2019-03-14

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Logistic Regression Model Training based on the Approximate Homomorphic Encryption

Andrey Kim[1], Yongsoo Song[2], Miran Kim[2], Keewoo Lee[1], and Jung Hee Cheon[1]

[1] Seoul National University, Seoul, Republic of Korea

[2] University of California, San Diego, United States

Nankai-Baidu
Joint Laboratory

Parallel and Distributed
Software Technology Lab

Contents

- Abstract (iDASH)
- Related Work
- Paper Algorithm
- Paper Technique
- Recent Work

Abstract

- We apply the homomorphic encryption scheme of Cheon et al., and devise a new encoding method. In addition, we adapt Nesterov's accelerated gradient method.
- Our method shows a state-of-the-art performance (2017) of homomorphic encryption system in a real-world application.
- The submission based on this work was selected as the best solution of Track 3 at iDASH privacy and security competition 2017.

Nankai-Baidu Joint Laboratory
Parallel and Distributed Software Technology Lab

Abstract

➢ iDASH

- 2017年10月14日，百度安全实验室参加了第7届 iDASH Privacy & Security Workshop，并受邀进行演讲。该 Workshop 是由 UCSD 和 Indiana University 主办，百度、美国国家人类基因研究院（NHGRI）和 Human Longevity 公司赞助。该 Workshop 致力于解决医疗大数据人类基因组分析中的隐私保护问题，并同时举办 iDASH 竞赛，是医疗数据隐私领域的重要学术会议与赛事。学术界享有盛誉的 ACM SIGSAC 副主席 Xiao feng Wang 教授主持了该会议，来自美国国家卫生研究院（NIH）、微软、Intel、IBM以及来自十几个国家的研究员出席了该会议。
- 微软研究院的Dr. Kristin Lauter做了题为《Cryptographic Tools for Genomic Privacy: Ready for Standardization? 》的演讲，总结了密码学工具在基因组分析隐私保护方向的研究，以及美国在标准化人类基因组分析的隐私保护的努力。Dr. Kristin 认为，基于同态运算的加密和基于 Intel SGX 的加密各有长处，配合使用可以达到保护隐私的效果。

Abstract

- 2017 Track 3: Homomorphic encryption (HME) based logistic regression model learning.

Track 3: Homomorphic encryption (HME) based logistic regression model learning

Jung Hee Cheon (Seoul National University), Andrey Kim (Seoul National University), Miran Kim (UCSD), Keewoo Lee (Seoul National University), and Yongsoo Song (Seoul National University)

- 2018 Track 2: Secure Parallel Genome Wide Association Studies using Homomorphic Encryption.

Track 2: Secure Parallel Genome Wide Association Studies using Homomorphic Encryption

Miran Kim(UTHealth), Baiyu Li(UCSD), Daniele Micciancio(UCSD), Yongsoo Song(UCSD)
Marcelo Blatt(Duality Technologies), Alexander "Sasha" Gusev(Dana Farber Cancer Institute), Yuriy Polyakov(Duality Technologies), Kurt Rohloff(Duality Technologies), Vinod Vaikuntanathan(Duality Technologies)

Related Work

- Naehrig et al. and Bos et al. both papers assume that the logistic model has already been trained and is publicly available.
- Aono et al. they shift the computations that are challenging to perform homomorphically to trusted data sources and a trusted client.
- Xie et al. construct PrivLogit which performs logistic regression in a privacy-preserving but distributed manner.
- Kim et al.

  Secure logistic regression based on homomorphic encryption.

Related Work

2017 iDASH Track 3

- **Doing Real Work with FHE: The Case of Logistic Regression**.
  Jack L.H. Crawford, Craig Gentry, Shai Halevi, Daniel Platt, and Victor Shoup.
  不像Logistics Regression，使用了复杂的比较操作，可能实用性不强
- **Logistic regression over encrypted data from fully homomorphic encryption**.
  Hao Chen, Ran Gilad-Bachrach, Kyoohyung Han, Zhicong Huang, Amir Jalali, Kim Laine, and Kristin Lauter. 参考了首尔国立大学密码实验室的方案
- **Logistic Regression Model Training based on the Approximate Homomorphic Encryption**.
  Andrey Kim, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Hee Cheon.
- **Privacy-Preserving Logistic Regression Training**.
  Charlotte Bonte, and Frederik Vercauteren. 近期工作

2018 iDASH Track 2

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Logistic Regression

Gradient Descent

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
-     % compute the gradient $\boldsymbol{g}$
-     $\mathbf{W} = \mathbf{W} - \boldsymbol{\alpha} \cdot \boldsymbol{g}$

Nesterov’s accelerated gradient method

Starting with a random initial $\mathbf{v}_0 = \boldsymbol{\beta}_0$, the updated equations for Nesterov's Accelerated GD are as follows:

$$\begin{cases} \boldsymbol{\beta}^{(t+1)} & = \mathbf{v}^{(t)} - \alpha_t \cdot \nabla J(\mathbf{v}^{(t)}), \\ \mathbf{v}^{(t+1)} & = (1-\gamma_t) \cdot \boldsymbol{\beta}^{(t+1)} + \gamma_t \cdot \boldsymbol{\beta}^{(t)}, \end{cases} \quad (1)$$

where $0 < \gamma_t < 1$ is a moving average smoothing parameter.

Logistic Regression

Gradient Descent

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
-     % compute the gradient g
-     $\mathbf{W} = \mathbf{W} - \alpha \cdot g$

Nesterov's accelerated gradient method

Starting with a random initial $\mathbf{v}_0 = \boldsymbol{\beta}_0$, the updated equations for Nesterov's Accelerated GD are as follows:

$$\begin{cases} \boldsymbol{\beta}^{(t+1)} & = \mathbf{v}^{(t)} - \alpha_t \cdot \nabla J(\mathbf{v}^{(t)}), \\ \Downarrow & \\ \mathbf{v}^{(t+1)} & = (1-\gamma_t) \cdot \boldsymbol{\beta}^{(t+1)} + \gamma_t \cdot \boldsymbol{\beta}^{(t)}, \end{cases} \tag{1}$$

where $0 < \gamma_t < 1$ is a moving average smoothing parameter.

Logistic Regression

Gradient Descent

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
- % compute the gradient g　　初始化数据至区间[-1,+1] 使得 Sigmoid函数的输入值较小
- $\mathbf{W} = \mathbf{W} - \alpha \cdot g$

Nesterov's accelerated gradient method

Starting with a random initial $\mathbf{v}_0 = \boldsymbol{\beta}_0$, the updated equations for Nesterov's Accelerated GD are as follows:

$$\begin{cases} \boldsymbol{\beta}^{(t+1)} & = \mathbf{v}^{(t)} - \alpha_t \cdot \nabla J(\mathbf{v}^{(t)}), \\ \mathbf{v}^{(t+1)} & = (1 - \gamma_t) \cdot \boldsymbol{\beta}^{(t+1)} + \gamma_t \cdot \boldsymbol{\beta}^{(t)}, \end{cases} \tag{1}$$

where $0 < \gamma_t < 1$ is a moving average smoothing parameter.

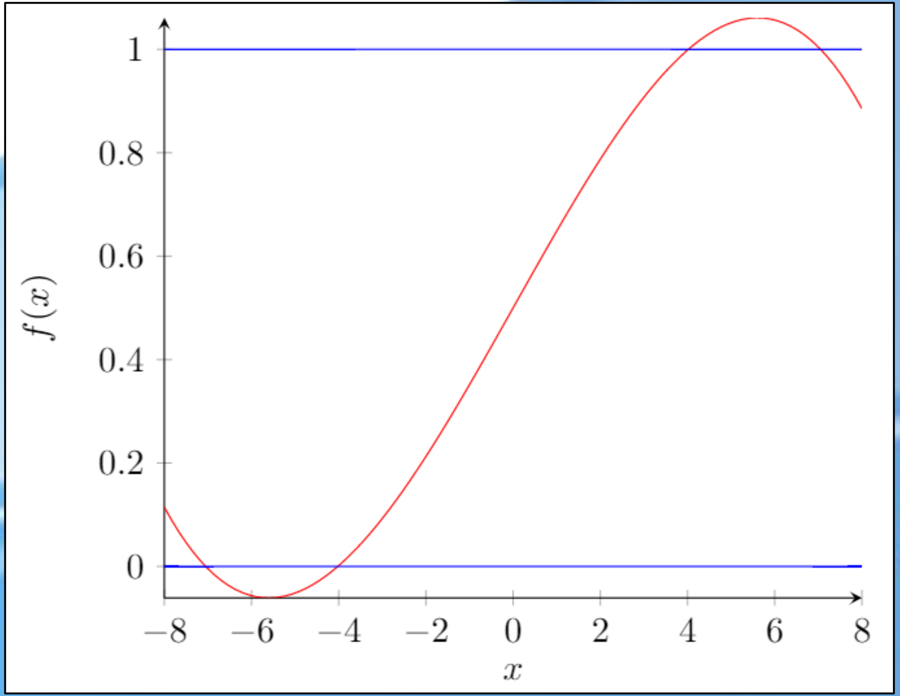
Baidu ...ratory ...ibuted ...gy Lab

Paper Technique

Gradient Descent

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
- % compute the gradient g
- $\mathbf{W} = \mathbf{W} - \boldsymbol{\alpha} \cdot \boldsymbol{g}$

初始化数据至区间[-1,+1] 使得 Sigmoid函数的输入值较小



✓应该必须初始化数据

需要使用多项式拟合Sigmoid函数

$Sigmoid(y\boldsymbol{W}^T\boldsymbol{X})$ 使得$y\boldsymbol{W}^T\boldsymbol{X}$落在很小的区间内

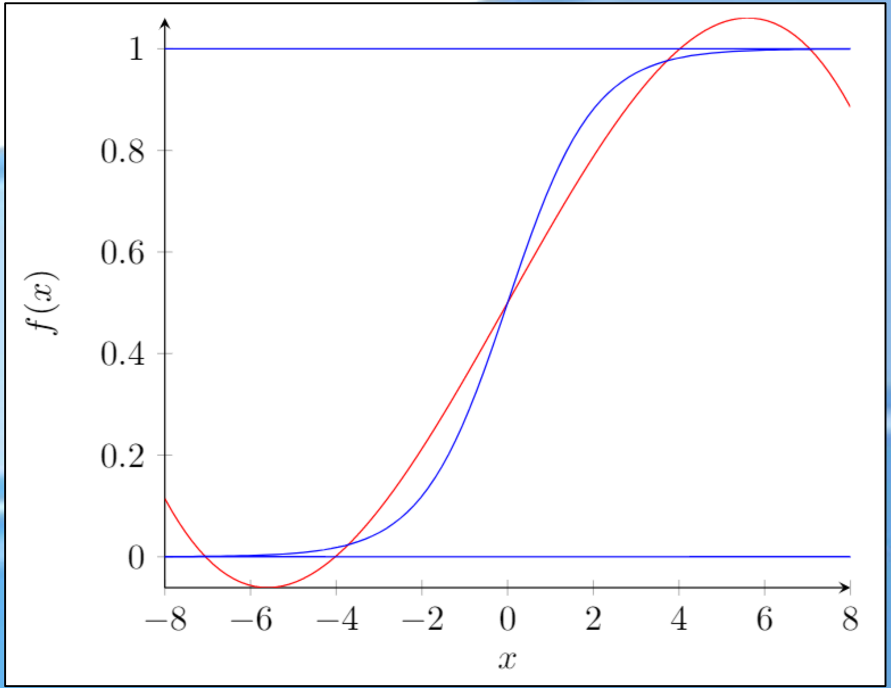
{0.5 + 1.20096/8*x - 0.81562/8^3*x^3};

Paper Technique

Gradient Descent

- **初始化权重向量W**
- **for i = 1 ：MAX_ITER do**
- **% compute the gradient g**
- **$\mathbf{W} = \mathbf{W} - \boldsymbol{\alpha} \cdot \boldsymbol{g}$**

初始化数据至区间[-1,+1] 使得 Sigmoid函数的输入值较小



✓应该必须初始化数据

需要使用多项式拟合Sigmoid函数

Least Squares Approximation

{0.5 + 1.20096/8*x - 0.81562/8^3*x^3};

Paper Technique

Gradient Descent

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
- % compute the gradient g 初始化数据至区间[-1,+1] 使得 Sigmoid函数的输入值较小
- $\mathbf{W} = \mathbf{W} - \alpha \cdot g$

Fig. 2. Graphs of sigmoid function and Taylor polynomials

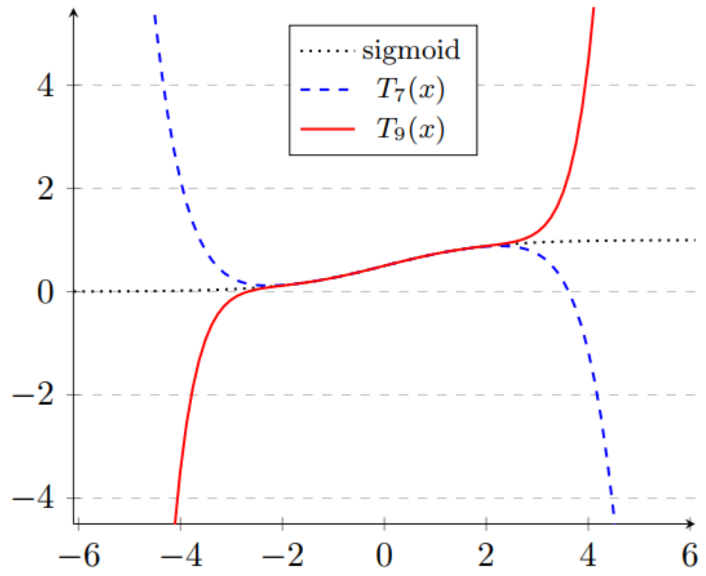
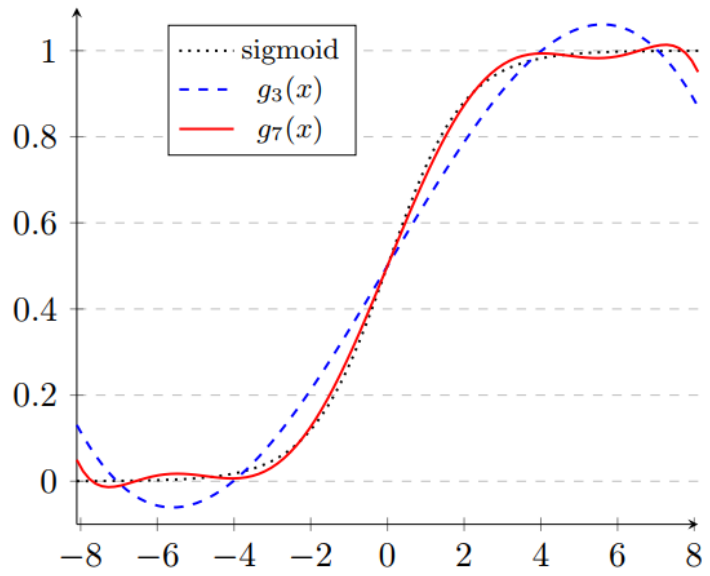


Fig. 3. Graphs of sigmoid function and least squares approximations



✓应该必须初始

需要使用多项式拟合S

Least Squares Approx

{0.5 + 1.20096/8*x - (

Paper Technique

Homomorphic Encryption Scheme: HEAAN

他们自己设计的同态加密库

同态加密的计算时间大 密文由系数为大整数的超长维数的向量构成

在应用到实际中，编码问题也会严重影响计算量

- The main idea is to treat an encryption noise as part of error occurring during approximate computations.
- We still have a problem that the bit size of message increases exponentially with the depth of a circuit without rounding.

We suggest a new technique – called *rescaling* – that manipulates the message of ciphertext.

Paper Technique

Homomorphic Encryption Scheme: HEAAN

- *rescaling* – Technically it seems similar to the modulus-switching method suggested by Brakerski and Vaikuntanatan.
- For an encryption $\boldsymbol{c}$ of m such that $\langle \boldsymbol{c}, sk \rangle = m + e \ (mod\ q)$, the rescaling procedure outputs a ciphertext $\lfloor p^{-1} \cdot \boldsymbol{c} \rceil \ (mod\ q/p)$, which is a valid encryption of m/p with noise about e/p. It reduces the size of ciphertext modulus and consequently removes the error located in the LSBs of messages, similar to the rouding step of fixed/floating-point arithmetic, while almost preserving the precision of plaintexts.

- For a plaintext modulus t and a ciphertext modulus q
- BGV $\langle \boldsymbol{c}, sk \rangle = m + te \quad (mod\ q)$
- BFV $\langle \boldsymbol{c}, sk \rangle = qI + (q/t)m + e \quad (mod\ q)$
- HEAAN $\langle \boldsymbol{c}, sk \rangle = m + e \quad (mod\ q)$

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Paper Technique

Homomorphic Encryption Scheme: HEAAN

- *rescaling* – Technically it seems similar to the modulus-switching method suggested by Brakerski and Vaikuntanatan.
- For an encryption $\boldsymbol{c}$ of m such that $\langle \boldsymbol{c}, sk \rangle = m + e \ (mod\ q)$, the rescaling procedure outputs a ciphertext $\lfloor p^{-1} \cdot \boldsymbol{c} \rceil \ (mod\ q/p)$, which is a valid encryption of m/p with noise about e/p. It reduces the size of ciphertext modulus and consequently removes the error located in the LSBs of messages, similar to the rouding step of fixed/floating-point arithmetic, while almost preserving the precision of plaintexts.
- $\text{double } a; \ \text{double } b; \ \xRightarrow{Encode} \ \text{int } A = 1000a; \ \text{int } B = 1000b$

  $A \times B = 1000000ab \xRightarrow{Rescaling} 1000ab \xRightarrow{Decode} ab$
- 否则，应该无法使用代替Sigmoid函数的近似多项式
  （因为多项式输出结果会超出[0,1]）

Paper Technique

Track 2: team evaluation (overall results)

Team	Submission	Schemes	End to End Performance: Running time (mins)	End to End Performance: Peak Memory (M)	Evaluation result (F1- Score) at different cutoffs: 0.01 Gold	0.01 Semi	0.001 Gold	0.001 Semi	0.0001 Gold	0.0001 Semi	0.00001 Gold	0.00001 Semi
A*FHE	A*FHE -1 +	HEAAN	922.48	3,777	0.977	0.999	0.986	0.999	0.985	0.999	0.966	0.998
	A*FHE -2		1,632.97	4,093	0.882	0.905	0.863	0.877	0.827	0.843	0.792	0.826
Chimera	Version 1 +	TFHE & HEAAN (Chimera)	201.73	10,375	0.979	0.993	0.987	0.991	0.988	0.989	0.982	0.974
	Version 2		215.95	15,166	0.339	0.35	0.305	0.309	0.271	0.276	0.239	0.253
Delft Blue	Delft Blue	HEAAN	1,844.82	10,814	0.965	0.969	0.956	0.944	0.951	0.935	0.884	0.849
UC San Diego	Logistic Regr +	HEAAN	1.66	14,901	0.983	0.993	0.993	0.987	0.991	0.989	0.995	0.967
	Linear Regr		0.42	3,387	0.982	0.989	0.980	0.971	0.982	0.968	0.925	0.89
Duality Inc	Logistic Regr +	CKKS (Aka HEAAN), pkg: PALISADE	3.8	10,230	0.982	0.993	0.991	0.993	0.993	0.991	0.990	0.973
	Chi2 test		0.09	1,512	0.968	0.983	0.981	0.985	0.980	0.985	0.939	0.962
Seoul National University	SNU-1	HEAAN	52.49	15,204	0.975	0.984	0.976	0.973	0.975	0.969	0.932	0.905
	SNU-2		52.37	15,177	0.976	0.988	0.979	0.975	0.974	0.969	0.939	0.909
IBM	IBM-Complex	CKKS (Aka HEAAN), pkg: HElib	23.35	8,651	0.913	0.911	0.169	0.188	0.067	0.077	0.053	0.06
	IBM- Real		52.65	15,613	0.542	0.526	0.279	0.28	0.241	0.255	0.218	0.229

+ no statistical significance in terms of discrimination, see following tables

https://yongsoosong.github.io/images/idash18_track2.jpg

Paper Technique

Track 2: team evaluation (overall results)

Team	Submission	Schemes	End to End Performance: Running time (mins)	End to End Performance: Peak Memory (M)	0.01 Gold	0.01 Semi	0.001 Gold	0.001 Semi	0.0001 Gold	0.0001 Semi	0.00001 Gold	0.00001 Semi
					Evaluation result (F1- Score) at different cutoffs							
A*FHE	A*FHE -1 +	HEAAN	922.48	3,777	0.977	0.999	0.986	0.999	0.985	0.999	0.966	0.998
	A*FHE -2		1,632.97	4,093	0.882	0.905	0.863	0.877	0.827	0.843	0.792	0.826
Chimera	Version 1 +	TFHE & HEAAN (Chimera)	201.73	10,375	0.979	0.993	0.987	0.991	0.988	0.989	0.982	0.974
	Version 2		215.95	15,166	0.339	0.35	0.305	0.309	0.271	0.276	0.239	0.253
Delft Blue	Delft Blue	HEAAN	1,844.82	10,814	0.965	0.969	0.956	0.944	0.951	0.935	0.884	0.849
UC San Diego	Logistic Regr +	HEAAN	1.66	14,901	0.983	0.993	0.993	0.987	0.991	0.989	0.995	0.967
	Linear Regr		0.42	3,387	0.982	0.989	0.980	0.971	0.982	0.968	0.925	0.89
Duality Inc	Logistic Regr +	CKKS (Aka HEAAN) pkg: PALISADE	3.8	10,230	0.982	0.993	0.991	0.993	0.993	0.991	0.990	0.973
	Chi2 test		0.09	1,512	0.968	0.983	0.981	0.985	0.980	0.985	0.939	0.962
Seoul National University	SNU-1	HEAAN	52.49	15,204	0.975	0.984	0.976	0.973	0.975	0.969	0.932	0.905
	SNU-2		52.37	15,177	0.976	0.988	0.979	0.975	0.974	0.969	0.939	0.909
IBM	IBM-Complex	CKKS (Aka HEAAN) pkg: HElib	23.35	8,651	0.913	0.911	0.169	0.188	0.067	0.077	0.053	0.06
	IBM- Real		52.65	15,613	0.542	0.526	0.279	0.28	0.241	0.255	0.218	0.229

+ no statistical significance in terms of discrimination, see following tables

https://yongsoosong.github.io/images/idash18_track2.jpg

Paper Technique

A New Encoding Method

HEAAN支持的操作

计算时间： 密文×密文 >> 明文×密文

Rotate： [a , b, ⋯ , z] >> [b , ⋯ , z , a]

SIMD： 密文1 = [a, b, ⋯ , z]

密文2 = [A, B, ⋯ , Z] 模运算

密文1 + 密文2 = [a + A, b + B, ⋯ , z + Z]

密文1 × 密文2 = [a × A, b × B, ⋯ , z × Z]

$$Z = \begin{bmatrix} z_{10} & z_{11} & \cdots & z_{1f} \\ z_{20} & z_{21} & \cdots & z_{1f} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n0} & z_{n1} & \cdots & z_{nf} \end{bmatrix}$$

$$\Downarrow$$

$$Z \mapsto \mathbf{w} = (z_{10}, \ldots, z_{1f}, z_{20}, \ldots, z_{2f}, \ldots, z_{n0}, \ldots, z_{nf})$$

A more efficient encoding method to encrypt a matrix

A training dataset consists of n samples $z_i \in \mathbb{R}^{f+1}$

Paper Technique

A New Encoding Method

SIMD　　密文1 = [a,　b,　··· ,　z]　　密文2 = [A,　B,　··· ,　Z]　　模运算

密文1 × 密文2 = [a × A,　b × B,　··· ,　z × Z]

$$\mathbf{ct}_z = \mathbf{Enc}\begin{bmatrix} z_{10} & z_{11} & \cdots & z_{1f} \\ z_{20} & z_{21} & \cdots & z_{1f} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n0} & z_{n1} & \cdots & z_{nf} \end{bmatrix}, \qquad \mathbf{ct}_\beta^{(0)} = \mathbf{Enc}\begin{bmatrix} \beta_0^{(0)} & \beta_1^{(0)} & \ldots & \beta_f^{(0)} \\ \beta_0^{(0)} & \beta_1^{(0)} & \ldots & \beta_f^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{(0)} & \beta_1^{(0)} & \ldots & \beta_f^{(0)} \end{bmatrix}$$

$$\boldsymbol{z}_i^T \boldsymbol{\beta}^{(t)} = \left[\boldsymbol{z}_{i0}, \boldsymbol{z}_{i1}, \cdots, \boldsymbol{z}_{if}\right] \cdot \left[\boldsymbol{\beta}_{i0}, \boldsymbol{\beta}_{i1}, \cdots, \boldsymbol{\beta}_{if}\right]^T$$

Paper Technique

A New Encoding Method

SIMD 密文1 = [a, b, ⋯ , z] 密文2 = [A, B, ⋯ , Z] 模运算

密文1 × 密文2 = [a × A, b × B, ⋯ , z × Z]

$$ct_z \times ct_{\beta}^{(0)} = ct_1 \; (SIMD\ Multiply\ and\ Rescale\ it)$$

$$\mathrm{ct}_1 = \mathrm{Enc}\begin{bmatrix} z_{10}\cdot\beta_0^{(t)} & z_{11}\cdot\beta_1^{(t)} & \cdots & z_{1f}\cdot\beta_f^{(t)} \\ z_{20}\cdot\beta_0^{(t)} & z_{21}\cdot\beta_1^{(t)} & \cdots & z_{1f}\cdot\beta_f^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n0}\cdot\beta_0^{(t)} & z_{n1}\cdot\beta_1^{(t)} & \cdots & z_{nf}\cdot\beta_f^{(t)} \end{bmatrix}$$

$$z_i^T\beta^{(t)} = \left[z_{i0}, z_{i1}, \cdots, z_{if}\right]\cdot\left[\beta_{i0}, \beta_{i1}, \cdots, \beta_{if}\right]^T$$

Paper Technique

A New Encoding Method

SIMD　　密文1 = [a, 　b, 　⋯ , 　z]　　密文2 = [A, 　B, 　⋯ , 　Z]　　模运算

密文1 × 密文2 = [a × A, 　b × B, 　⋯ , 　z × Z]

$$\boldsymbol{ct_1 \leftarrow Add\left(ct_1, Rotate\left(ct_1;\ 2^j\right)\right) \quad for\ j = 0, 1, \dots, log(f+1) - 1}$$

$$\mathbf{ct}_1 = \mathbf{Enc}\begin{bmatrix} z_{10}\cdot\beta_0^{(t)} & z_{11}\cdot\beta_1^{(t)} & \cdots & z_{1f}\cdot\beta_f^{(t)} \\ z_{20}\cdot\beta_0^{(t)} & z_{21}\cdot\beta_1^{(t)} & \cdots & z_{1f}\cdot\beta_f^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n0}\cdot\beta_0^{(t)} & z_{n1}\cdot\beta_1^{(t)} & \cdots & z_{nf}\cdot\beta_f^{(t)} \end{bmatrix}$$

$$\boldsymbol{z_i^T\beta^{(t)} = \left[z_{i0}, z_{i1}, \cdots, z_{if}\right]\cdot\left[\beta_{i0}, \beta_{i1}, \cdots, \beta_{if}\right]^T}$$

Nankai-Baidu Joint Laboratory

Parallel and Distributed Software Technology Lab

Paper Technique

A New Encoding Method

SIMD　　密文1 = [a,　b,　⋯ ,　z]　　密文2 = [A,　B,　⋯ ,　Z]　　模运算

密文1 × 密文2 = [a × A,　b × B,　⋯ ,　z × Z]

$$ct_1 \leftarrow Add\left(ct_1, Rotate(ct_1; 2^j)\right) \quad for\ j = 0, 1, \dots, log(f+1) - 1$$

$$\mathrm{ct}_2 = \mathrm{Enc}\begin{bmatrix} \mathbf{z}_1^T\beta^{(t)} & \star & \cdots & \star \\ \mathbf{z}_2^T\beta^{(t)} & \star & \cdots & \star \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_n^T\beta^{(t)} & \star & \cdots & \star \end{bmatrix}$$

$$z_i^T\beta^{(t)} = [z_{i0}, z_{i1}, \cdots, z_{if}] \cdot [\beta_{i0}, \beta_{i1}, \cdots, \beta_{if}]^T$$

Paper Technique

A New Encoding Method

SIMD 密文1 = [a, b, ⋯ , z] 密文2 = [A, B, ⋯ , Z] 模运算

密文1 × 密文2 = [a × A, b × B, ⋯ , z × Z]

This step performs a constant multiplication in order to annihilate the garbage values.

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}$$

$$\mathrm{ct}_3 = \mathbf{Enc}\begin{bmatrix} \mathbf{z}_1^T\beta^{(t)} & 0 & \cdots & 0 \\ \mathbf{z}_2^T\beta^{(t)} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_n^T\beta^{(t)} & 0 & \cdots & 0 \end{bmatrix}$$

$$\boldsymbol{z}_i^T\boldsymbol{\beta}^{(t)} = [\boldsymbol{z}_{i0}, \boldsymbol{z}_{i1}, \cdots, \boldsymbol{z}_{if}] \cdot [\boldsymbol{\beta}_{i0}, \boldsymbol{\beta}_{i1}, \cdots, \boldsymbol{\beta}_{if}]^T$$

Paper Technique

A New Encoding Method

SIMD　　密文1 = [a,　b,　⋯ ,　z]　　密文2 = [A,　B,　⋯ ,　Z]　　模运算

密文1 × 密文2 = [a × A,　b × B,　⋯ ,　z × Z]

$$\boldsymbol{ct_3 \leftarrow Add\left(ct_3, Rotate\left(ct_3; -2^j\right)\right)\ for\ j = 0, 1, \ldots, log(f+1) - 1}$$

$$\mathrm{ct}_4 = \mathrm{Enc}\begin{bmatrix} \mathbf{z}_1^T\boldsymbol{\beta}^{(t)} & \mathbf{z}_1^T\boldsymbol{\beta}^{(t)} & \cdots & \mathbf{z}_1^T\boldsymbol{\beta}^{(t)} \\ \mathbf{z}_2^T\boldsymbol{\beta}^{(t)} & \mathbf{z}_2^T\boldsymbol{\beta}^{(t)} & \cdots & \mathbf{z}_2^T\boldsymbol{\beta}^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{z}_n^T\boldsymbol{\beta}^{(t)} & \mathbf{z}_n^T\boldsymbol{\beta}^{(t)} & \cdots & \mathbf{z}_n^T\boldsymbol{\beta}^{(t)} \end{bmatrix}$$

Apply ct4 to the polynomial(sigmoid), and so on.　Some steps need Rescaling!

Recent Work

Fixed Hessian Method

- 初始化权重向量W
- for i = 1 ：MAX_ITER do
- % compute the gradient g and the Hessian matrix H
- $\mathbf{W} = \mathbf{W} - \boldsymbol{H}^{-1} \cdot \boldsymbol{g}$

Böhning et al. $\bar{\boldsymbol{H}} = -\frac{1}{4}\boldsymbol{X}^T\boldsymbol{X}$

Recent Work

Fixed Hessian Method

- 初始化权重向量W
- for i = 1 : MAX_ITER do
-     % compute the gradient g and the Hessian matrix H
-     $\mathbf{W} = \mathbf{W} - \boldsymbol{H}^{-1} \cdot \boldsymbol{g}$

Bonte et al. $\boldsymbol{B} = \mathbf{diag}()$

Gerschgorin’s circle theorem

Recent Work

Gerschgorin's circle theorem

2 Gershgorin's Theorem

Theorem 2.1 *(Gershgorin's Theorem Round 1)*

Every eigenvalue of matrix A_{nn} satisfies:

$$|\lambda - A_{ii}| \leq \sum_{j \neq i} |A_{ij}| \qquad i \in \{1, 2, ..., n\}$$

Recent Work

$|\lambda - 10| \leq |-1| + |0| + |1|$
$|\lambda - 10| \leq |0.2| + |1| + |-1|$

>>
$|\lambda - 10| \leq 2$
$|\lambda - 10| \leq 2.2$

>>
$D(10, 2)$

>>
$8 \leq \lambda \leq 12$

Example [edit]

Use the Gershgorin circle theorem to estimate the eigenvalues of:

$$A = \begin{bmatrix} 10 & -1 & 0 & 1 \\ 0.2 & 8 & 0.2 & 0.2 \\ 1 & 1 & 2 & 1 \\ -1 & -1 & -1 & -11 \end{bmatrix}.$$

Starting with row one, we take the element on the diagonal, a_{ii} as the center for the disc.
and apply the formula:

$$\sum_{j \neq i} |a_{ij}| = R_i$$

to obtain the following four discs:

$$D(10, 2),\ D(8, 0.6),\ D(2, 3),\ \text{and } D(-11, 3).$$

Note that we can improve the accuracy of the last two discs by applying the formula to th
$D(2, 1.2)$ and $D(-11, 2.2)$.

The eigenvalues are 9.8218, 8.1478, 1.8995, -10.86