の MAE の分布

Shapley 値と LIME それぞれについての平均 SR を表 4.3 に示す．補助データセットの大きさ $|\mathcal{X}_{aux}|$ が小さいときは Shapley 値の属性推定リスクの方がより高いが，$|\mathcal{X}_{aux}|$ が大きいときは LIME の属性推定リスクの方がより高い．これは，LIME の出力 $\boldsymbol{w}$ からの元の入力 $\boldsymbol{x}$ の推定には学習データ $|\mathcal{X}_{aux}|$ が必要であることに依る．したがって，学習データが少ない時の属性推定は Shapley 値より難しいが，学習データが多い時は LIME より Shapley 値の方が難しい．

**表 4.3:** Shapley 値と LIME の属性推定リスク比較

| | 平均 SR | |
|---|---|---|
| | $\|\mathcal{X}_{aux}\| = 100$ | $\|\mathcal{X}_{aux}\| = 1600$ |
| Shapley 値 | **0.65** | 0.77 |
| LIME | 0.62 | **0.83** |

### 4.3.2 結果 2

全ての入力属性について，SR が 1.00 となった．したがって，$|\mathcal{X}_{rand}|$ が十分に大きいとき，説明モデル $f$ と属性推定モデル $\psi$ が線形モデルであれば攻撃が出来る．

### 4.3.3 結果 3

図 8 の MAE と SR に共通して，SGD が最も属性推定の精度が低く，RMSprop が最も高い．全ての最適化アルゴリズムで，$|\mathcal{X}_{aux}|$ が増加するにつれて MAE は減少した．また，Adam と RMSprop は MAE が小さく，SGD と Momentum は MAE が大きい傾向が見られた．同様に，$|\mathcal{X}_{rand}|$ が増加するにつれて SR も増加した．Adam と RMSprop は SGD，Momentum と異なり，モデルの訓練中に学習率の調整を行う手法である．したがって，学習率の調整を行うことで属性推定の精度が高くなる．

#### 4.3.4 結果 4

相関が弱いときには MAE が大きいものも小さいものも存在していたが，相関が強いときには MAE が小さくなった．最も MAE が大きかったのは hours-per-week の列であり，その相関係数は 0.052 であった．

#### 4.3.5 質的変数に対するエンコーディング手法の影響

本研究では，データセット内の質的変数を全て One-hot エンコーディングして数値に変換している．これにより，モデル $f$ への入力は本来の 14 次元から 119 次元に変換されている．この変換を行う写像 $\epsilon$ は単射であるが全射でない．そのため，$\psi$ によって推定される 119 次元のベクトルは $\epsilon$ の値域に収まらない．これは，One-hot エンコーディングされた列は本来 0 か 1 のどちらかであるが，$\psi$ によって推定された結果は各列ごとに sigmoid 関数を通しているため，取る値は $(0, 1)$ であることで説明される．したがって，元の次元数よりも情報量の多いベクトルを推定するため，本来の属性推定リスクとは異なる評価が得られているはずである．

# 第 5 章

# おわりに

Luo ら [8] の手法に基づき，Shapley 値と LIME の属性推定リスクを調べた．オープンデータを用いた実験結果より，全ての説明モデル $f$ に対して，どちらもランダムな予測よりも高い精度で属性推定された．また，Shapley 値と LIME の双方で，補助データセットの大きさが増えるにつれて攻撃精度が増加する傾向が見られた．さらに，$f$ と $\psi$ が線形モデルのとき，Shapley 値から正確にプライベートな入力特徴量の推定が可能であることを証明した．

属性推定のリスクを抑えるために，公開する Shapley 値や LIME の値にノイズを加えることを提案する．また，2022 年に Bozorgpanah ら [19] はデータそのものを匿名加工や差分プライバシーによって保護しても，ある程度であれば Shapley 値の有用性を損なわないことを報告している．そのため，データと説明ベクトルに対する加工によって属性推定リスクを下げられることが期待される．

今後の課題として，説明ベクトルにノイズを加えたときの属性推定リスクの調査や Shapley 値と LIME 以外の説明可能性技術に対する属性推定リスクの調査が挙げられる．

# 謝辞

本研究を行うにあたって，多くの方々よりご指導いただきました．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，菊池浩明教授に深く感謝申し上げます．また，共同で付録 A の研究を行った谷口輝海さん，研究室の皆様に深く感謝の意を表するとともに，謝辞とさせていただきます．

# 参考文献


[1] Cynthia Rudin, “Stop explaining black box machine learning models forhigh stakes decisions and use interpretable models instead”, *Nature Machine Intelligence* 1, 5, 206-215, 2019.

[2] Jianbo Chen, et al. “Learning to Explain: An Information-Theoretic Perspective on Model Interpretation”, In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July*, Vol. 80. PMLR, 882-891, 2018.

[3] Sakai, Akira, Masaaki Komatsu, Reina Komatsu, Ryu Matsuoka, Suguru Yasutomi, Ai Dozen, Kanto Shozu, Tatsuya Arakaki, Hidenori Machino, Ken Asada, and et al. 2022. “Medical Professional Enhancement Using Explainable Artificial Intelligence in Fetal Cardiac Ultrasound Screening” Biomedicines 10, no. 3: 551.

[4] Zest AI, “Why picking the right AI-credit decisioning partner matters”, Zest AI Insights. (Accessed November 3, 2023. https://www.zest.ai/insights/why-zaml-makes-your-ml-platform-better )

[5] Amazon Web Services, Inc, “Amazon SageMaker Clarify Model Explainability”, Amazon SageMaker Documentation. (Accessed November 3, 2023. https://docs.aws.amazon.com/sagemaker/latest/dg/clarify-model-explainability.html )

[6] Microsoft, “Model interpretability”, Azure Machine Learning Documentation. (Accessed November 3, 2023. https://learn.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability )

[7] Amazon Web Services, Inc. “Amazon SageMaker Studio”, Amazon SageMaker Documentation. (Accessed November 3, 2023. https://docs.aws.amazon.com/sagemaker/latest/dg/studio-updated.html )

[8] Xinjian Luo, Yangfan Jiang, and Xiaokui Xiao, “Feature Inference Attack on Shapley Values”, In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS ’22), November, Los Angeles, CA, USA*. ACM, New York, NY, USA, pp.1-15, 2022.

[9] Adam Paszke, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, In Advances in Neural Information Processing Systems 32 (pp. 8024–8035). Curran Associates, Inc. 2019.

[10] Becker Barry, and Kohavi Ronny, “Adult”, UCI Machine Learning Repository. (https://doi.org/10.24432/C5XW20)

[11] Sérgio Moro, Paulo Cortez, and Paulo Rita, “A data-driven approach to predict the success of bank telemarketing”, Decision Support Systems 62, 22–31, 2014.

[12] I-Cheng Yeh and Che-hui Lien, “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients”, Expert systems with applications 36, 2, 2473–2480, 2009.

[13] Lloyd S Shapley, “A value for n-person games”, Vol. 2. Princeton UniversityPress, 303-317, 1953.

[14] Marco Ribeiro, Sameer Singh, and Carlos Guestrin, ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”, In Proceedings of the 2016 Conference of the North American Chapter of the Association

for Computational Linguistics: Demonstrations, pages 97–101, San Diego, California. Association for Computational Linguistics, 2016.

[15] Bottou, Léon, "Online Algorithms and Stochastic Approximations"., Cambridge University Press, ISBN 978-0-521-65263-6, 1998.

[16] Ilya Sutskever, James Martens, George Dahl, Geoffrey Hinton, "On the importance of initialization and momentum in deep learning", In Proceedings of the 30th international conference on machine learning (ICML-13). Vol. 28. Atlanta, GA. pp. 1139-1147, 2013.

[17] Geoffrey Hinton, "Coursera Neural Networks for Machine Learning Lecture 6", (https://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf)

[18] Diederik P. Kingma, Jimmy Ba, "Adam: A Method for Stochastic Optimization", ICLR 2015, 2015

[19] Bozorgpanah, A., Torra, V., and Aliahmadipour, L, "Privacy and Explainability: The Effects of Data Protection on Shapley Values", Technologies 10, 6, 125, 2022.

# 付録 A

# 歩容に基づく個人識別における Kinect と OpenPose の多人数同時個人識別精度

## A.1　はじめに

人の歩き方の特徴を表す歩容は解像度の低いカメラ映像からでも取得できることから，犯罪捜査などの分野において，個人識別新たな手段として，近年注目されている．歩容に基づく属性推定・個人識別手法には，深度センサやウェアラブルデバイスなどの特定のハードウェアを本人に装着する方法 [1] や，歩容のシルエット画像列などを用いて外部から観測する方法 [2] が知られている．しかし，特定のハードウェアを用いる方法は使用場面が限られ，シルエット画像列を用いる方法では服装や髪型，携帯品などの外乱の影響を受けやすいという問題がある．加えて，街中に設置された防犯カメラ映像等を用いる際には，画角内に複数の人間が映っていることが想定されるが，既存手法では主に単独での歩行についてしか評価されていなかった．

そこで，本研究では，複数人数を同時にリアルタイムで検出する機能を持つ，Kinect[5] と OpenPose[3] に注目する．Kinect と OpenPose について，何人まで同時に識別できるか，どの程度の精度で識別できるかを明らかにすることを目的とする．本研究の個人識別の構成と流れを図 A.1 に示す．

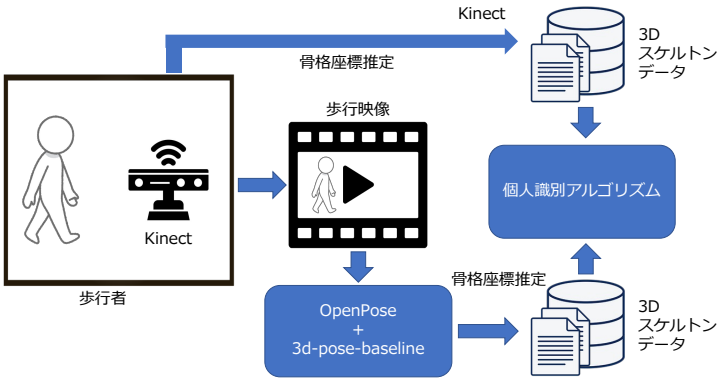



**図 A.1:** 個人識別システム構成図

## A.2　準備

### A.2.1　OpenPose

OpenPose[3] は，Zhe らによって開発されたオープンソースである. 静止画像または動画からリアルタイムに複数人数の 2D 姿勢推定を行う深層学習モデルである. 姿勢推定 (Human Pose Estimation) では，人の頭部，肩，肘，手，腰，膝，足を検出し，人がどのような姿勢を取っているかを推定する. OpenPose は深度センサなどの特別な機器を必要とせず，単眼カメラのみで姿勢の推定ができ，25 点を検出できる. 図 A.2 に推定結果のプロット例を示す. 多人数を同時に検出できる利点の一方で，人数の増加に伴う推定精度の劣化は明らかではない.

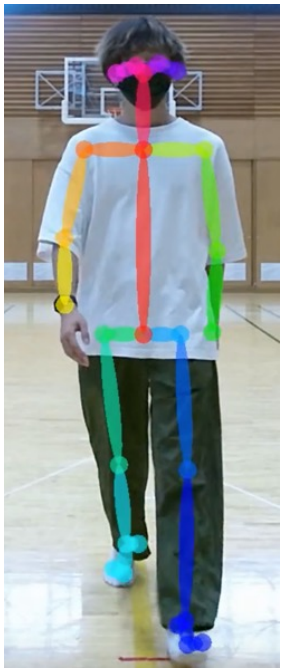


**図 A.2:** OpenPose の姿勢推定例

### A.2.2　3d-pose-baseline

3d-pose-baseline[4] は，Julieta らによって開発されたオープンソースであり，OpenPose の出力を入力とし，深度を推定する深層学習モデルである. 2 次元画像や動画から 3 次元の姿勢を推定する.

本研究では，OpenPose と 3d-pose-baseline を用いて歩行映像から 3 次元の骨格座標データを取得する. 図 A.3 に 3d-pose-baseline の出力を 3D プロットした例を示す.

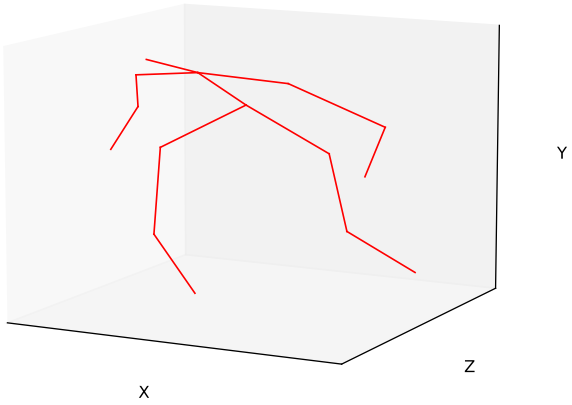



**図 A.3:** 3d-pose-baseline の 3 次元姿勢推定例

### A.2.3 Kinect

Kinect[5] は Microsoft 社によって開発されたゲーム向けデバイスである．RGB カメラと深度センサ，マイクを備え，姿勢推定や音声認識を提供する．姿勢推定で検出される関節の数は一人当たり 25 点であり，手指検出や手のポーズ検出ができる．個人を追跡する機能を持ち，最大で 6 人までを同時に検知する．

本研究では，Kinect for Windows v2[5] を用いて映像と姿勢情報の取得を行った．図 A.4 に Kinect で取得した姿勢情報の 3D プロット例を示す．

### A.2.4 推定器の比較

表 A.1 に本研究で用いた姿勢推定ツール Kinect, OpenPose, 3d-pose-baseline の機能や特性を示す．

なお，本研究では OpenPose と 3d-pose-baseline を組み合わせて姿勢の 3 次元情報を取得するため，この 2 ツールをまとめて OpenPose と呼称することとする．

### A.2.5 DTW 距離

DTW（Dynamic Time Warping）距離 [7] は，2 つの時系列データ間の類似度の 1 つである．時系列データ $S$ と $T$ の DTW は，$S$ の各データ点に対して，$T$ の最小距離の点を選び，それらの距離の総和で定める．そのため，時系列の長さが異なっていたり周期がずれていたりしても類似度を与える．

図 A.5 の時系列データ $S = (0, 2, 1)$, $T = (1, 3, 0, 1)$ の例を考えよ．表 A.2 の距離行列 $dist(S, T)$ を求める．ここで，$i$ 行 $j$ 列（$i > 0, j > 0$）の要素 $dist[i, j]$ は

$$\begin{aligned} dist[i, j] =& cost(S[i-1], T[j-1]) \\ &+ min(dist[i-1, j], dist[i-1, j-1], dist[i, j-1]) \end{aligned}$$

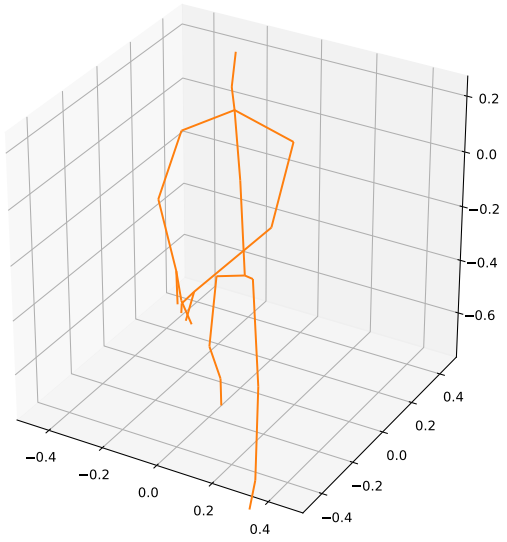



**図 A.4:** Kinect による 3D スケルトンデータの例

**表 A.1:** ツールの機能比較

| ツール | 出力 | 特徴 | 用途 | 原理 |
|---|---|---|---|---|
| Kinect | カラー画像:<br>1920×1080<br>深度画像:<br>512 × 424<br>FPS:30<br>1 人あたり 25 関節 | 処理が高速なため,<br>リアルタイムでの推論が可能.<br>最大で 6 人までを同時に追跡可能 | リアルタイム骨格検出 (3d)<br>音声認識 | 深度センサを用いた<br>RandomForest による推論 |
| OpenPose | 関節位置座標 2D データ<br>25 関節<br>各関節の推論信頼度 | ハードウェア不要<br>多人数の同時検出が可能<br>人物追跡ができない | 2D 姿勢推定 | 深層学習 |
| 3d-pose-baseline | 関節位置座標 3D データ<br>16 関節 | OpenPose の出力から深度を推定する | 3D 姿勢推定 | 深層学習 |

とする. *cost* 関数は 2 つのデータ点の距離を求める関数である．この例ではデータ点が 1 次元の値であるが，多次元のときは，マンハッタン距離もしくはユークリッド距離を用いて定める．このとき，求める時系列 $S, T$ の DTW 距離は，$dist[3, 4] = 3$ である．

### A.2.6 先行研究

三好ら [8] は，Kinect を用いて取得した歩容データから特徴量を定義し，男女の平均の差から性別の識別をした. また，推定率の高い順に特徴量を統合し，99.86% の推定率を達成した.

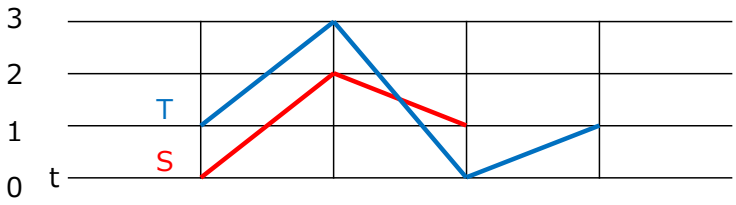



**図 A.5:** 時系列データ S と T の例

**表 A.2:** 距離行列 $dist(S,T)$ の例

| | | | | | |
|---|---|---|---|---|---|
| 1 | ∞ | 2 | 4 | 3 | 3 |
| 2 | ∞ | 2 | 2 | 4 | 5 |
| 0 | ∞ | 1 | 4 | 4 | 5 |
| - | 0 | ∞ | ∞ | ∞ | ∞ |
| S \ T | - | 1 | 3 | 0 | 1 |

阪田ら [9] は，歩容のシルエット画像列 GEI を入力とする CNN を構成し，年齢の推定を行なった. 図 A.6 に示すように，性別や大まかな年代を推定した上で，年齢の推定を行なった. その結果，平均絶対誤差が 5.83 歳となり，既存研究を大きく上回る性能を示した.

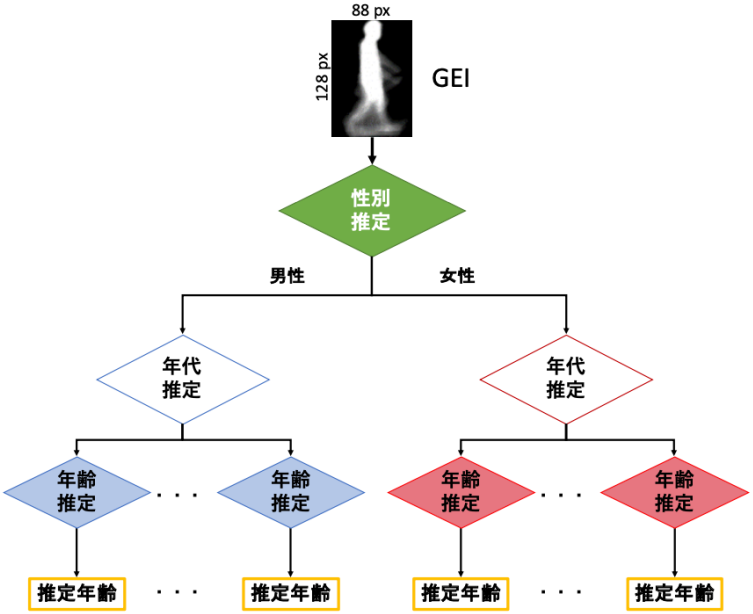



**図 A.6:** 多段階年齢推定器のフローチャート ([9] より引用)

## A.3 個人識別

### A.3.1 個人識別手法

本研究では，森ら [1] の手法に従い，個人識別を行なう. 森手法では，Kinect と OpenPose を用いて取得した関節の 3 次元座標をそれぞれ測定し，一歩分の時系列データの DTW 距離を算出して個人識別を行う. 識別手法は以下の 4 ステップから成る.

1. サイクル切り出し
2. 関節座標の相対座標化
3. DTW 距離の計算
4. 個人識別

### A.3.2 サイクル切り出し

身体の部位 $\ell$ の時刻 $t$ における 3 次元空間の絶対座標を $a_\ell(t) = (x, y, z)$ とする．ここで，時刻 $t$ の単位はフレームレートに対応する. 測定時間の絶対座標の時系列データ $\langle a_\ell(t_1), a_\ell(t_2), \ldots \rangle$ から歩行の 1 サイクル分を抽出する.

まず，時刻 $t$ の左右の足の絶対座標 $a_{LF}(t)$, $a_{RF}(t)$ から，両足の間の間隔を

$$\Delta(t) = sign \cdot \|a_{RF}(t) - a_{LF}(t)\|$$

により計算する. ここで，$sign$ は $\{-1, +1\}$ の値を取る符号であり，右足が前の状態を正とする.

次に，両足間の距離 $(\Delta(1), \ldots, \Delta(n))$ の時系列データにフーリエ変換を適用し，全周波数成分の 1/30 の低周波数成分のみを残して，残りを 0 とする．すなわち，ローパスフィルタをかけることでノイズを除去し，そのピーク間を 1 サイクルとする.

### A.3.3 関節座標の相対座標化

歩行中の各関節の座標について，身体の中心付近に位置する比較的安定した関節が原点となるように相対座標化を行う.

関節 $\ell$ の時刻 $t$ における絶対座標を $a_\ell(t)$，中心の関節の時刻 $t$ における絶対座標を $a_c(t)$ とすると，相対座標 $r$ は

$$r_\ell(t) = a_\ell(t) - a_c(t)$$

と定める. 本研究において，身体の中心 $c$ は Kinect で Spine-Mid，OpenPose で Spine(脊椎) を用いた.

### A.3.4 DTW 距離の計算

各時系列データの類似度を DTW 距離を用いて定める. 本研究では，2.4 節の $cost$ 関数として 3 次元ベクトルのユークリッド距離

$$\|\mathbf{p}_i - \mathbf{q}_i\| = \sqrt{(p_{i,x} - q_{i,x})^2 + (p_{i,y} - q_{i,y})^2 + (p_{i,z} - q_{i,z})^2}$$

を用いる. 歩行 1 サイクルの関節 $\ell$ の 2 つの時系列データ $R_\ell = \langle r_\ell(t_1), ..., r_\ell(t_n)\rangle$ と $R'_\ell = \left\langle r'_\ell(t_1), ..., r'_\ell(t_{n'})\right\rangle$ の DTW 距離 $d(R, R')$ を $R$ と $R'$ の類似度とする. DTW 距離の性質から，$R = R'$ ならば $d(R, R') = 0$ であり，$n$ と $n'$ は一致する必要はない.

また，複数の関節を用いたときの類似度は次のように定める. 異なる関節 $m$ と関節 $\ell$ について 2 つの時系列データ $(R_\ell, R_m)$ と $(R'_\ell, R'_m)$ があるとき，統合 DTW 距離 $D((R_\ell, R_m), (R'_\ell, R'_m))$ は，$\ell$ と $m$ についての DTW 距離の L2 ノルム (ユークリッド距離)，すなわち，$\sqrt{d(R_\ell, R'_\ell)^2 + d(R_m, R'_m)^2}$ とする. 同様に，$k$ 種の関節を統合した場合も，$k$ 次元のユークリッド距離で類似度を定める.

### A.3.5 個人識別

#### 単独歩行

ある単独歩行のデータに対して，その他の単独歩行のデータ全てとの間で DTW 距離を計算し，最も DTW 距離が小さかった歩行データの該当者を識別結果とする．すなわち，サイクル切り出しと相対座標化を行った単独歩行のデータ $N$ 組のうち $i$ 番目の歩行データを $W_i$ と表したとき，次の問題の解 $j^*$ が識別結果である.

$$j^* = \underset{i \neq j \in \{1,...,N\}}{\arg\min}\ D(W_i, W_j)$$

#### 複数人歩行

単独歩行の識別と同様にして行う．ある複数人歩行データの識別結果は，全ての単独歩行データとの間で DTW 距離を計算し，最も DTW 距離が小さかった歩行データの該当者を表す id である.

## A.4 実験

### A.4.1 歩行実験

#### 実験目的

Kinect と OpenPose の多人数同時個人識別の精度を比較するため，Kinect を用いて歩行映像と各関節座標の時系列を得る実験を行う.

#### 実験方法

Kinect によって得た映像と関節座標を保存するためのシステムは Processing を用いて開発した．Processing で Kinect for Windows v2 を扱うためのライブラリとして KinectPV2[5] を利用した.

単独歩行と複数人歩行をそれぞれ観測した．複数人歩行は 2〜6 人が同時に歩行し，そのデータを得る実験である．実験参加者の詳細を表 A.3 に示す. 実験環境は図 A.7 の通りである.

単独歩行では Kinect に対して直進する方向とそこから $\pm$ 30 $^\circ$ 傾いた方向の 3 パターンについて，1 人当たり 2 回ずつ測定した. 複数人歩行では，2 ~ 6 人と人数を変えながら，全員が直進するパターンを 3 回，交差が発生するパターンを 3 回測定した.

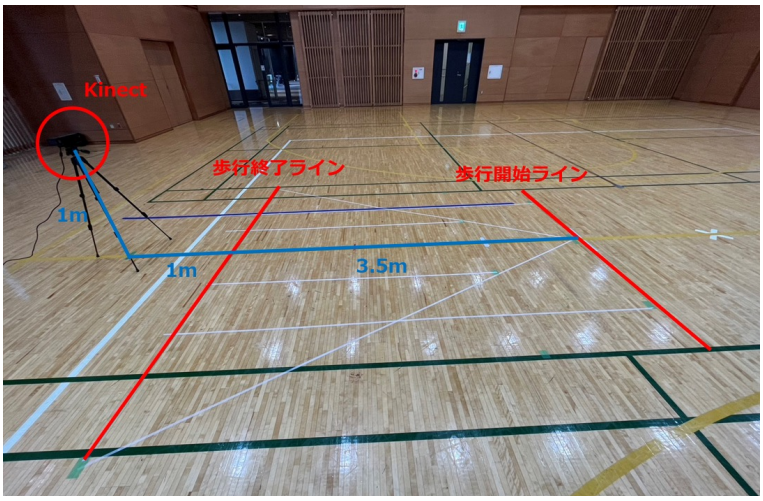



**図 A.7:** 実験環境

**表 A.3:** 実験参加者の情報

| 項目 | 環境 |
|---|---|
| 実験日 | 2022 年 7 月 16 日 |
| 実験時刻 | 9:30 から約 2 時間 |
| 場所 | 明治大学 中野キャンパス 多目的ホール |
| 年齢 | 20 代前半 |
| 性別 | 男性 4 名 女性 3 名 |
| 人数 | 7 名 |

### A.4.2 センシング誤り評価実験

**実験目的**

Kinect で推定した関節座標データに対して Kinect のセンシング誤りによるデータの誤差がどの程度存在しているのかを調べる．歩行の向きや人数の変化による Kinect の推定精度を評価する．

**実験方法**

データからランダムにフレームを選んで可視化し，手動でラベル付けを行う．データの選び方は単独歩行と複数人歩行で異なる．

単独歩行については，実験参加者 7 人の歩行データについて，それぞれ 3 種類の歩行方向から 3 フレームずつランダムに選ぶ．可視化するフレームの合計は $7 \times 3 \times 3 = 63$ フレーム分である．

複数人歩行については，各歩行人数のデータに対して 1 人あたり 2 フレーム，合計 $7 \times 2 = 14$ フレームをランダムに選んで可視化する．(ただし 2 人歩行については 1 人参加していない参加者が存在するため，複数人歩行の合計可視化フレーム数としては 68 フレームである．)

ラベル付けに関しては，正常に見えるもの，一部に異常が見られるもの，全体的に異常なものの 3 種類に分

類する．各ラベルのサンプルを次の図 A.8, A.9, A.10 に示す．

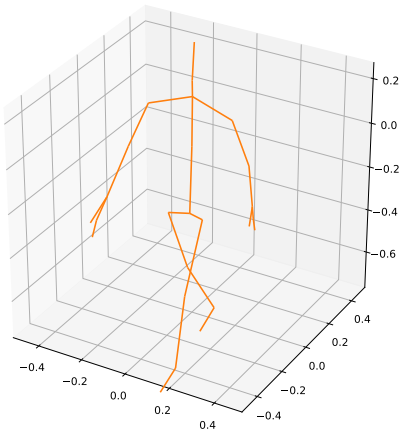



**図 A.8:** 正常なフレーム

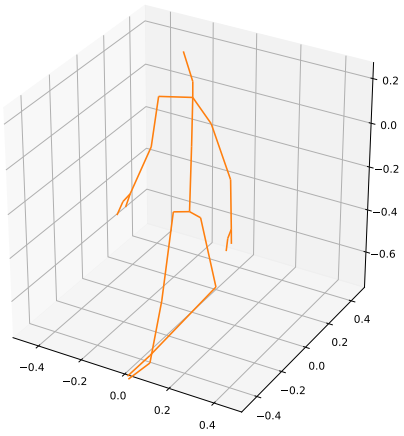



**図 A.9:** 一部に異常が見られるフレーム

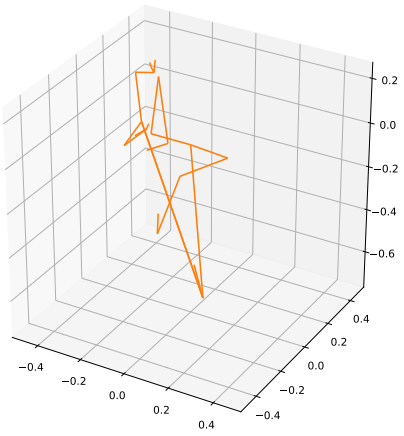



**図 A.10:** 全体的に異常なフレーム

**実験結果**

歩行方向を正面と正面以外に分けたときの各ラベルの占める割合を表 A.4 に示す．複数人歩行と単独歩行を合わせて歩行人数に関してラベルの割合を表 A.5 と図 A.11 に示す．

**表 A.4:** 各歩行方向に対するセンシング状態の割合

| 歩行方向 (Kinect 基準) | 正常 | 一部異常 | 全体異常 |
|---|---|---|---|
| 正面 | 1.0 (21 / 21) | 0.0 (0 / 21) | 0.0 (0 / 21) |
| 正面以外 | 0.48 (20 / 42) | 0.33 (14 / 42) | 0.19 (8 / 42) |
| 合計 | 0.65 (41 / 63) | 0.22 (14 / 63) | 0.13 (8 / 63) |

**表 A.5:** 歩行人数に対するセンシング状態の割合

| 歩行人数 | 正常 | 一部異常 | 全体異常 |
|---|---|---|---|
| 1 | 0.65 (41 / 63) | 0.22 (14 / 63) | 0.13 (8 / 63) |
| 2 | 0.67 (8 / 12) | 0.17 (2 / 12) | 0.17 (2 / 12) |
| 3 | 0.64 (9 / 14) | 0.29 (4 / 14) | 0.071 (1 / 14) |
| 4 | 0.36 (5 / 14) | 0.43 (6 / 14) | 0.21 (3 / 14) |
| 5 | 0.29 (4 / 14) | 0.50 (7 / 14) | 0.21 (3 / 14) |
| 6 | 0.43 (6 / 14) | 0.29 (4 / 14) | 0.29 (4 / 14) |

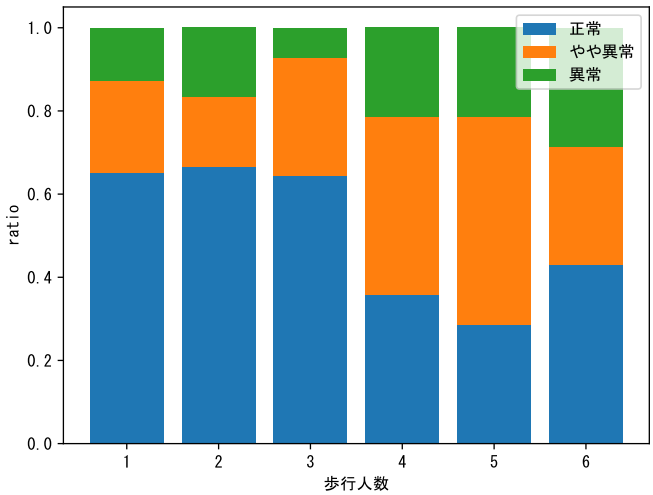



**図 A.11:** 歩行人数変化とセンシング誤りの割合

### A.4.3 個人識別実験

**実験目的**

単独歩行・複数人歩行に対する Kinect と OpenPose の精度を調べる.

**実験方法**

3.5 節の手法に基づき，各歩行データに対して個人識別を行う．また，$1 \leq k \leq 5$ の top と topk の推定を含めた精度 $top_kAcc$ を用いて，Kinect と OpenPose の間で精度を比較する．ただし，$top_kAcc$ の計算においては，DTW 距離が最小のものから昇順で該当個人 id を並べたものを識別結果としている．

**実験結果**

単独歩行における識別結果の混同行列を表 A.6, A.7 に示し，$top_kAcc$ ($k = 1, 2, 3, 4, 5$) を図 A.12 に示す. また，歩行人数を増加させたときの識別精度の推移を $top_kAcc$ ($k = 1, 3, 5$) について求め, その結果を図 A.13, A.14, A.15 に示す.

**表 A.6:** 混同行列 (OpenPose)

| 真値＼予測 | *A* | *B* | *C* | *D* | *E* | *F* | *G* | FRR(%) |
|---|---|---|---|---|---|---|---|---|
| *A* | 2 | 0 | 2 | 1 | 0 | 0 | 1 | 66.7 |
| *B* | 0 | 2 | 1 | 0 | 0 | 3 | 0 | 66.7 |
| *C* | 0 | 0 | 3 | 2 | 0 | 1 | 0 | 50.0 |
| *D* | 0 | 0 | 1 | 4 | 0 | 1 | 0 | 33.3 |
| *E* | 1 | 0 | 1 | 0 | 2 | 0 | 2 | 66.7 |
| *F* | 0 | 1 | 1 | 1 | 0 | 2 | 1 | 66.7 |
| *G* | 1 | 0 | 1 | 1 | 0 | 0 | 3 | 50.0 |
| FAR(%) | 5.6 | 2.8 | 19.4 | 13.9 | 0.0 | 13.9 | 8.3 | - |

**表 A.7:** 混同行列 (Kinect)

| 真値＼予測 | *A* | *B* | *C* | *D* | *E* | *F* | *G* | FRR(%) |
|---|---|---|---|---|---|---|---|---|
| *A* | 5 | 0 | 1 | 0 | 0 | 0 | 0 | 16.7 |
| *B* | 0 | 4 | 1 | 0 | 0 | 0 | 1 | 33.3 |
| *C* | 1 | 0 | 4 | 0 | 0 | 1 | 0 | 33.3 |
| *D* | 0 | 0 | 0 | 5 | 0 | 0 | 1 | 16.7 |
| *E* | 0 | 0 | 0 | 0 | 5 | 0 | 1 | 16.7 |
| *F* | 0 | 0 | 1 | 0 | 0 | 4 | 0 | 16.7 |
| *G* | 0 | 0 | 0 | 2 | 2 | 0 | 2 | 66.7 |
| FAR(%) | 2.8 | 0.0 | 8.3 | 5.6 | 5.6 | 2.8 | 8.3 | - |

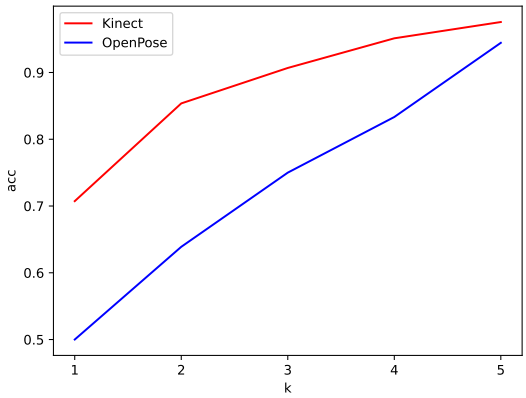



**図 A.12:** 単独歩行 top-k acc

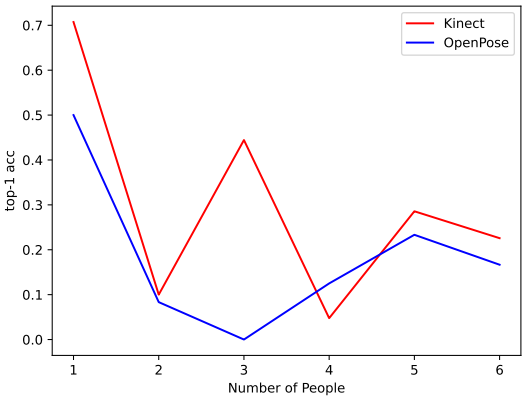



**図 A.13:** 人数変化に伴う精度推移 (top-1 acc)

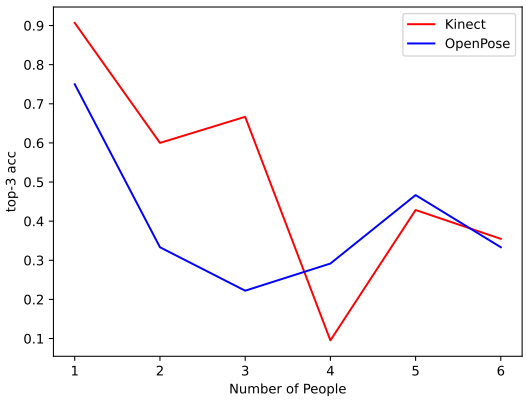



**図 A.14:** 人数変化に伴う精度推移 (top-3 acc)

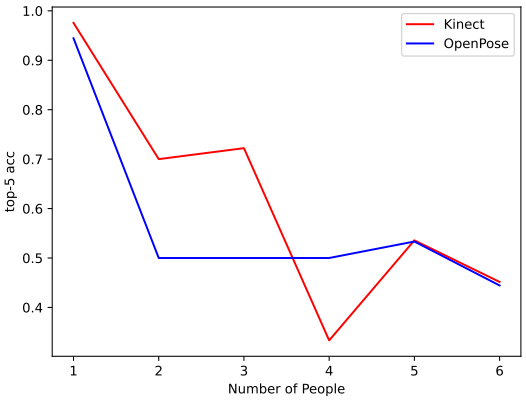



**図 A.15:** 人数変化に伴う精度推移 (top-5 acc)

### A.4.4 考察

#### Kinect のセンシング誤り

単独歩行について，Kinect に対する歩行角度でセンシング精度に違いが見られた．Kinect は姿勢推定を行うとき，各画素に対して 25 関節のうちどの関節に該当するかを分類し，関節ごとに分類された画素の中心座標を求めることで姿勢推定を行っている．そのため，Kinect に対して角度が付くことで，Kinect から見えづらい関節が増えて姿勢推定は失敗しやすくなると考えられる．また，歩行人数に対する Kinect のセンシング精度について，4, 5 人歩行のとき低下した．ここから，Kinect が問題なく性能を発揮できるのは 3 人同時姿勢推定までであると言える．また，6 人歩行のとき 1 3 人歩行ほどではないが 4, 5 人歩行より精度が上がった．これは，そもそも 6 人を同時に捉えられず，歩行人数が少ない状態に等しくなっているからと考えられる．実際に，6 回の 6 人歩行のうち 2 回はそもそも 6 人分の歩容が得られていなかった．

**個人識別**

歩行人数が 1 ~ 3 人のとき，$k$ の値に関わらず OpenPose よりも Kinect の方が識別精度 top-k acc. が高かった．しかし，4 人歩行で精度が逆転した．6 人歩行に対しては Kinect の方がやや有利であったがあまり精度が変わらなかった．このことは，Kinect のセンシング精度が 3 人までは性能が保たれ，4 人以降で精度が減少するという先述の考察と一貫している．従って，6 人歩行について 6 人同時に捉えられていないことを考慮すると，3 人以下の歩行は Kinect が有利であり，4 人以上の歩行は OpenPose がやや有利，7 人以上の歩行について Kinect の制約から OpenPose が有利である，と結論づける．

## A.5　結論

Kinect のセンシング精度は Kinect に対してまっすぐ歩行するとき高く，そうでないとき精度が低下する．また，3 人までは精度を落とさず姿勢推定が出来るが，4 人以上で精度が低下する．個人識別については，3 人までは Kinect が有利であり，4 人以上では OpenPose がやや有利であることを実験に基づき明らかにした．今後の課題として，歩行実験の参加者数を増やすことや，7 人以上の歩行に対する個人識別について調べることが挙げられる．

# 参考文献


[1] 森 駿文，菊池 浩明, “ 歩容データの DTW 距離に基づく個人識別手法の提案と外乱に対する評価 ”，情報処理学会論文誌，Vol.60, No.9, 1538-1549, 2019.

[2] Ju Han, Bir Bhanu, “ Individual recognition using gait energy image ” , IEEE transactions on pattern analysis and machine intelligence, 28(2)，316-322，2005.

[3] Zhe， Gines，Tomas，Shih-En，Yaser, “ OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields ”，CVPR，pp.7291-7299，2017.

[4] Julieta, Rayat, Javier, James, “ A simple yet effective baseline for 3d human pose estimation ” , ICCV， pp. 2640-2649, 2017.

[5] Thomas Sanchez Lengeling,“ Kinect v2 library for Processing ”(`https://github.com/ThomasLengeling/KinectPV2`)，2016.

[6] 渡邊宏, “「Kinect v2」はここがスゴい！新旧比較と Kinect による NUI 開発の最前線 ”，MONOist，2014.

[7] Sakoe, H. and Chiba, S, “ Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transaction on Acoustics, Speech, and Signal Processing ” ,Vol.ASSP-26, No.1, pp.43–49，1978.

[8] 三好駿, 森駿文, 菊池浩明, “ 歩容データからの属性暴露リスクについて ”，情報処理学会第 81 回全国大会, pp.3_421-3_422, 2019.

[9] 阪田 篤哉, 武村 紀子, 八木 康史, “ 多段階畳み込みニューラルネットワークを用いた歩容に基づく年齢推定 ”， 2018 年 5 月コンピュータビジョンとイメージメディア研究会，吹田，Vol. 2018-CVIM-212，No. 23，pp. 1-5，May 2018.