明治大学総合数理学部

2016 年度

卒　業　研　究

プライバシーを保護した線形回帰システムの実装と評価

学位請求者　先端メディアサイエンス学科

濱 永 千 佳

目次

第 1 章

はじめに

2015 年に個人情報保護法が改正され，マイナンバー制度も導入された．ベネッセ社で大規模な個人情報流出事件 [1] も発生しており，個人情報保護に関して，社会の関心が高まっている．

本研究では，医療データベースを対象とする．医療データベースには，年齢，性別などの基本情報に加え，既往歴や入退院時の状況，診療情報など，多くの情報が登録されており，ほとんどの項目が個人情報と成り得るからである．プライバシー保護をしつつ情報を活用する方法には匿名加工などのプライバシー保護データパブリッシクングや秘密分散などの様々な手法が提案されている．しかし，特異な症例のデータは匿名加工しても，登録情報から特定可能という問題があった．

そこで，本研究では，準同型性公開鍵暗号を用いることで，データの価値を失うことなく活用するプライバシー保護データマイニング (Privacy Preserving Data Mining, PPDM) を試みる．

5000 例規模の脳卒中患者の実在する患者のデータを用いて，同一の患者群についての情報を持つ 2 つのデータセットを生成する．両データセットは分散管理されており，平文のままで情報を持ち出すこと，活用することが認められていないという状況を想定し，2 者間における垂直分割方式での PPDM を行うことを目的とする．

本稿では，最もシンプルな統計計算である線形回帰を取り上げる．個人情報を秘匿したままで線形回帰を行う際の正確性，パフォーマンスの 2 点を明らかにすることを試み，漏洩の可能性のあるデータについても検討する．

本論文は，6 章で構成されている．研究の背景となっている医療データベースや，解決方法であるプライバシー保護データマイニングについては 2 章で説明する．3 章で，加法準同型公開鍵や作成したシステム scLinear について述べ，システムを用いて行った実験について 4 章で述べる．5 章では，システムの改良を検討するため，完全準同型暗号を用いて考察した．6 章は，まとめとなっている．また，システム scLinear の実行例や実装した機能を紹介するためのコマンド集を付録とした．

第 2 章

プライバシー保護データマイニングと DPC

2.1 プライバシー保護データマイニング

プライバシー保護データマイニング (Privacy Preserving Data Mining, PPDM) とは，複数の人や組織の間で，それぞれが秘匿管理している個人情報を互いに秘匿した状態で，データマイニングを行うことである．秘匿情報を他の人に公開しないよう，暗号化された状態で計算が可能な準同型公開鍵暗号を用いてデータの暗号化を行う．準同型暗号を用いて暗号化されたデータで演算を行い，得られた結果のみを必要な人に渡す．これにより，そのままのデータを公開することなく，所持している秘匿データ同士のデータマイニングを行う．本研究では，2.2.3 に示すユースケースの環境を想定した．

また，プライバシー保護データマイニングを大きく分類し，データを加工することで個人情報保護を目指す技術と，データを暗号化した状態で個人情報を公開することなく新たな知識の発見を目指す技術を言うこともある．この場合のプライバシー保護データマイニングは，入力情報となるデータベースを加工し公開できることを目指す匿名加工技術，データマイニング結果の出力データをノイズを加えることで安全に公開する技術，準同型公開鍵暗号を用いた秘匿計算，データベースを複数に分散させる秘密分散などの技術を総称したものである．

2.2 DPC

2.2.1 DPC データセット

DPC データセットは，病名や治療行為の表コードによる患者の大規模データベースであり，疾患，治療の組合せのデータからなる [2]．年齢，性別，疫病名，重症度，退院時転帰などの診療情報を含んでいる．DPC データベースは 255 万 3283 件のデータが登録されており，内 7 万 828 件が脳梗塞に関する情報である．

2.2.2 実験使用データ

本稿の実験で用いるデータとその統計量を，表 2.1，表 2.2 に示す．表 2.1 は擬似的に合成したもので，表 2.2 は実際の患者のデータである．

表 2.1 は心疾患の擬似患者データである．虚血性心疾患を患ったデータのみを抽出しており，年齢，入院日数が総入院費用にどの程度影を及ぼすのかを実験する．この擬似データの一部を図 2.1 に示す．実験で使用した入院日数は，入院日と退院日から算出している．また総入院費用 call のほかに，検査費 ckensa，画像診断

表 2.1　心疾患データ

変数		データ数	最小・最大値	平均値	管理者
入院日数 (日)	x_1	655	0 - 101	14.56	B
年齢 (歳)	x_2	655	0 - 95	52.25	B
総入院費用 (円)	y	655	597 - 1291937	77930	A

表 2.2　脳卒中データ

変数		最小・最大値	平均値	管理者
Death	y	0 - 1	0.12	A
Age	x_1	40 - 106	72.03	A
Sex	x_2	1 - 2	1.431	B
Japan Coma Scale	x_3	0 - 3	0.957	B
modified Rankin Scale	x_4	0 - 5	3.556	B
Stroke Type	x_5	1 - 3	1.432	B
Liver Disease	x_6	0 - 1	0.022	B

費 cgazo，手術費 cope などの項目もある．

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	age	sex	admdate	disdate	did	dres1	dres2	dcin1	dcin2	dcc1	dcc2	opek1	opek2	opek3	call	ckensa	cgazo	cope
2	54	0	11-Sep-02	4-Oct-02	I	I200		I219				K614-3			331645	20525	1620	27700
3	56	0	14-Sep-02	19-Sep-02	I	I200		F453				K614	K615		175325	5125	405	24020
4	63	0	19-Jul-02	19-Aug-02	I	I200	E119	E119	K295			K5882			554518	17510	10761	184866
5	65	1	22-Jun-02	3-Jul-02	I	I200		E059	I10						49232	7686	960	0
6	66	0	22-Oct-02	26-Oct-02	I	I200		I10	E785						56003	2850	350	0
7	68	0	19-Sep-02	12-Oct-02	I	I200						K614-3	K614-3		452852	15755	4908	48040
8	74	0	12-Jul-02	31-Jul-02	I	I200	N189	I10	E119			K614	K614		702902	12885	1400	53763
9	76	0	3-Sep-02	7-Sep-02	I	I200									58961	3570	425	0
10	77	0	1-Oct-02	13-Oct-02	I	I200						K614-3			157133	8375	2993	24020

図 2.1　擬似 DPC：心疾患データ

後述する実験において，表 2.2 に示す脳卒中の患者のデータを使用する．患者の個人情報 3 項目，脳卒中の分類 1 項目，既往歴 6 項目，入院時の状態 2 項目の中から 7 項目を抽出する．年齢，性別，患者の退院時の生死を表す説明変数 Death に加えて，病歴から肝疾患 Liver Disease の有無を表す項目と入院時の病状から 3 項目を選んでいる．

Japan Coma Scale は，入院時の意識レベルを 4 段階に分類したものであり，数字が小さいほど意識が明瞭である．modified Rankin Scale は，脳卒中により生じる身体障害についての項目であり，数値が大きいほど寝たきりなど重度の障害である．Stroke Type は脳卒中の病型を示しており，脳梗塞，脳内出血，くも膜下出血の 3 分類である [3].

Japan Coma Scale，modified Rankin Scale それぞれの項目における，患者の割合を図 2.2，図 2.3 に示す．Japan Coma Scale は離散値であるが，散布を可視化するため，x，y に小さな (x に標準偏差 0.05 平均 0，y に標準偏差 0.03 平均 0) 正規乱数を加えている．例えば，JapanComaScale= 0，Death= 1 は 12 例である．

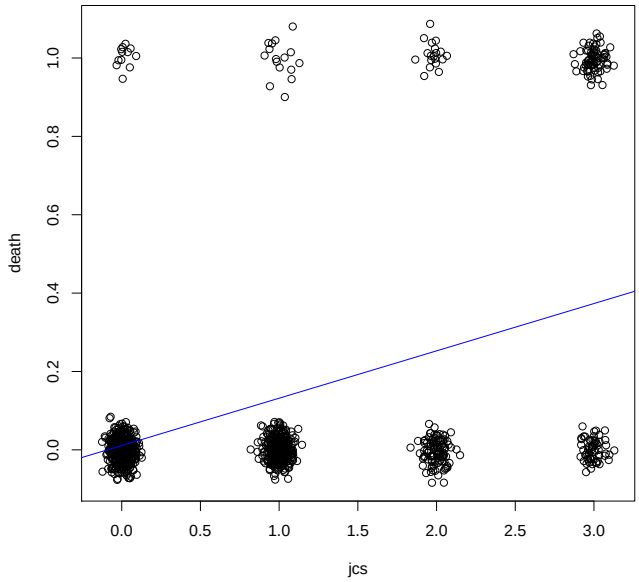

図 2.2　Japan Coma Scale

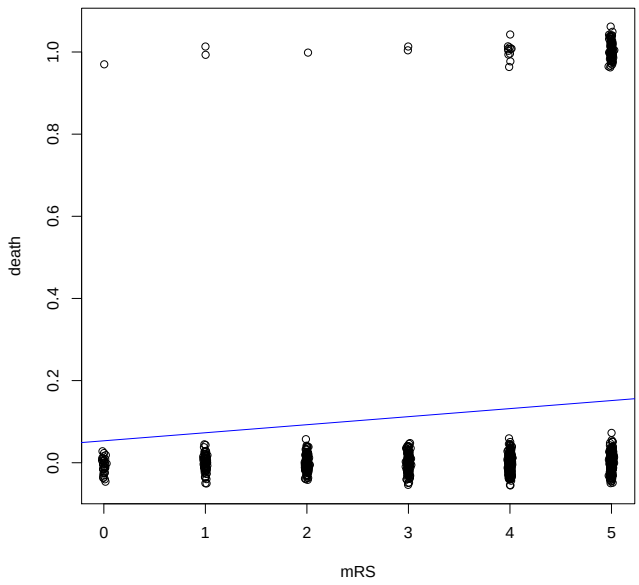

図 2.3　modified Rankin Scale

modified Rankin Scale も同様である (x に標準偏差 0.01 平均 0，y に標準偏差 0.02 平均 0)．死亡 Death の数値が 0 のとき，患者は退院して生存していることを表している．

2.2.3 データの分割について

市町村などの地方自治体とその地域の病院の間において，地域への健康の促進のため等にデータを活用するユースケースを考える．例えば，自治体などの公的機関は死亡届などから直接の原因を知ることができたとしても，既往歴や血圧等の診療情報を知ることは難しい．また，病院などの医療機関は，医療サービスのために個人情報をデータとして持っているが，個人情報保護の観点から外部への情報提供を厳しく管理しており，簡単には情報を提供できない．

そのような場合でも 2 者間で安全にデータを秘匿計算することを目的とする．ユーザ A を官公庁などの公的機関，ユーザ B を病院などの医療機関として想定している．表 2.1，表 2.2 それぞれを 2 つに分け，それぞれが所有しているデータを示す．表 2.1 のデータは，y を持つユーザ A と x_1，x_2 を持つユーザ B に所有されている．表 2.2 については，y，x_1 を持つユーザ A と，それ以外の全てのデータを持つユーザ B に所有されており，互いに所有するデータを秘匿したいと考えているとする．

第 3 章

加法準同型公開鍵暗号を用いたシステム scLinear

3.1 加法準同型公開鍵暗号

準同型公開鍵暗号は，暗号文の状態で演算が可能な公開鍵暗号である．RSA 暗号，ElGamal は乗法準同型性，Paillier 暗号，楕円 Elgamal 暗号は加法準同型性を満たす暗号として知られている．

加法準同型公開鍵暗号は，暗号文同士の加算が可能であり，2 つの暗号文 $Enc(a)$，$Enc(b)$ を与えられた場合に，平文の和である $Enc(a+b)$ を算出することができる．また，Paillier 暗号では暗号文と平文を用いて，定数倍の演算が可能である．暗号文 $Enc(a)$ と定数 c を所持していた場合，$Enc(ca)$ を求めることができる．

3.2 scLinear

提案する方式を，scLinear システムとして実装した．

各々異なるデータセットを有するユーザ A とユーザ B が，互いにデータセットを参照させることなく，安全に正しく線形回帰を実行することを本システムの目的とする．

Paillier 暗号 [4] を用いて，互いのデータを暗号化したままで，線形回帰 $y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m$ を実行し，線形式の係数 α, β_1, β_2, ..., β_m を得る．

単回帰

$$y = \alpha + \beta x \tag{3.1}$$

2 変数の重回帰

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 \tag{3.2}$$

3 変数以上の多変数に対応した重回帰

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_m x_m \tag{3.3}$$

と表し，3 つに分けて方式を提案する．ここで，レコード数を n，説明変数 x の数を m とする．

3.3 単回帰

目的変数 y と（3.1）式の右辺の差の二乗の総和 $S = \sum_{i=1}^{n}(y_i - \alpha - \beta x)^2$ を求め，傾き β，切片 α について偏微分した．偏導関数を 0 として，次の方程式を立てる．

$$\begin{cases} \dfrac{\partial S}{\partial \alpha} = -2\sum_{i=1}^{n}(y_i - \alpha - \beta x_i) = 0 \\ \dfrac{\partial S}{\partial \beta} = -2\sum_{i=1}^{n} x_i(y_i - \alpha - \beta x_i) = 0 \end{cases}$$

これを解いて，傾き β は，

$$\beta = \frac{n\sum_{i=0}^{n} x_i y_i - \sum_{i=0}^{n} x_i \sum_{i=0}^{n} y_i}{x_i^2 - (\sum_{i=0}^{n} x_i)^2} = \frac{D}{C} \tag{3.4}$$

切片 α は，

$$\alpha = \frac{\sum_{i=0}^{n} x_i^2 \sum_{i=0}^{n} y_i - \sum_{i=0}^{n} x_i \sum_{i=0}^{n} x_i y_i}{x_i^2 - (\sum_{i=0}^{n} x_i)^2} = \frac{E}{C} \tag{3.5}$$

により算出される．ユーザ B は秘匿計算を用いて，A には x_i を見せずに，傾き β，切片 α の 2 値をユーザ A の持つ y の暗号文とユーザ B の持つデータで計算する．ユーザ A は復号することで傾き β，切片 α だけを得る．ユーザ各々が持つ情報を表 3.1 に示す．

表 3.1　ユーザが持つ情報

	ユーザ A	ユーザ B
担当	暗号・復号	回帰計算
所持データ	$y_1, y_2, \cdots, y_n$	$x_{1,1}, x_{2,1}, \cdots, x_{n,1}$
所持暗号情報	秘密鍵 $(p, q), n$	公開鍵 (n, g)

垂直分割での計算にあたり，加算と乗算は秘匿して行ったが，$\frac{D}{C}$，$\frac{E}{C}$ は平文で求める．傾き β，切片 α は，割り切れない数をとることが多く，乗法逆元を求めて掛け算としても，元の数値には戻らない．従って，プログラムとして実装した秘匿計算は，和，差，積である．

この問題を解決するため，Algorithm 1 に示す単回帰の提案方式では，傾き β，切片 α についての計算を，ユーザ B が暗号文 $Enc(C)$，$Enc(D)$，$Enc(E)$ をユーザ A に渡し，ユーザ A が復号した後で，式 (3.4)，(3.5) より傾き β，切片 α を算出する．

3.4　2 変数の重回帰

2 変数に限定した重回帰 $y = \alpha + \beta_1 x_1 + \beta_2 x_2$ については，単回帰と同じく，偏微分を用いて解く．目的変数 y と (3.2) 式の右辺の差の二乗の総和 $S = \sum_{i=1}^{n}(y_i - \alpha - \beta_1 x_{1i} - \beta_2 x_{2i})^2$ を求め，傾き β_1，β_2，切片 α

Algorithm 1：scLinear(単回帰)	
	A	暗号鍵を生成
	$A \to B$	公開鍵を共有
1.	A	データ y_1, y_2, ..., y_n を暗号化.
	$A \to B$	$Enc(y_1)$, ..., $Enc(y_n)$　を送る.
2.	B	データ $x_{1,1}$, $x_{2,1}$, $\cdots$,$x_{n,1}$ から,
		$Enc(C) = n\sum x_i^2 - \sum x_i \sum x_i$
		$Enc(y)$, データ x から,
		$Enc(D) =$
		$(\prod Enc(y_i)^{x_i n})((\prod Enc(y_i))^{\sum x_i})^{-1}$
		$Enc(E) =$
		$(\prod Enc(y_i))^{\sum {x_i}^2}((\prod Enc(y_i)^{x_i})^{\sum x_i})^{-1}$
		を出力.
3.	$B \to A$	$Enc(C)$, $Enc(D)$, $Enc(E)$ を送る.
4.	A	復号し, C,D,E を求め,
		$\beta = \dfrac{D}{C}$, $\alpha = \dfrac{E}{C}$ より,
		傾き β, 切片 α を求める

について偏微分した．偏導関数を 0 として，方程式

$$\begin{cases} \dfrac{\partial S}{\partial \beta_1} = -2\displaystyle\sum_{i=1}^{n} x_{1i}(y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \alpha) = 0 \\ \dfrac{\partial S}{\partial \beta_2} = -2\displaystyle\sum_{i=1}^{n} x_{2i}(y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \alpha) = 0 \\ \dfrac{\partial S}{\partial \alpha} = -2\displaystyle\sum_{i=1}^{n} (y_i - \beta_1 x_{1i} - \beta_2 x_{2i} - \alpha) = 0 \end{cases}$$

を解いて，傾き β_1，β_2，切片 α を

$$\beta_1 = \frac{\sum x_{1i} y_i \sum x_{2i}^2 - \sum y_i x_{2i} \sum x_{1i} x_{2i}}{\sum x_{1i}^2 \sum x_{2i}^2 - \sum {x_{1i} x_{2i}}^2} = \frac{D_2}{C_2}$$

$$\beta_2 = \frac{\sum x_{2i} y_i \sum x_{1i}^2 - \sum y_i x_{1i} \sum x_{1i} x_{2i}}{\sum x_{1i}^2 \sum x_{2i}^2 - \sum {x_{1i} x_{2i}}^2} = \frac{E_2}{C_2}$$

$$\alpha = n\sum y_i - \beta_1 \sum x_{1i} - \beta_2 \sum x_{2i}$$

で算出する（Algorithm 2）.

3.5　多変数に対応した重回帰

2 変数に限らない重回帰を提案する．この重回帰を実装するにあたり，[5] において提案された，水平分割方式と垂直分割方式での多変数の重回帰を参考とした．[5] の著者らとともに，垂直分割方式について再検討し，実在する患者データでの実験と 3.4 で提案する 2 変数の重回帰との比較を行った [7]．計算方法を以下に示す．

Algorithm 2：scLinear(2 変数の重回帰)	
1.	Algorithm 1 の 1 までは同じ．
2. B	データ x_1，x_2，$Enc(y)$ から， $\sum x_1$，$\sum x_2$，$\sum y$ を求め， $Enc(C_2) = \sum x_1^2 \sum x_2^2 - \sum x_1 {x_2}^2$ $Enc(D_2) =$ $(\prod Enc(y_i)^{x_{1i}})^{\sum {x_{2i}}^2} ((\prod Enc(y_i)^{x_{2i}})^{\sum x_{1i}x_{2i}})^{-1}$ $Enc(E_2) =$ $(\prod Enc(y_i)^{x_{2i}})^{\sum {x_{1i}}^2} ((\prod Enc(y_i)^{x_{1i}})^{\sum x_{1i}x_{2i}})^{-1}$ を出力．
3. $B \to A$	$Enc(C_2)$，$Enc(D_2)$，$Enc(E_2)$， $Enc(\sum x_1)$，$Enc(\sum x_2)$ を送る．
4. A	復号し，C_2,D_2,E_2,$\sum x_1$,$\sum x_2$ を求め， 傾き β_1，β_2，切片 α を求める．

$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$ について，偏微分を用いる．行列式での計算を用いて，線形式の係数 α, β_1, β_2, $\cdots$, β_m を算出する（Algorithm 3）．

$S = \sum (y_i - \alpha - \beta_1 x_{1i} - \cdots - \beta_m x_{mi})^2$ を求め，それらを最小化する各係数 β を，総和の微分を 0 とおいて次の方程式を立てる．

$$\begin{cases} \dfrac{\partial S}{\partial \alpha} = -2 \displaystyle\sum_i^n (y_i - \alpha - \beta_1 x_{i,1} - \cdots - \beta_m x_{i,m}) = 0 \\ \dfrac{\partial S}{\partial \beta_1} = -2 \displaystyle\sum_i^n x_{i,1} (y_i - \alpha - \beta_1 x_{i,1} - \cdots - \beta_m x_{i,m}) = 0 \\ \cdots \end{cases}$$

これらを整理して，$m+1$ 個の連立方程式を

$$FX = G \tag{3.6}$$

と行列で表し，これを満たす X を求めればよい．ここで，

$$F = \begin{pmatrix} \sum x_{i,1}^2 & \sum x_{i,1}x_{i,2} & \cdots & \sum x_{i,1} \\ \sum x_{i,1}x_{i,2} & \sum x_{i,2}^2 & \cdots & \sum x_{i,2} \\ \vdots & & & \vdots \\ \sum x_{i,1} & \sum x_{i,2} & \cdots & \sum 1 \end{pmatrix},$$

$$X = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \\ \alpha \end{pmatrix}, \qquad G = \begin{pmatrix} \sum x_{i,1} y_i \\ \vdots \\ \sum x_{i,m} y_i \\ \sum y_i \end{pmatrix}$$

とする．F の逆行列を左からかけて，係数を得る．

Algorithm 3：scLinear(多変数の重回帰)	
1.	Algorithm 1 の 1 までは同じ．
2. B	行列 F，G を求める．
	（ここで，A のみで求められるものは計算しない．）
3. $B \to A$	$Enc(F)$，$Enc(G)$ を送る．
4. A	復号し，F，G を求め，
	$FX = G$ となる X を求める．

表 3.2　提案手法の比較

	(3.1) 単回帰	(3.2) 2 変数の重回帰	(3.3) 多変数の重回帰 $(m = 2)$
モデル	$y = \alpha + \beta x$	$y = \alpha + \beta_1 x_1 + \beta_2 x_2$	$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m$
B から A へ送るデータ	C，D，E	C_2，D_2，E_2，$\sum x_1$，$\sum x_2$	F_2，G_2
暗号文数	3	5	7

3.6　3 方式の比較

表 3.2 で，提案方式の比較を示す．多変数の重回帰については，2 変数の場合と比べやすくするため $m = 2$ の場合で記載している．m の場合においては，F からは $\frac{(m+m^2)}{2} - 1$ 個，G からは m 個の情報を送信している．しかし，計算の面では，合計値のみを求める多変数の重回帰に対し，係数を求める際に必要となる最小限の数値のみをユーザ A に渡す 2 変数の重回帰の方が，秘匿計算において煩雑な計算を要求する．

$m = 2$ の場合において，暗号化した状態でユーザ B がユーザ A に公開する情報を比較する．Algorithm 2 で送るデータは，C_2，D_2，E_2，$\sum x_1$，$\sum x_2$ の 5 つである．C_2，D_2，E_2 は計算途中の数値であり，$\sum x_1$，$\sum x_2$ は統計値である．一方，Algorithm 3 で送るデータは，

$$F_2 = \begin{pmatrix} \sum x_1^2 & \sum x_1 x_2 & \sum x_1 \\ \sum x_1 x_2 & \sum x_2^2 & \sum x_2 \\ \sum x_1 & \sum x_2 & \sum 1 \end{pmatrix},$$

$$G_2 = \begin{pmatrix} \sum x_1 y_i \\ \sum x_2 y_i \\ \sum y_i \end{pmatrix}$$

であり，7 つの情報を送信する．($\sum x_1 x_2$ などの重複する情報を除いてカウントする．また，$\sum y_i$，$\sum 1$ のユーザ A のみで計算できるものは除いている．) この Algorithm 3 で公開する情報は，全て統計値である．

Algorithm 2 と 3 の違いは，計算途中の結果を公開して情報量を少なくするか，統計値を公開することで情報量を多くするかである．

第 4 章

実験

4.1 processing での試作プログラム：暗号化なしの線形回帰

3.1 に示した線形回帰式の計算結果が正確性があるかを検証するため，暗号を使用せずに回帰計算を行うプログラムを作成した．実験環境は表 4.1 に示す実験 4.1 で行い，表 2.1 のデータを使用した．

表 4.1　実験環境

	実験 (4.1，4.2.3)	実験 (4.2.4)	実験 (4.3)	改良の検討
OS	Windows 7	Windows 7	OS X El Capitan	
メモリ	4 GB	11.7 GB	8 GB	
CPU	Intel(R) Core(TM) i5	Intel Xeon X5460	Intel(R) Core(TM) i5	
クロック	1.8GHz	3.16 GHz	2.9 GHz	
使用言語	Java(1.8.0_91-b14) R(3.1.0)	Java(1.8.0_45-b15) R(3.1.2)	Java(1.8.0_111-b14) R(3.3.1)	C/C++
鍵長	2048[bit]			

Processing でのプログラムで実行した結果を表 4.2 に示す．試作プログラムが正確に実行できているかを検証するために，統計ソフト R の lm 関数を用いて同じデータの組み合わせで実行した結果と比較する．

試作プログラムの実行結果は，R での実行結果と比べて少数点第 5 位までに差が見られなかった．傾き β の少数点第 6 位のみが異なる結果を示しているが，これは R と試作プログラムのでの少数点以下の設定の違いによるものであり，3.1 に示す回帰計算式を用いた計算は，大きな誤差がなく傾き β，切片 α を算出することができると分かった．

表 4.2　試作プログラムと R の比較

	試作プログラム	R
傾き β	0.011508	0.011509
切片 α	0.0945	0.094500

4.2 scLinear の実験 (1)

表 4.1 の環境上に scLinear システムとして実装した．

4.2.1 実験目的

システム scLinear を用いて，以下の 2 項目を評価する．

1. 本システムの計算結果の正確性．
2. 本システムのパフォーマンス．

4.2.2 実験方法

DPC データセットについて線形回帰を実施する．本研究では，2 つの実験から，提案方式について評価する．ユーザ間でのデータのやりとりはネットワーク通信で行われることが多いが，本システムでは通信の過程を省略し，通信内容を一次ファイルに出力する手法をとっている．

単回帰，2 変数の重回帰

擬似 DPC データについて単回帰，2 変数の重回帰をそれぞれ 10 回ずつ実施する．$n = 100$ 行，300 行，500 行，655 行の 4 つの異なるデータセットについて測定する．

多変数に対応した重回帰

$m = 3$，4，5，6 の重回帰を実施する．$n = 1000$ 行，2000 行，5000 行のデータセットについて測定し，提案方式について評価する．

4.2.3 単回帰，2 変数の重回帰

正確性

表 4.3 に単回帰の計算結果を，表 4.4 に 2 変数の重回帰の計算結果を示す．

scLinear の実行結果は，単回帰においては R の計算結果と差がなかったが，2 変数の重回帰については，小数第 3 位以降に R の結果との違いが見られた．$n = 655$ の β_1，α と，$n = 300$ における α である．特に α での誤差が目立つが，本システムが小数点第 5 位まで算出して各変数を求めており，β_1，β_2 を求めた後に $\alpha = n\sum y - \beta_1 \sum x_1 - \beta_2 \sum x_2$ の算出を行っているため，β_1，β_2 でのわずかな誤差が累積していることが原因であると考えている．scLinear は，単回帰においては非常に高い精度（小数第 3 位）で，2 変数の重回帰においては小数点第 2 位までの精度を持つと分かった．

パフォーマンス

図 4.1 と図 4.2 に scLinear の処理時間を示す．ここで，点線は 2 変数の重回帰を，実線は単回帰の結果を示している．

Algorithm1 の step2 を行なう Linear クラスにおいて，単回帰分析は 1 データあたりの平均処理時間が 1.9 ミリ秒であった．一方，Algorithm2 では，1 データに対する計算が 1 つ多く，秘匿計算において大きなレコー

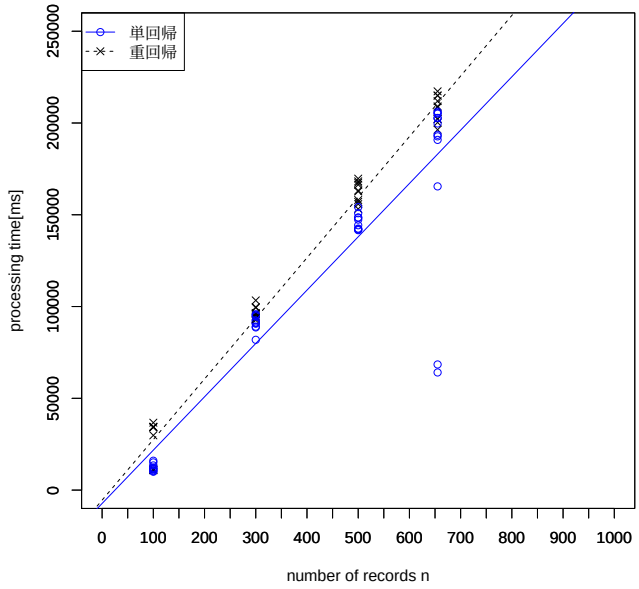

図 4.1　実験 4.2.3：システム実行時間

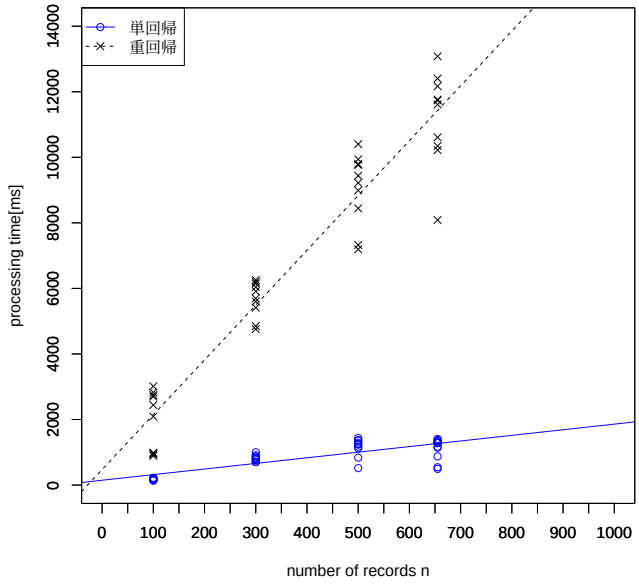

図 4.2　実験 4.2.3：Linear クラス実行時間

表 4.3　実行結果（単回帰）

	傾き β		切片 α	
n	scLinear	R	scLinear	R
655	5099.358	5099.358	5002.521	5002.521
500	67751.810	67751.810	1021.751	1021.751
300	72369.082	72369.082	322.378	322.378
100	89508.076	89508.076	786.790	786.790

表 4.4　実行結果（2 変数の重回帰）

	傾き β_1		傾き β_2		切片 α	
n	scLinear	R	scLinear	R	scLinear	R
655	4995.554	4995.555	41.304	41.304	3042.752	3042.759
500	998.417	998.417	245.842	245.842	54332.010	54332.010
300	302.882	302.882	128.136	128.136	65339.083	65339.084
100	4730.629	4730.629	273.939	273.939	1744.523	1744.523

表 4.5　実験 4.2.3：1 データあたりの平均実行時間 [ms]

モデル	暗号化処理 ユーザ A	秘匿計算実行 ユーザ B	復号化と算出 ユーザ A
単回帰	320	1.9	1.0
2 変数の重回帰	320	16.7	0.6

ドの数値（例えば，日数 50 日）での 50 乗を計算するなど，複雑な計算を要求される．1 データあたり平均 16.7 ミリ秒を要した．暗号化の処理時間には平均 320 ミリ秒と単回帰，重回帰に差が見られなかった．復号化については，単回帰は平均 1.0 ミリ秒，重回帰は平均 0.6 ミリ秒であり，大きな差はなかった（表 4.5）．

4.2.4　多変数に対応した重回帰

正確性

表 4.6 に 6 変数の重回帰での計算結果を示す．秘匿計算を行った結果を scLinear に，R での実行結果を coefficient に示している．

scLinear の実行結果は，R の結果と差が見られなかった．また，$n = 1000$，2000 においても差がなかったため，scLinear は正確に計算できていると言える．

パフォーマンス

図 4.3，図 4.4 に異なる条件下での scLinear の処理時間を示す．図 4.3 は，サイズの異なる 3 つのデータセットにおけるシステムの実行時間について，図 4.4 は同じデータにおける変数 m を変えた場合の，暗号化処理を除いたシステム実行時間である．ここで，一点鎖線はユーザ A の処理時間を，点線はユーザ B の処理時間を，破線はデータの暗号化にかかった時間を，実線は合計時間を示している．

表 4.6　線形回帰モデルの係数と提案方式の比較 ($n = 5000$)

	提案方式	R				
variables	scLinear	coefficient	Std. Error	t value	$Pr(>\|t\|)$	
α	-0.1731982	-0.1731982	0.0290099	-5.970	$2.53e-09$	***
Age	0.0015410	0.0015410	0.0003576	4.310	$1.67e-05$	***
Sex	-0.0217865	-0.0217865	0.0083993	-2.594	0.009519	**
JapanComaScale	0.1283596	0.1283596	0.0049296	26.039	$< 2e-16$	***
modifiedRankinScale	0.0121227	0.0121227	0.0034845	3.479	0.000507	***
StrokeType	0.0292522	0.0292522	0.0073582	3.975	$7.12e-05$	***
LiverDisease	0.0095770	0.0095770	0.0324591	0.295	0.767970	

図 4.3 より，レコード数 n に線形に，システムの処理時間が増加していることが分かる．また，図 4.4 より，変数 m に対して線形であり，計算に使用する変数が増加すると，比例して全体の処理時間が増加している．(図 4.3 におけるシステム全体の実行時間 Total は $y = 224.576n + 9023.692$ で，暗号化処理にかかる時間は $y = 223.374n - 1649.538$，n はレコード数である)．$n = 100$ 万件のデータセットを実行した場合，システム全体の実行は

$$225 \times 10^6 + 9024 = 2.24 \times 10^8[\text{ms}] = 2.60[\text{day}]$$

かかる．そのほとんどは暗号化処理にかかる

$$223 \times 10^6 - 1650 = 2.23 \times 10^8[\text{ms}] = 2.59[\text{day}]$$

時間であり，その後の計算のみであれば約 20 分で実行できる．

4.2.5　実験 4.2.3，4.2.4 考察

多変数に対応させた場合，ユーザ各々でデータを処理する時間が増加しているが，2 変数の重回帰では，ユーザ A の処理時間は最大でも 2000[ms] と，n に依存しない．

実験 4.2.3 において Algorithm 1 の step2 を行なう Linear クラス以外の処理時間では大きな差が見られなかったことから，Linear クラスの処理時間がシステム実行時間に大きな影響を与えていると考えられる．2 変数の場合は，計算途中のデータをどの程度秘匿したいかによってアルゴリズムを選択して使用できると考える．

3 つのアルゴリズムを提案したが，どの計算方法でも小数点第 5 位程度までは R と非常に近い数値を求めることができていた．

表 4.6 より，死亡という目的変数に対して，Japan Coma Scale，modified Rankin Scale，Stroke Type だけでなく，年齢，性別が生死に影響を与えていると分析できた．Japan Coma Scale の係数は 0.1283596 であり，入院時の意識レベルが 1 大きく（より悪く）なると，死に近づきやすくなることを示している．

scLinear を用いて，脳卒中の患者の DPC データについて考察し，入院時の意識の状態 Japan Coma Scale が，係数の絶対値が最大であることを示した．すなわち，死亡 Death という結果に対して最も支配的である．

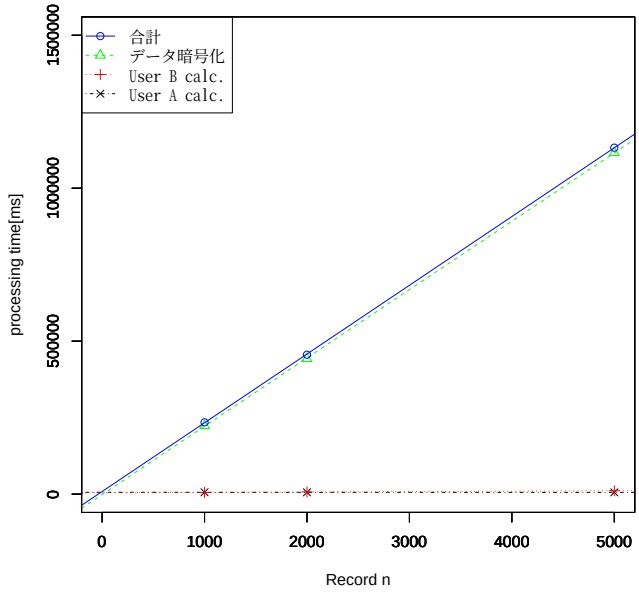

図 4.3　実験 4.2.4：レコード数 n におけるシステムの実行時間

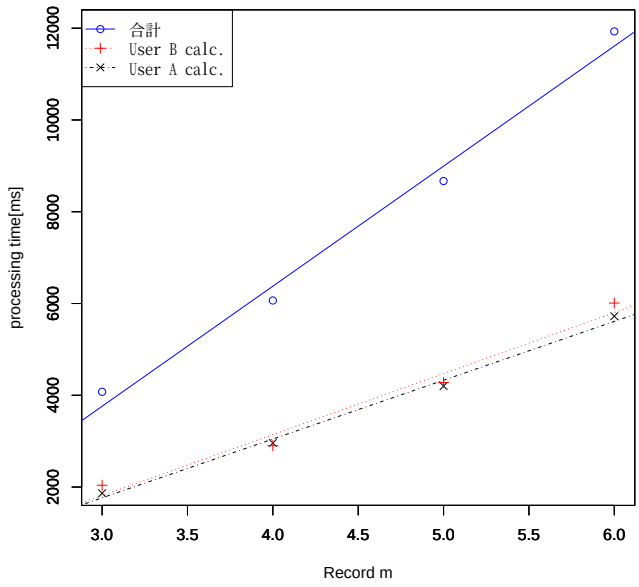

図 4.4　実験 4.2.4：変数 m におけるシステム実行時間（$n = 1000$，暗号化時間は除く）

4.3 実験 (2)

4.2.4 で行った実験結果の図 4.4 より，ユーザの実行時間は変数 m が増加に対して線形に増加することが分かる．しかし，レコード数 n とユーザそれぞれの実行時間は全体の実行時間に比べてごくわずかであったため，図 4.3 では分かりやすく示すことができていない．

そこで，多変数の重回帰の実験を行い，ユーザそれぞれの実行時間の傾向を評価する．

4.3.1 実験方法

実験 4.2.4 で用いたデータセットが用意できなかったため， 表 4.7 に示す同じ項目で構成されたレコード数 $n = 1000$ の擬似 DPC データセットを使用した．

4.3 の環境において，$m = 6$ の重回帰を実施する．$n = 10$ 行,100 行,1000 行のデータセットについてそれぞれ 10 回ずつ測定した．

表 4.7　擬似：脳卒中データ

変数		最小・最大値	平均値	管理者
Death	y	0 - 1	0.10	A
Age	x_1	26 - 108	72.68	A
Sex	x_2	0 - 1	0.458	B
Japan Coma Scale	x_3	0 - 3	0.947	B
modified Rankin Scale	x_4	0 - 5	3.458	B
Stroke Type	x_5	0 - 2	0.404	B
Liver Disease	x_6	0 - 1	0.014	B

4.3.2 実験 (2) 結果

図 4.5 にシステム全体の実行時間を，図 4.6 にユーザ A，B それぞれの処理時間を示す．

システムの実行時間の図 4.5 は実行時間 [ms] は異なるが，実験 4.2.4 で得られた図 4.3 と同じく，システム全体の実行時間が線形に増加していることを示している．

5000[ms] までの実行時間を細かく表示した結果が図 4.6 である．図 4.6 より，計算を行うユーザ B はレコード数 n に対して線形に増加しており，復号を行うユーザ A はレコード数 n に依存していないことが分かる．

この実験ではレコード数 n が 1000 行であったが，100 万件を超える大きなデータベースで実行する場合においても，ユーザ A の処理時間は n に依存しないため 4000[ms] ほどで処理できると考えられる．

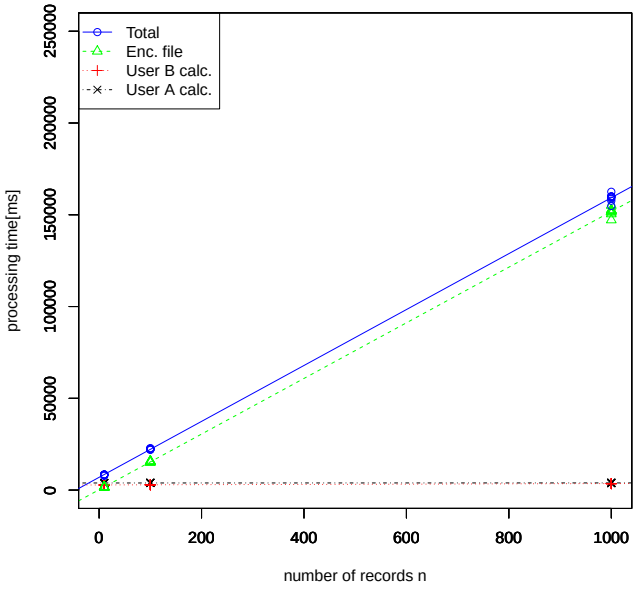

図 4.5　実験 4.3：システムの実行時間

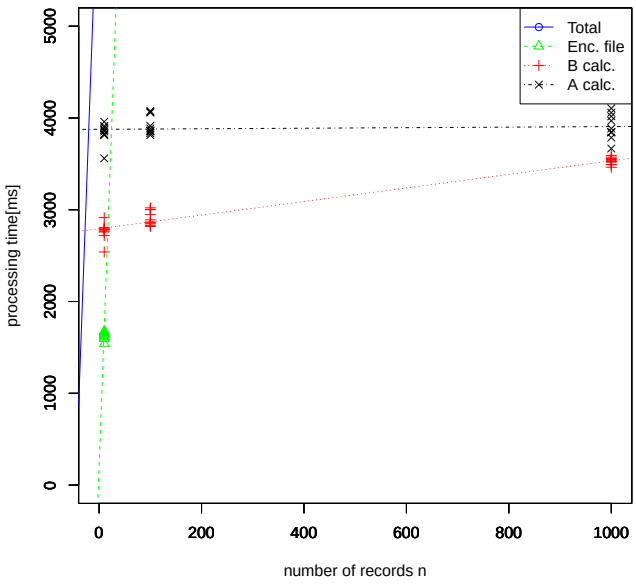

図 4.6　実験 4.3：ユーザの処理時間

第 5 章

完全準同型公開鍵暗号と scLinear の改良の検討

提案方式では，計算途中の数値やデータのうち統計量を相手ユーザに提示していた．しかし，医療データであっても統計量を相手に公開することができない場合にもデータを活用できるよう，全ての過程を暗号化したままで実装したい．

5.1 完全準同型暗号

完全準同型暗号（Fully Homomorphic Encryption）は，暗号文同士の計算で加法，乗法のどちらも求められる暗号である．

任意の計算を暗号文同士から行うことができるが，計算が煩雑になることで処理時間が長くなる，ノイズが起こり誤差が生じやすくなるなどの課題もある．

5.2 HElib

HElib は IBM 社が公開している，完全準同型暗号を C++ 上に実装したライブラリ [6] である．提案システムで用いていた Paillier 暗号は，暗号文同士の足し算は行えるが，掛け算は平文と暗号文での計算であった．完全準同型暗号ライブラリの HElib は，暗号文同士での足し算，掛け算の両方を行うことができる．計算結果に大きな誤差が起こらないよう，暗号文ノイズを削減する手法 bootstrapping もライブラリに含まれている．

5.3 改良の検討

scLinear システムでの提案では，Paillier 暗号を使用した．Paillier 暗号は加法準同型であるため，暗号文同士の加算と，暗号文と平文を用いた定数倍のみの演算が可能であった．しかし，暗号文同士での乗算が計算できないため，scLinear では単回帰の場合においても 3.4 の $\frac{D}{C}$ の演算ができず，計算途中の情報を相手ユーザに送信し，復号を行った後に傾き β，切片 α を求めていた．また，重回帰においても，乗算ができないために 2 変数の重回帰 3.2 で提案した計算途中の情報もしくは，多変数の重回帰 3.3 で提案した統計量のどちらかを，相手ユーザに渡す方式で提案した．

そこで，暗号文同士の乗算を含む全ての計算が暗号文同士で計算できれば傾き β，切片 α を計算を行うユー

表 5.1　HElib での実行時間 [sec]

	暗号化	復号	加算	乗算
最長	0.119506	0.139702	0.001142	0.144726
最短	0.100093	0.118716	0.000424	0.121745
平均	0.106915	0.128065	0.000541	0.130216
標準偏差	0.003698	0.004574	0.000089	0.003918

表 5.2　処理時間の比較 ($m = 6$，$n = 1000$)

	scLinear	HElib での推定	平均時間	推定処理回数
暗号化	223.742[sec]	749.262[sec]	0.106915[sec]	7008
計算	6.010[sec]	783.765[sec]	0.130216[sec]	5994
復号	5.736[sec]	4.354[sec]	0.128065[sec]	34
合計	235.488[sec]	1537.382[sec]		

ザのみで算出できるのではないかと考える．

完全準同型暗号の実行速度を知るため，scLinear で Paillier 暗号を用いて提案した方式を，完全準同型暗号ライブラリである HElib に置き換えた場合のパフォーマンスを推測する．

5.3.1　HElib を用いた提案方式

scLinear で提案した方式 3.3 の Algorithm 3 を，秘匿計算を行う部分を Paillier から HElib に置き換えたシステムを実装できた場合の実行時間をを推定する．$m = 6$，$n = 1000$ とする．ユーザ B が所有しているデータ同士の計算は，暗号化せずに行うこととし，HElib の測定と合わせて C 言語での暗号文を用いない計算の平均時間を計測した．ユーザが行う暗号化，計算，復号の推定処理回数は表 5.2 に示す．表 5.2 の計算は，HElib での加算，乗算と C 言語での加算，乗算の 4 つを用いて推定した．それぞれの平均時間と推定処理回数を，表 5.3 に示す．表 5.2 には，最も時間を要するとした HElib 乗算の結果を表記している．

5.4　HElib を用いたパフォーマンス測定実験

HElib を用いて，暗号化，足し算，掛け算，復号のそれぞれにおける時間を計測した．計測を行った環境は，表 4.1 である．100 回行い，それぞれのフェーズでの平均時間を求めた．結果を表 5.1 に示す．

表 5.1 の結果を用いて，3.3 で提案した Algorithm 3 に基づいて HElib 上でシステムを実装できた場合の実行時間を推定し，提案方式 3.3 と比較する．1000 行のデータにおける，表 5.1 での平均時間より推定した実行時間を表 5.2 に示す．

HElib での推定時間のほうが，暗号化，線形回帰計算を行う部分の処理時間が長くかかる結果になった．Algorithm 3 で述べたことより，scLinear で行う暗号化はユーザ A が所有する項目と F の数値であり，7008 回の暗号化を行う HElib と比べて処理時間が半分以下で済んでいる．これは，scLinear に実装した乗法計算が加法準同型性であり，暗号文と平文の計算で乗算の演算ができたことが，暗号化にかかる時間に関係しているのではないかと推測する．

表 5.3　計算：推定処理時間

	平均時間	推定処理回数
HElib 加算	0.000541[sec]	6000
HElib 乗算	0.130216[sec]	5994
C 加算	0.0000001[sec]	28000
C 乗算	0.0000001[sec]	28000

HElib での加算，乗算と，C での暗号文を用いない加算，乗算の平均時間を表 5.3 に示し，表 5.2 の状況において推定した処理回数を記載した．Algorithm 3 と同条件にするため，ユーザ B のみで算出可能な F の数値は暗号化を用いず，計算を行うようにした．

表 5.3 より，HElib の乗算に最も時間がかかっていることがわかる．また，暗号化を用いない計算の処理時間は合計で 0.0028（加算のみ）となり，HElib を用いた計算処理時間に比べると，加算乗算を合わせても HElib の加算 1 回の平均時間以下である．HElib の計算の推定処理時間は，scLinear の処理時間と比較しても膨大であり，実装する際には計算の高速化を図る必要があることが分かった．

第 6 章

おわりに

医療機関と公的機関を例とした 2 組織間で，各々が情報を暗号化して秘匿したまま線形回帰を求めるプロトコルを実装し，その性能を評価した．実際の DPC データを用いて実験を行い，どの方法でも正確性のある結果を算出することができることを示し，パフォーマンスを評価した．

統計値のみの公開であっても，相手ユーザに提示する情報があり，暗号化したままではあるが公開している．そこで，相手に漏れてしまう情報をなくすべく，提案システムの改良についても検討した．HElib を用いた場合，すべての情報を暗号化して計算を行うため，提案システムより処理時間が長くなることが想定される．今後，HElib を用いたシステムを実装する場合においては，どう計算処理の高速化を図るかが課題である．

謝辞

本論文の作成にあたり，様々なご指導を頂きました指導教官の菊池浩明教授に感謝の意を表します．東京大学大学院医学系研究科の康永秀生先生，松居宏樹先生，橋本英樹先生には，実験に使用した DPC データセットを提供していただき，医療情報についてのご指導をいただきました．心より感謝いたします．

参考文献

[1] ベネッセホールディングス，“事故の概要”（http://www.benesse.co.jp/customer/bcinfo/01.html，2015 年 6 月参照）

[2] 松田，伏見，“診療情報による医療評価：DPC データから見る医療の質”，東京大学出版会．2012.

[3] 日本脳卒中学会，“脳卒中ガイドライン 2009”（http://www.jsts.gr.jp/jss08.html，2016 年 5 月参照）

[4] P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT 1999, pp.223-238，1999.

[5] 菊池，橋本，康永，“DPC データベースからのプライバシーを保護した線形回帰による入院日数モデルの学習”，DICOMO2014 シンポジウム，pp．219-223，2014.

[6] shaih/HElib（https://github.com/shaih/HElib，2016 年 5 月参照）

[7] 濱永，菊池，康永，松居，橋本，“プライバシーを保護した垂直分割線形回帰システムの実装と DPC データセットを用いた評価”，DICOMO2016 シンポジウム，情報処理学会，pp.1471-1478，2016.

付録 A

scLinear 実行例・コマンド集

scLinear や試作プログラムを作成した．実行例，実行時のコマンドや付属させた機能などを記す．

A.1　実行例

A.1.1　Processing での試作プログラム

Processing での試作プログラムでの実行画面を図 A.1 に示す．一度に複数の回帰計算を求めることのできるプログラムである．

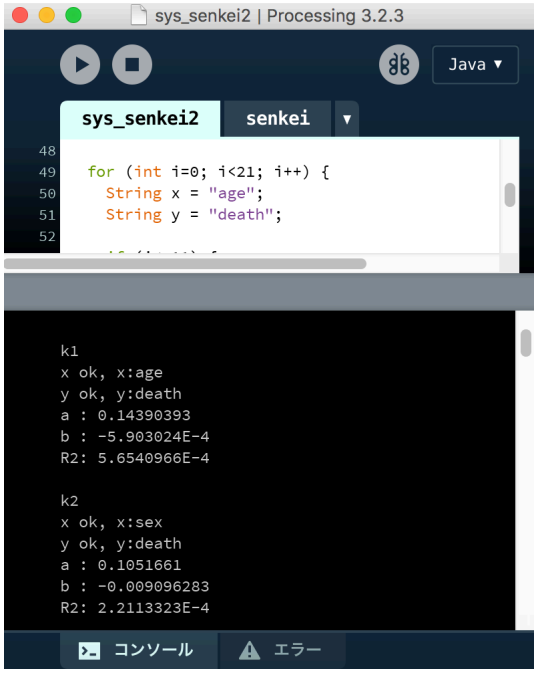

図 A.1　実行画面例

A.1.2　scLinear

scLinear の実行画面例を図 A.2 に示す．実行しているのは，提案方式の 3.3 である．

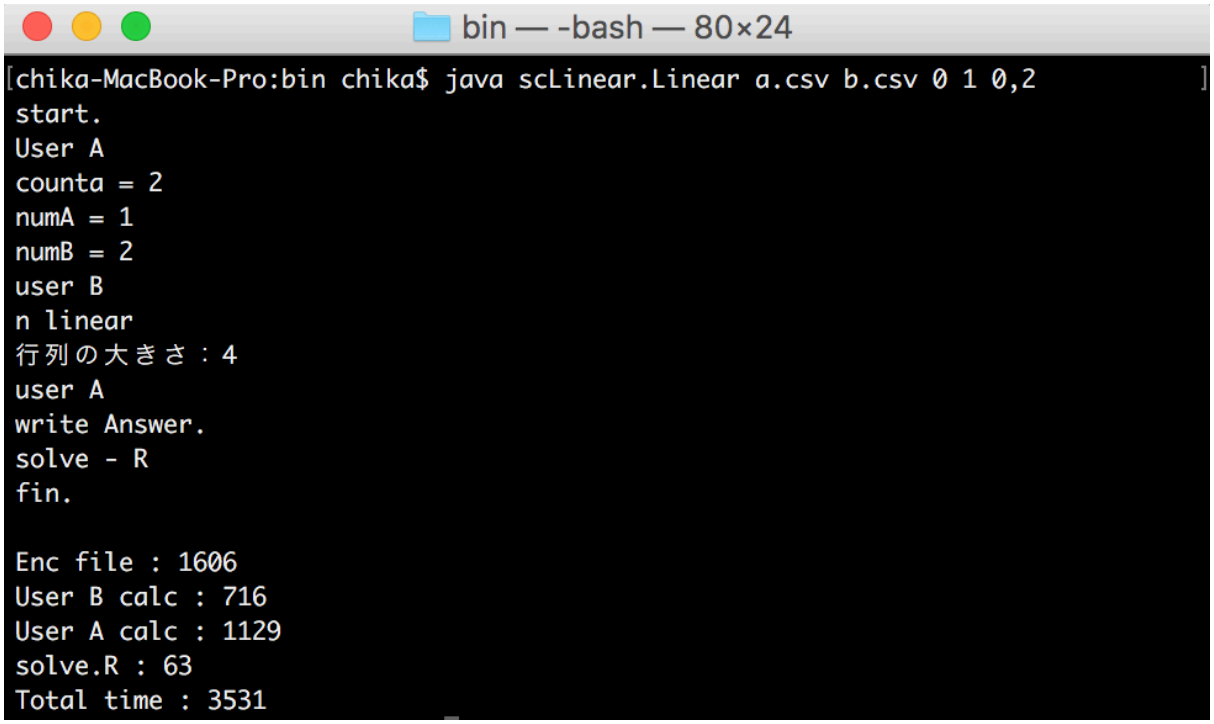

図 A.2　実行画面例

実行画面には，必要な情報だけを表示させている．それぞれの処理時間は ms で表示させているが，傾きや切片，鍵の大きさなどの細かな情報は csv ファイルに書き出している．

A.2 コマンド一覧

＊実行コマンド
(3 変数) java scLinear.Linear a.csv b.csv 0 1 0,2
(4 変数) java scLinear.Linear a.csv b.csv 0 1,2 0,2
第 1 引数：User A のファイル指定 (a.csv)
第 2 引数：User B のファイル指定 (b.csv)
第 3 引数：User A のファイルから，y の列を指定する (0)
第 4 引数：User A のファイルから，x の列を指定する（1,2 複数可
第 5 引数：User B のファイルから，x の列を指定する（0,2 複数可)

＊ファイルについて
a.csv, b.csv：任意の csv ファイル
Key.txt：鍵のファイル
solve.R：R プログラム
Set.txt：使用するファイルの設定
　count：User A の所持ファイルの 行数 を指定する
　aSize：User A の所持ファイルの 列数 を指定する
　encF：User A のファイルの暗号化処理を行うかを指定する
　　F or f の場合は、暗号化処理をスキップする（それ以外は暗号化処理を行う）

＊生成ファイル
Enc_a.csv：User A の所持ファイルから生成した、暗号化ファイル

answerA.csv, answerB.csv：User B calc の結果
　　User A のみで計算可能な部分を-1 と置き、それ以外の部分は暗号化してある
answerA2.csv, answerB2.csv：User A calc の結果
　　answer.csv をもとに平文を生成

result.csv：solve.R、answerA2,answerB2 より求めた結果
　$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_m + \alpha$ のとき
　　result.csv には β_1，β_2，$\cdots$，β_n，α の順で書き込まれている

Time.csv：かかった時間を計測
　　暗号化処理をスキップした場合，Enc file を -1 と表示する