SCIS 2025 2025 Symposium on
Cryptography and Information Security
Kokura, Japan, Jan. 28 – 31, 2025
The Institute of Electronics,
Information and Communication Engineers

準同型暗号 CKKS 方式を用いたガウス過程回帰の構成
Gaussian Process Regression using by homomorphic encryption CKKS scheme

河原 直翔 * 若杉 飛鳥 † 服部 大地 † 小寺 雄太 *
Naoto Kawahara Asuka Wakasugi Daichi Hattori Yuta Kodera
野上 保之 *
Yasuyuki Nogami

あらまし 準同型暗号とは，暗号化されたデータに対して計算を行うことができる技術である．これは 1970 年代から研究されているテーマであり，2009 年に Gentry によって完全準同型暗号の構成が提案され，以降 1 世代から 4 世代まで分類される．そして，実数を扱うことができる CKKS 方式が 4 世代の方式である．準同型暗号を用いることで，個人情報のような機密性の高いデータに対して，機械学習アルゴリズムを適用することができる．いくつかのアルゴリズムは CKKS 方式で実装されているが，ガウス過程回帰アルゴリズムに関する研究は知られていない． 本稿では，そのアルゴリズムの構成及び実装を与える．また，広く利用されているデータセットを対象に， CKKS 方式を用いたガウス過程回帰の訓練及び，予測アルゴリズムに対する実行時間を計測する．そして，暗号文での訓練及び，予測後の出力値と，cleartext で scikit-learn モジュールを用いたそれらの出力値で誤差を算出する．その結果，暗号文でのガウス過程回帰は，訓練及び，予測後の出力値ともに，データセットのサイズに対して，2 乗に比例する実行時間で算出でき，cleartext のガウス過程回帰とほぼ同じ出力値を得ることができることが示された．

キーワード ガウス過程回帰，準同型暗号，CKKS，秘密計算，プライバシー保護

1 はじめに

現代の情報化社会では，大量のデータを利用でき，それらのデータの活用法として，機械学習が注目されている．機械学習により，与えられたデータからパターンを抽出し，未知のデータに対する予測をすることができる．機械学習を用いるサービスを展開する産業や企業は多く，その活用事例はさらに広がっていくことが予想される．その中でも，ガウス過程回帰 [11] は特徴として，予測した際に，結果を平均値と分散として出力することができ，予測の不確実性を示すことが可能である．

一方，機械学習に使用されるデータには制約が伴う．例えば，法律規制による個人に関するデータや企業の競争優位性を守るための機密情報など，一部のデータは非公開にしたいという制約が存在する．その場合，クラウドサービスや複数企業のデータを用いる機械学習サービスを活用することが難しい．このような困難を解決すると，機密性の高いデータも機械学習に用いることができる．以上の背景から，データのプライバシーを保護しながら，演算を行うための技術である秘密計算の重要性が増している．

秘密計算技術の一つである準同型暗号とは，暗号化されたデータに対して計算を行うことができる技術であり，1978 年に Rivest, Adleman, Dertouzos [12] によって提案された．その後 Gentry [10] によって，2009 年に暗号文で加算と乗算を実現できる完全準同型暗号が提案された．本稿では，準同型暗号によるガウス過程回帰アルゴリズムを構成する．

1.1 先行研究

準同型暗号に関するこれまでの研究は，1 世代から 4 世代に分類される．順に，イデアル格子暗号方式 [10]，BGV・BFV 方式 [2, 8]，TFHE 方式 [5]，CKKS 方式 [4] とされる．BFV 方式や CKKS 方式を使用した機械学習アルゴリズムは数多く知られている．準同型暗号を用いた機械学習アルゴリズムとして，線形回帰 [1]，最小二乗

* 岡山大学大学院,700-0082 岡山県岡山市北区津島中 3-1,Okayama University, 3-1 Tsushima-naka Kita-ku Okayama City Okayama 700-0082 Japan
† EAGLYS 株式会社, 東京都渋谷区千駄ヶ谷 5 丁目 27-3 やまとビル 7F,Eaglys Inc., 7F Yamato Building 5-27-3 Sendagaya Shibuya-ku Tokyo Japan

法 [7]，リッジ回帰 [6] などが提案されている．BFV 方式を使用したガウス過程回帰アルゴリズム [9] は提案されているが，CKKS 方式を使用したアルゴリズムについては，著者らが調べた限り知られていない．ガウス過程回帰アルゴリズムでは，与えられた行列の逆行列を求める必要がある．CKKS 方式の暗号文による逆行列を算出するアルゴリズムとして [13] が提案されている．

1.2 構成

本稿は次のように構成される．まず 1 章ではガウス過程回帰の概要やデータの保護の必要性，準同型暗号について説明した．2 章ではガウス過程回帰アルゴリズムと準同型暗号 CKKS 方式について概説する．続く 3 章では，準同型暗号 CKKS 方式を用いたガウス過程回帰の提案アルゴリズムを構成する．4 章にて，提案アルゴリズムの実行時間や誤差評価について議論する．最後の 5 章で結論を述べる．

1.3 貢献

本稿では，準同型暗号 CKKS 方式を用いたガウス過程回帰のアルゴリズム及びプロトコルの提案を行い，アルゴリズムの実装および実験を行った．実験の結果，2 つの異なるデータセットに対して，暗号文でのガウス過程回帰での，実行時間がデータ数に対して 2 乗に比例することが確認をした．また，scikit-learn を使った cleartext でのガウス過程回帰の出力結果と比較して，絶対誤差が小さいことを確認した．

2 準備

2.1 CKKS 方式

本節では，CKKS 方式の概要を説明する．分析対象データを cleartext という．また，暗号化対象データを plaintext（平文）と呼び，cleartext から plaintext への変換アルゴリズムを encode，その逆変換アルゴリズムを decode という．CKKS 方式で用いられる encode 方式には，Coefficients（以下 Coeff）方式と Slot 形式の 2 種類が知られている．Coeff 方式とは，[4] らによる encode 方式である．Slot 方式とは，cleartext に対して，フーリエ変換アルゴリズムを適応させてから，Coeff 方式を用いる encode 方式である．本稿では，Slot 形式の encode/decode アルゴリズムを $\mathrm{Encode}_{\mathrm{Slot}}, \mathrm{Decode}_{\mathrm{Slot}}$ とし，その平文を p_{Slot} と表す．平文空間は，整数係数多項式環の円分多項式による剰余環によって定められ，本稿では，その多項式の長さを N とする．

cleartext が 1 次元配列のデータ型であるとき，Coeff 方式及び Slot 方式いずれの場合でも，平文は長さ N の多項式で表現される．本稿では，cleartext の配列の長さを n で表し，$n \leq N$ とする．秘密鍵を sk, 公開鍵を pk とし，$\mathrm{Encrypt}_{\mathrm{pk}}(p_{\mathrm{Slot}})$ で定まる暗号文を c_{Slot} とする．c_{Slot} は長さ N の整数係数多項式である．このとき，sk による復号関数を用いると，$\mathrm{Decrypt}_{\mathrm{sk}}(c_{\mathrm{Slot}}) = p_{\mathrm{Slot}}$ が成り立つ．c_{Slot} に対して，次の 4 つのアルゴリズムの組 (Add, Sub, Mul, Inner_Product_Slot) が定義される．以下では，c_{Slot} が定まる暗号文空間内での演算を考える．Add/Sub/Mul では，それぞれ，Slot 形式による暗号文を多項式加算/減算/乗算する．Inner_Product_Slot とは，Slot 形式による暗号文同士に対して，多項式を配列とみなした際の内積を算出する．以降ではこの演算を IPS と略記する．詳細は [13] を参照されたい．

さらに，cleartext が $m \times n$ 行列の場合でも，暗号化アルゴリズム及び上記の演算を同様に定められる．こちらも詳細は [13] を参照されたい．

2.2 ガウス過程回帰

本節ではガウス過程回帰の基礎的な事項を説明する．

2.2.1 ガウス過程回帰

事前分布

ガウス過程回帰は，モデルを訓練し予測した際に，出力を数値ではなく正規分布で返す機械学習アルゴリズムである．x は無限次元の標準正規分布に従い，$f(x)$ は正規分布に従うと仮定する．n 個の訓練データ点 $X = [x_1, x_2, \ldots, x_n]$ に対して，関数値 $f(X) = [f(x_1), f(x_2), \ldots, f(x_n)]$ は次の正規分布に従う．

$$f(X) \sim \mathcal{N}(0, K(X, X))$$

ここで，n' 個の新たな入力データ点 $X^* = [x_1^*, x_2^*, \ldots, x_{n'}^*]$ を与え，0 はゼロベクトルとすると，$K(X^*, X)$ はカーネル行列とよばれ，次のように定義される:

$$K(X^*, X) = \begin{pmatrix} k(x_1^*, x_1) & \ldots & k(x_1^*, x_n) \\ \vdots & \ddots & \vdots \\ k(x_{n'}^*, x_1) & \ldots & k(x_{n'}^*, x_n) \end{pmatrix}$$

ここで，$1 \leq i, j \leq n$ に対して，$k(x_i^*, x_j)$ を次のように定める:

$$k(x_i^*, x_j) = \begin{cases} \sigma_a^{*2} \exp\left(-\frac{||x_i^* - x_j||^2}{2\sigma_b^2}\right) & \text{if } x_i \neq x_j, \\ \sigma_a^{*2} \exp\left(-\frac{||x_i^* - x_j||^2}{2\sigma_b^2}\right) + \sigma_c^{*2} & \text{if } x_i = x_j. \end{cases} \quad (1)$$

特に，行列 $K(X, X)$ は共分散行列と呼ばれ，データ間の相関関係を表現する．共分散行列は対称であり，正定値である必要がある．式 1 を用いた $K(X, X)$ はこれを満たす．また，$\sigma_a, \sigma_b, \sigma_c$ は，ハイパーパラメータと呼ばれ，これはガウス過程回帰を実行する以前に定められている定数値であり，通常は訓練データによって調整される．

Algorithm 1 compute_kernel_matrix(CKM)

Input: $V_1 = [v_{1,1}, v_{1,2}, \ldots, v_{1,\ell_1}] \in \mathbb{R}^{\ell \times \ell_1}$, $V_2 = [v_{2,1}, v_{2,2}, \ldots, v_{2,\ell_2}] \in \mathbb{R}^{\ell \times \ell_2}$

Output: $K(V_1, V_2)$

1: **for** $i := 1$ to ℓ_1 **do**
2: **for** $j := 1$ to ℓ_2 **do**
3: $K(V_1, V_2)_{ij} \leftarrow k(v_{1,i}, v_{2,j})$
4: **return** $K(V_1, V_2)$

Algorithm 2 training_in_plaintext

Input: $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times d}$, $y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^n$

Output: $K_{\text{inv}}, \text{yy}$

1: $K \leftarrow \text{CKM}(X^T, X^T)$
2: $K_{\text{inv}} \leftarrow \text{CIM}(K)$
3: $\text{yy} \leftarrow K_{\text{inv}} \cdot y$
4: **return** K_{inv}, yy

観測モデル

訓練データ $X = [x_1, x_2, \ldots, x_n]$ に対応する訓練データ $y = [y_1, y_2, \ldots, y_n]$ を与える．そして，y は，ある関数値 $f(X)$ にノイズ $\boldsymbol{\epsilon}$ が加わったものであると考える．ここで，$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I)$ は独立したガウスノイズと仮定され，$\sigma^2 I$ はノイズの分散を表す．また，σ はハイパーパラメータの 1 つである．

事後分布

ガウス過程回帰では，$X^* = [x_1^*, \ldots, x_{n'}^*]$ に対応する $y^* = [y_1^*, y_2^*, \ldots, y_{n'}^*]$ の予測を行うために，事後分布を計算する．事後分布は以下のように表される．

$$\begin{bmatrix} y \\ y^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K(X,X) + \sigma^2 I & K(X^*,X)^T \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix} \right)$$

事後分布から，X^* に対応する．y^* を正規分布として，平均と標準偏差を得ることができる．diagonal は行列の対角成分をベクトル形式で取り出す操作を表し，sqrt は平方根を取る操作を表す．平均 f^*，分散 v^*，標準偏差 σ^* は，それぞれ次のように計算される:

$$f^* = K(X^*, X)\, Sy$$
$$v^* = \text{diagonal}(K(X^*, X^*) - K(X^*, X)\, SK(X^*, X)^T)$$
$$\sigma^* = \text{sqrt}(v^*)$$

2.2.2 ガウス過程回帰のアルゴリズム

本節では，ガウス過程回帰における訓練フェーズと予測フェーズにおけるアルゴリズムを紹介する．まず，サブルーチンとして，与えられた 2 つの行列から，カーネル行列を求めるアルゴリズムを **アルゴリズム 1** で与える．ここで，3 行目の $K(V_1, V_2)_{ij}$ とは，$1 \le i \le \ell_1, 1 \le j \le \ell_2$ に対して，$K(V_1, V_2)$ の (i, j) 成分を表す．

Algorithm 3 predict_in_plaintext

Input: $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{n \times d}$, $X^* = [x_1^*, x_2^*, \ldots, x_{n'}^*] \in \mathbb{R}^{n' \times d}, K_{\text{inv}}$, yy

Output: f^*, σ^*

1: $K^* \leftarrow \text{CKM}(X^{*T}, X^T)$
2: $K^{**} \leftarrow \text{CKM}(X^{*T}, X^{*T})$
3: $f^* \leftarrow K^* \cdot \text{yy}$
4: $\sigma^* \leftarrow \text{diagonal}(K^{**} - K^* \cdot K_{\text{inv}} \cdot K^{*T})$
5: **return** f^*, σ^*

訓練フェーズにおけるアルゴリズムは **アルゴリズム 2** で示される．ここで，2 行目の CIM とは，compute_inver-se_matrix のことであり，与えられた正方行列の逆行列を求めるサブルーチンを表す．訓練フェーズでは，訓練データ $X = [x_1, x_2, \ldots, x_n]$ に対して転置行列 X^T を求め，X^T, X^T に対して，CKM を用いて，共分散行列を求める．その後，CIM を用いて，K_{inv} にその逆行列を，yy に K_{inv} と，$y = [y_1, y_2, \ldots, y_n]$ の積を代入する．

最後に，予測フェーズのアルゴリズムは，**アルゴリズム 3** で与えられる．ここで，6 行目の diagonal は行列の対角成分をベクトル形式で取り出すサブルーチンを表す．訓練データ $X = [x_1, x_2, \ldots, x_n]$ と新しい入力データ $X^* = [x_1^*, x_2^*, \ldots, x_{n'}^*]$ に対して転置行列 X^T, X^{*T} を求めて，カーネル行列 K^* と 共分散行列 K^{**} を求める．それらと訓練フェーズで計算した K_{inv}, yy との行列計算によって平均と標準偏差を算出する．

2.2.3 ガウス過程回帰の漸近計算量

訓練フェーズ

アルゴリズム 2 の 1 行目で $n \times d$ 行列に対して，カーネル行列を計算している．サブルーチンの，カーネル関数の計算量は $O(d)$ ゆえ，共分散行列を求めるのに $O(dn^2)$ だけ必要である．2 行目では，共分散行列の逆行列を算出する．逆行列計算の計算量オーダーは，一般に $O(n^3)$ である．3 行目の yy の計算では， $n \times n$ 行列と n ベクトルの積であるため，$O(n^2)$ となる．以上のことからガウス過程回帰アルゴリズムの訓練フェーズの計算量オーダーは $O(n^3), O(dn^2)$ となる．

予測フェーズ

アルゴリズム 3 の 1，2 行目で共分散行列を計算しており，訓練データ $n \times d$ 行列，テストデータ $n' \times n$ 行列に対して 1 行目では $O(dnn')$，2 行目では $O(dn'^2)$ となる．3 行目の平均の計算では，$n' \times n$ 行列と n ベクトル

Algorithm 4 training_in_ciphertext

Input: C_X, C_y, n

Output: $C_{K_{\text{inv}}}, C_{\text{yy}}$

1: $X \leftarrow \text{Decrypt}_{\text{sk}}(C_X)$
2: $K \leftarrow \text{CKM}(X^T, X^T)$
3: $C_K \leftarrow \text{Encrypt}_{\text{pk}}(K)$
4: $C_{K_{\text{inv}}} \leftarrow \text{ECIM}(C_K)$
5: $c_{\text{yy}} \leftarrow \text{IPS}(C_{K_{\text{inv}}}, c_y, n)$
6: **return** $C_{K_{\text{inv}}}, C_{\text{yy}}$

の積であるため $O(nn')$ である．5 行目の分散の計算では，$n' \times n$ 行列と $n \times n$ の行列積が計算量をしめており $O(n'n^2)$ となる．以上のことからガウス過程回帰アルゴリズムの予測フェーズは $O(dnn'), O(dn'^2), O(n'n^2)$ となる．

3 実装対象アルゴリズム

本章では CKKS 方式によってプライバシー保護されたガウス過程回帰のプロトコルについて説明する．実際の機械学習サービスを想定して， Alice と Bob の 2 者間で通信を行うと仮定する． Alice が秘密鍵をもっているクライアント側，Bob がサーバ側として暗号文状態の計算を実行する．

3.1 訓練フェーズのプロトコル

実装した CKKS 方式を用いたガウス過程回帰の訓練フェーズのアルゴリズムを 3.1.1 小節で，想定されるプロトコルについて 3.1.2 小節で説明する．

3.1.1 具体的なアルゴリズム

本稿で実装した CKKS 方式を用いるガウス過程回帰の訓練フェーズについて **アルゴリズム 4** に示す． 2.1 節の手法を用いて，暗号化された訓練データ C_X, C_y に対して，暗号化された $C_{K_{\text{inv}}}, C_{\text{yy}}$ を返す．ここで，4 行目の ECIM とは，暗号化された行列に対して，逆行列を算出するサブルーチンを表し，暗号文に対する行列演算の中では最大の計算量であることが知られている．ここでは [13] を用いる．またこれらの計算量は，正整数 r を用いて，$O(rm^2N^2)$ となり，1, 2 行目の CKM の計算量が $O(dn^2)$ であるため，このアルゴリズムの計算量は $O(rm^2N^2)$ となる．

3.1.2 想定されるプロトコル

実装した訓練フェーズのアルゴリズムは次のような Alice と Bob による通信プロトコルが想定される． 暗号文の組 C_X, C_y を Bob が所有し，これらの暗号文に対応する秘密鍵 sk を Alice が所有しているものとする．

Algorithm 5 predict_in_ciphertext

Input: $C_X, C_{X^*}, C_{K_{\text{inv}}}, c_{\text{yy}}, n$

Output: C_{f^*}, C_{σ^*}

1: $X \leftarrow \text{Decrypt}_{\text{sk}}(C_X)$
2: $X^* \leftarrow \text{Decrypt}_{\text{sk}}(C_{X^*})$
3: $K^* \leftarrow \text{CKM}(X^{*T}, X^T)$
4: $K^{**} \leftarrow \text{CKM}(X^{*T}, X^{*T})$
5: $C_{K^{**}} \leftarrow \text{Encrypt}_{\text{pk}}(K^{**})$
6: $C_{K^*} \leftarrow \text{Encrypt}_{\text{pk}}(K^*)$
7: $C_{K^{*\mathsf{T}}} \leftarrow \text{Encrypt}_{\text{pk}}(K^{*\mathsf{T}})$
8: $C_a \leftarrow \text{IPS}(C_{K^*}, C_{K_{\text{inv}}}, n)$
9: $C_b \leftarrow \text{IPS}(C_a, C_{K^{*\mathsf{T}}}, n)$
10: $C_{f^*} \leftarrow \text{IPS}(C_{K^*}, C_{\text{yy}}, n)$
11: $C_{V^*} \leftarrow \text{Sub}(C_{K^{**}}, C_b)$
12: $V^* \leftarrow \text{Decrypt}_{\text{sk}}(C_{V^*})$
13: $\sigma^* \leftarrow \text{sqrt}(\text{diagonal}(V^*))$
14: $C_{\sigma^*} \leftarrow \text{Encrypt}_{\text{pk}}(\sigma^*)$
15: **return** C_{f^*}, C_{σ^*}

このとき，想定されるプロトコルは次の 7 ステップからなる:

1. Bob は C_X を Alice へ送信する．
2. Alice は C_X に対して $\text{Decrypt}_{\text{sk}}$ を行い，cleartext X を得る．
3. Alice は X^T を用いて共分散行列 K を計算する．
4. Alice は K に対して，$\text{Encrypt}_{\text{pk}}$ を行い暗号文 C_K を得る．
5. Alice は C_K を Bob へ送信する．
6. Bob は C_K に対して ECIM を実行し逆行列の暗号文 $C_{K_{\text{inv}}}$ を得る．
7. Bob は $C_{K_{\text{inv}}}, C_y, n$ に対して，IPS を実行し，暗号文 C_{yy} を得る．

3.2 予測フェーズのプロトコル

3.2.1 小節で実装した CKKS 方式を用いたガウス過程回帰の予測フェーズのアルゴリズムを，3.2.2 小節では，想定されるプロトコルについて説明する．

3.2.1 具体的なアルゴリズム

予測フェーズについて **アルゴリズム 5** に示す．暗号化されたデータ C_X, C_{X^*} 及び訓練フェーズで得られた $C_{K_{\text{inv}}}, C_{\text{yy}}$ に対して，それらの分布の平均と標準偏差を暗号化した C_{f^*}, C_{σ^*} を返す．またこれの計算量は，3.1.1 小節と同様に考えると，$O(m^2nN^2)$ となる．

3.2.2 想定されるプロトコル

実装した予測フェーズのアルゴリズムは次のような Alice と Bob による通信プロトコルが想定される．暗号文の組 $C_X, C_{X^*}, C_{K_{\mathrm{inv}}}, C_{\mathrm{yy}}$ を Bob が所有し，これらの暗号文に対応する秘密鍵 sk を Alice が所有しているものとする．このとき，想定されるプロトコルは次の 15 ステップからなる:

1. Bob は C_X, C_{X^*} を Alice へ送信する．
2. Alice は C_X, C_{X^*} に対して $\mathrm{Decrypt}_{sk}$ を行い，cleartext X, X^* を得る．
3. Alice は X^{*T}, X^T を用いて共分散行列 K^* を得る．
4. Alice は X^* を用いて共分散行列 K^{**} を得る．
5. Alice は $K^{**}, K^*, K^{*\mathsf{T}}$ に対して $\mathrm{Encrypt}_{\mathrm{pk}}$ を行い，$C_{K^{**}}, C_{K^*}, C_{K^{*\mathsf{T}}}$ を得る．
6. Alice は $C_{K^{**}}, C_{K^*}, C_{K^{*\mathsf{T}}}$ を Bob へ送信する．
7. Bob は $C_{K^*}, C_{K_{\mathrm{inv}}}, n$ に対して IPS を実行し，暗号文 C_a を得る．
8. Bob は $C_a, C_{K^{*\mathsf{T}}}, n$ に対して IPS を実行し暗号文 C_b を得る．
9. Bob は暗号文 $C_{K^*}, C_{\mathrm{yy}}, n$ に対して，IPS を実行し平均の暗号文 C_{f^*} を得る．
10. Bob は暗号文 $C_{K^{**}}, C_b$ に対して，Sub を実行し分散の暗号文 C_{V^*} を得る．
11. Bob 暗号文 C_{V^*} を Alice へ送信する．
12. Alice は C_{V^*} に対して $\mathrm{Decrypt}_{\mathrm{sk}}$ を行い，cleartext V^* を得る．
13. Alice は V^* の対角成分を取り出し平方根計算を行い，標準偏差 σ^* のベクトルを得る．
14. Alice は σ^* に対して，$\mathrm{Encrypt}_{pk}$ を行い，標準偏差の暗号文 C_{σ^*} を得る．
15. Alice は C_{σ^*} を Bob へ送信する．

4 実験

本章では，いくつかのデータセットに対して，**アルゴリズム** 4 及び 5 を実際に実装し，その実行時間などを考察する．

表 1: 実験環境

OS	Ubuntu 22.04.3 LTS
CPU	Intel(R) Xeon(R) Silver 4314
クロック周波数	2.40 GHz
RAM	64 GB
SEAL	4.0.0
Python	3.12.3
NumPy	1.26.4
scikit-learn	1.5.1

4.1 実験要件

CKKS 方式を用いた暗号文でのガウス過程回帰の精度を比較する対象となる，cleartext でのガウス過程回帰に scikit-learn [3] を用いる．具体的には，GaussianProcessRegressor クラスを使用し，訓練フェーズには fit メソッドを，予測フェーズでは predict メソッドを使用する．クラスをインスタンス化する際の引数について以下で説明する． kernel に ConstantKernel * RBF + WhiteKernel を指定した．これは 2.2.1 小節での，事前分布で示した，$k(x_i, x_j)$ に対応する．さらに，random_state に 1, n_restarts_optimizer に 10, optimizer に fmin_l_bfgs_b を指定した．これは fit メソッドでハイパーパラメータの最適化を行うためであり，これらは L-BFGS-B によって最適化される．また，alpha は 0 を指定した．本パラメータは L-BFGS-B にて最適化されないため，kernel にノイズ成分を補完させた．これは 2.2.1 小節での，観測モデルで示した，ノイズの分散 σ^2 に対応する．残りの変数の値は，GaussianProcessRegressor クラスにて用いられている値を使用した．

4.1.1 実験環境

本研究の実験環境を表 1 に示す．

4.1.2 データセット

実験に使用するデータセットには scikit-learn の load_-diabetes(以降 ld) と fetch_california_housing(以降 fch) を使用する．これらのデータセットは説明変数と目的変数にわけられており，それぞれを訓練データとテストデータに分割するために，scikit-learn の train_test_split 関数を使用した．引数の random_state には 1 を代入し，train_size にはデータ数に 0.8 をかけたものを用いる．これを用いて訓練データ，テストデータを分割する．また，データセットは cleartext 状態で事前に前処理を行う，説明変数に対して，正規化を行い，目的変数に対して，標準化を行う．正規化・標準化に対しては，それぞれ，scikit-learn の MinMaxScaler クラス，StandardScaler クラスを用いる．説明変数，目的変数ともに各クラスの，

訓練データに対して，fit メソッドでインスタンスへ，前処理に必要な情報を上書きする．その後，訓練データ，テストデータそれぞれを transform メソッドによって前処理する．以降では，前処理を終えた，訓練データの説明変数を X, 目的変数を y, テストデータの説明変数を X^*, 目的変数を y^* とする．

4.2 実験方法

本節では，本実験で用いる，暗号文でのガウス過程回帰についての前提条件並びに，本実験の実験手順を示す．

4.2.1 前提条件

暗号文でのガウス過程回帰を行う際，カーネル関数のハイパーパラメータには，cleartext でのガウス過程回帰の実験で得たハイパーパラメータを利用する．ハイパーパラメータの最適化は，暗号文でも可能であるが，複数回の訓練フェーズを実行する必要があり膨大な時間がかかるため，実験時間の短縮を図った．また，逆行列計算のアルゴリズム [13] の反復回数を決める iteration パラメータを 40 とする．

4.2.2 実験手順

まず，本検証を行う上で，データセットの妥当性を検証するために，各データセットに対し，cleartext でのガウス過程回帰を行い，その精度を R2 スコアで算出する．具体的には，データ数を 10 から 200 までの 10 刻みで増やし，それぞれに対して，4.1 節の手法で，X, y, X^*, y^* を作成する．X, y を用いて訓練フェーズを行い $K_{\mathrm{inv}}, \mathrm{yy}$ を得て，X^* を用いて，予測フェーズを行い f^*, σ^* を得る．f^* と y^* から R2 を測定する．

その後, 暗号文でのガウス過程回帰の実行時間を計測するために上記と同様の入力値を暗号化して, C_X, C_y, C_{X^*}, C_{y^*} を得て，それらに対して，提案手法の訓練フェーズ・予測フェーズを実行し，それぞれの実行時間を計測した．なお測定対象には訓練フェーズ，予測フェーズのみを対象とし，それらに必要な暗号化・復号処理は除外した．また，実行時間測定は訓練フェーズ・予測フェーズともに行列のサイズのみに依存すると考えられるため，データセットは ld のみを使用する．

さらに, 暗号文でのガウス過程回帰から得られた各出力値を復号したものと，cleartext でのガウス過程回帰の各出力値の誤差測定を行う．具体的には，$K_{\mathrm{inv}}, \mathrm{yy}, f^*, \sigma^*$ の誤差の最大値と最小値を測定した．また，それぞれの出力値は 1 よりも小さいため，絶対誤差のみを測定する．

4.3 実験結果

本節では，4.2.2 小節で示した実験手順に従い，実験した際の，実験結果を記載する．

Data Size	ld R2	fch R2
10	-1.20	-0.327
20	0.611	-0.0166
30	0.0395	-0.515
40	0.870	0.134
50	0.840	0.377
60	0.791	0.248
70	0.852	0.377
80	0.705	0.341
90	0.782	0.221
100	-1.87	0.253
110	0.288	0.247
120	-0.278	0.169
130	0.911	0.408
140	0.0752	0.164
150	0.224	0.201
160	0.430	0.575
170	0.476	0.256
180	0.748	0.182
190	0.658	0.328
200	0.280	0.456

表 2: 2 つのデータセットに cleartext でのガウス過程回帰を適応したときの R2 スコア

4.3.1 R2 スコア

測定した R2 を**表 2** に示す．データセット ld に対して，R2 の最小値と最大値は，それぞれ-1.87, 0.911 であり，データセット fch に対しては，-0.327, 0.575 となった．R2 の値は 1 未満であり，値が大きいほど精度が高いので，本データセットは，cleartext でのガウス過程回帰で適切に予測され得る．そのため，本データセットは暗号文のガウス過程回帰の評価に妥当であると判断した．

4.3.2 実行時間

暗号文のガウス過程回帰を用いた際の実行時間を示す．訓練フェーズでの X のサイズと実行時間の関係を**図 1** に示し，予測フェーズでの X^* のサイズと実行時間の関係を**図 2** に示した．また，**表 3** に抜粋した実行時間を示す．訓練フェーズ・予測フェーズともに，**図 1**，**図 2**，**表 3** より，実行時間は入力値のデータサイズの 2 乗に比例していることが実験的に示された．

4.3.3 データ数に対する誤差

データセット ld を用いた実験結果を**図 3** に，データセット fch を用いた実験結果を**図 4** に示す．**図 3**，**図 4** から $K_{\mathrm{inv}}, \mathrm{yy}, f^*, \sigma^*$ の絶対誤差の最大値は 10^{-4} 付近であることが確認できた．

Training Data Size	Execution Time [sec]	Test Data Size	Execution Time [sec]
40	407	10	2.59
80	1487	20	7.65
120	3240	30	15.3
160	5645	40	25.5

表 3: データ数に対する fit 関数と predict 関数の実行時間の抜粋

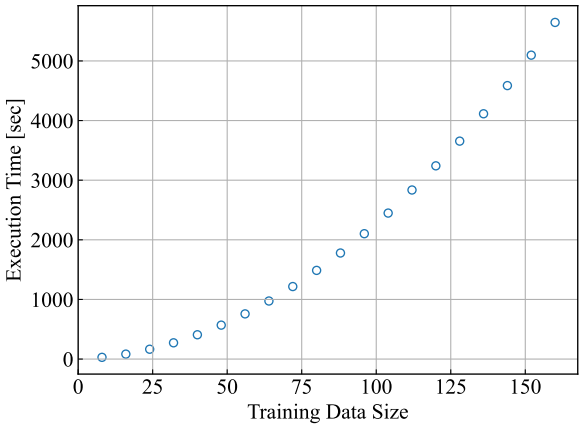

図 1: fit 関数の実行時間

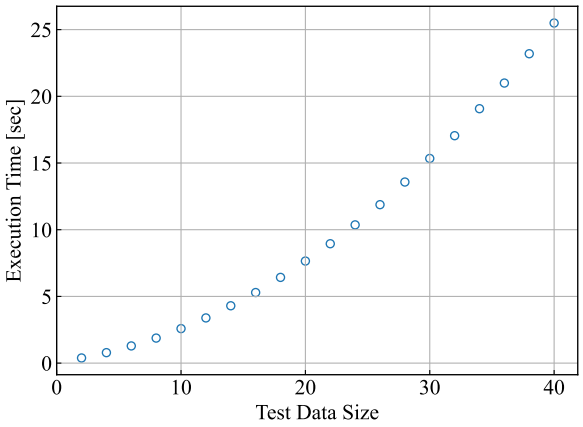

図 2: predict 関数の実行時間

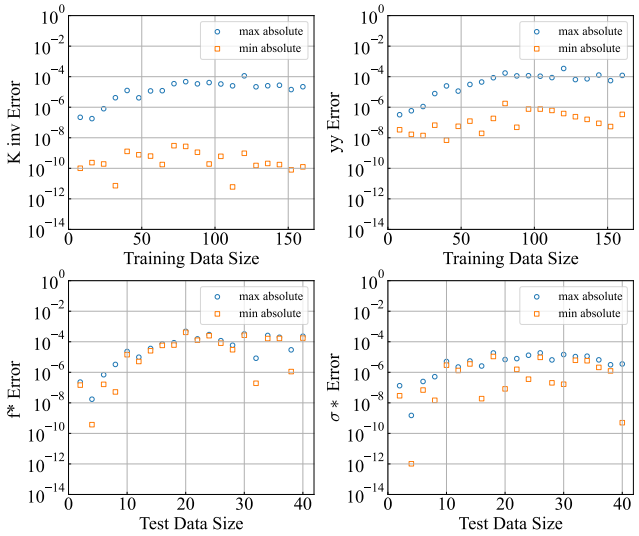

図 3: データセットが ld での計算値の誤差

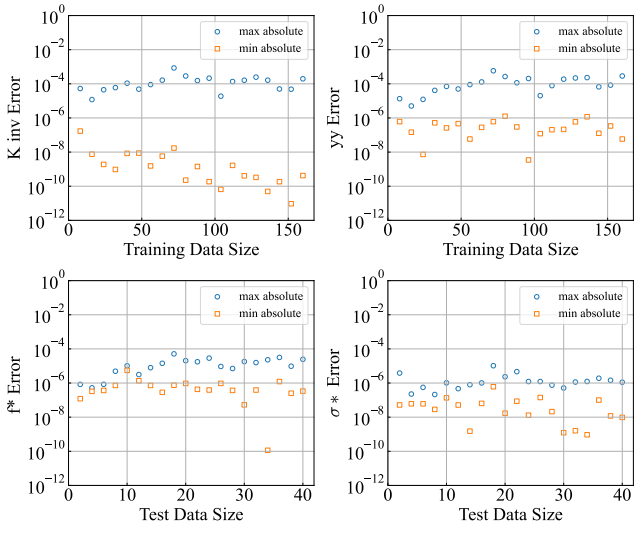

図 4: データセットが fch での計算値の誤差

5 まとめ

本稿では，準同型暗号 CKKS 方式を用いたガウス過程回帰のアルゴリズム及びプロトコルの提案を行った．また，アルゴリズムの実装を行った後，その性能評価として，scikit-learn の load_diabetes と fetch_california_housing を使用した．これらのデータセットの精度は，cleartext でのガウス過程回帰を用いると，表 2 に示した R2 スコアとなったことから，本データセットは，暗号文のガウス過程回帰の評価に有用であると判断した．訓練フェーズと予測フェーズの漸近計算量は，それぞれ，3.1.1 小節，3.2.1 小節 で示したように，$O(rm^2N^2)$ と $O(m^2nN^2)$ であり，さらに，**図 1**，**図 2**，**表 3** から，これらの実行時間がデータ数に対して 2 乗に比例していることを実験でも確認した．最後に予測フェーズにおいて，f^*, y^* の最大絶対誤差が 10^{-4} 程度であることを確認した．これは予測フェーズに使われる訓練フェーズのパラメータである，$K_{\mathrm{inv}}, \mathrm{yy}$ の誤差が 10^{-4} 程度であることに由来していると考えられる．以上から，暗号文でのガウス過程回帰は，2 つの異なるデータセットにおいて，訓練フェーズ及び予測フェーズともに，データセットのサイズに対して，2 乗に比例する実行時間で算出でき，cleartext のガウス過程回帰とほぼ同じ出力値を得ることができることが示された．今後の展望として，共分散行列の暗号化を完了した上で，提案プロトコルを実装し，通信を含めて性能の評価を行うことが挙げられる．

参考文献

[1] Adi Akavia et al. “Linear-Regression on Packed Encrypted Data in the Two-Server Model”. In: *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography*. WAHC’19. London, United Kingdom: Association for Computing Machinery, 2019, pp. 21–32. ISBN: 9781450368292. DOI: 10.1145/3338469.3358942. URL: https://doi.org/10.1145/3338469.3358942.

[2] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) Fully Homomorphic Encryption without Bootstrapping”. In: *ACM Trans. Comput. Theory* 6.3 (July 2014). ISSN: 1942-3454. DOI: 10.1145/2633600. URL: https://doi.org/10.1145/2633600.

[3] Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[4] Jung Hee Cheon et al. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology – ASIACRYPT 2017*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Cham: Springer International Publishing, 2017, pp. 409–437. ISBN: 978-3-319-70694-8.

[5] Ilaria Chillotti et al. “TFHE: Fast Fully Homomorphic Encryption Over the Torus”. In: *Journal of Cryptology* 33 (Apr. 2019). DOI: 10.1007/s00145-019-09319-x.

[6] Y. Du et al. “Implementing ML Algorithms with HE”. In: *Proceedings of the 14th International Conference on Cryptography and Security* 14 (2017).

[7] Pedro Esperanca, Louis Aslett, and Chris Holmes. “Encrypted accelerated least squares regression”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Ed. by Aarti Singh and Jerry Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, 20–22 Apr 2017, pp. 334–343. URL: https://proceedings.mlr.press/v54/esperanca17a.html.

[8] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. URL: https://eprint.iacr.org/2012/144.

[9] Peter Fenner and Edward Pyzer-Knapp. “Privacy-Preserving Gaussian Process Regression – A Modular Approach to the Application of Homomorphic Encryption”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.04 (Apr. 2020), pp. 3866–3873. DOI: 10.1609/aaai.v34i04.5799. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5799.

[10] Craig Gentry. “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*. STOC ’09. Bethesda, MD, USA: Association for Computing Machinery, 2009, pp. 169–178. ISBN: 9781605585062. DOI: 10.1145/1536414.1536440. URL: https://doi.org/10.1145/1536414.1536440.

[11] Carl Edward Rasmussen. “Gaussian Processes in Machine Learning”. In: Springer Berlin Heidelberg, 2004, pp. 63–71. ISBN: 978-3-540-28650-9. DOI: 10.1007/978-3-540-28650-9_4. URL: https://doi.org/10.1007/978-3-540-28650-9_4.

[12] R.L. Rivest, L. Adleman, and M.L. Dertouzos. “On data banks and privacy homomorphisms”. In: *Foundations of Secure Compuation* (1978), pp. 169–177. URL: https://cir.nii.ac.jp/crid/1570854175774388096.

[13] 磯川，若杉，服部，小寺，野上. “準同型暗号 CKKS 暗号方式における暗号文のランダム逆行列の考察”. In: **電子情報通信学会 情報セキュリティ研究会** *2025* **年 暗号と情報セキュリティシンポジウム** (2025).