Differentially private secure distributed logistic regression

Xiaoqian Jiang

Biomedical Informatics

University of California San Diego

IPAM workshop: algorithmic challenges of protecting biomedical data

1/11/2018

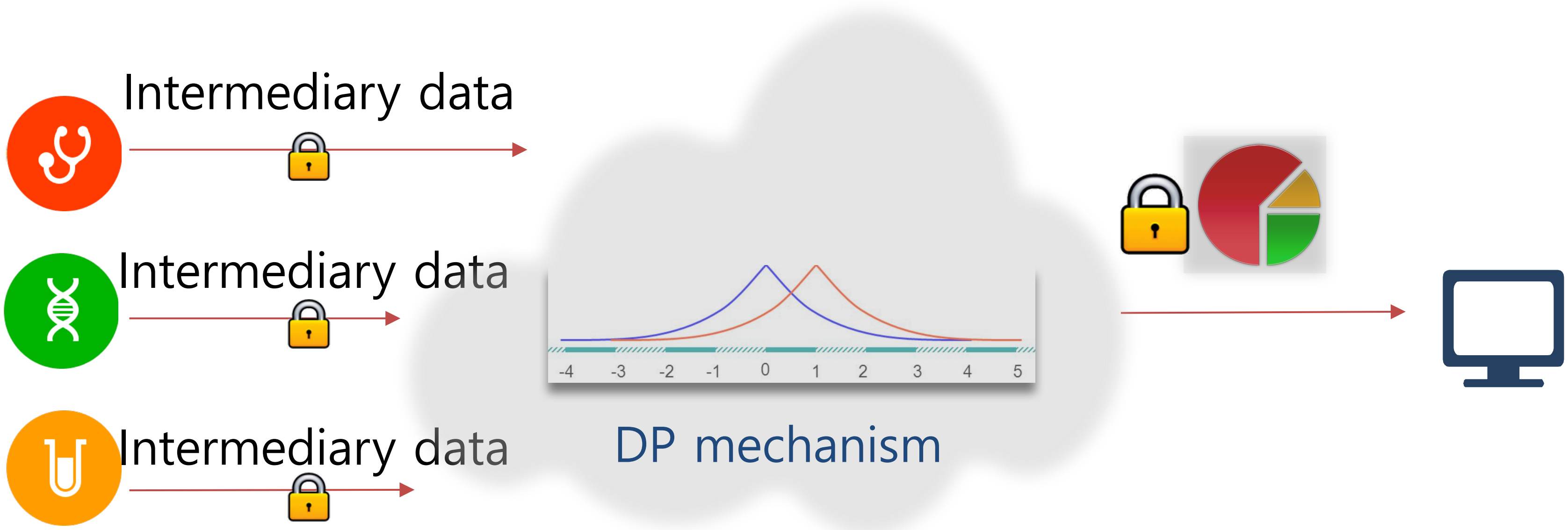
UC San Diego Division of Biomedical Informatics

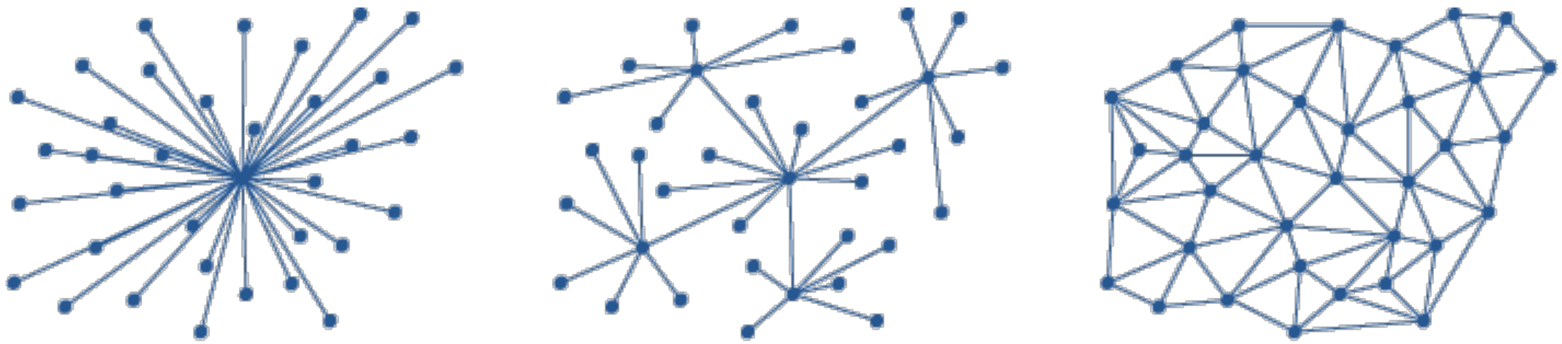
UC San Diego School of Medicine

Many attack models have been discovered…

- **Malin *2005***: Trails of hospital visit pattern might lead to information disclosure
- **Machanavajjhala *2007:*** Demographic statistics for certain cohort can lead to privacy lea kage.
- **Loukides *2010:*** Distribution of disease can lead to re-identification
- **Sweeney *2014:*** Demographics combined with phenotypes provide strong clues to reveal individuals' information
- **Bonomi *2017:*** Hospital visit frequency and interval can lead to re-identification

Homomorphic encryption and differential privacy might help

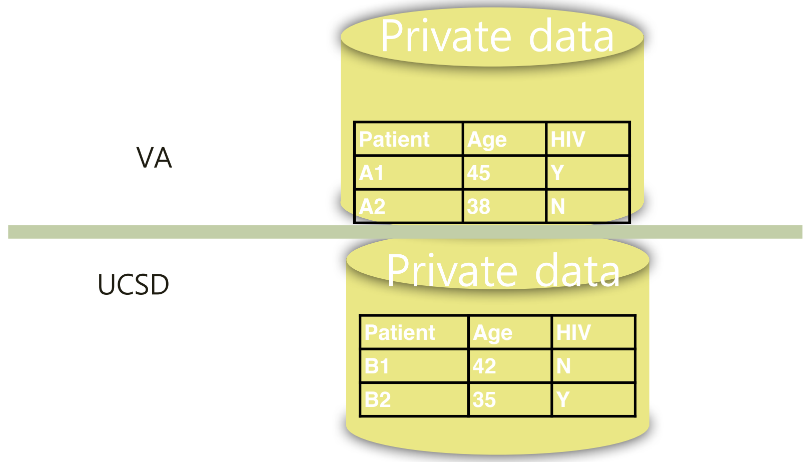




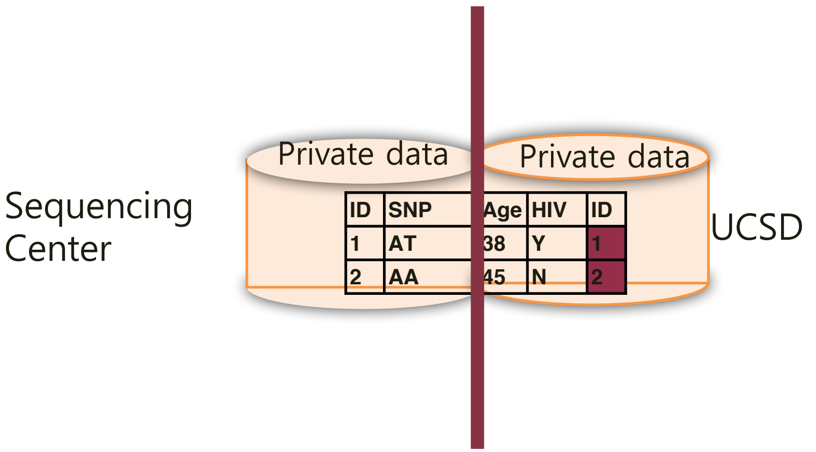
Privacy-Preserving Distributed Predictive Models

Two Representative Scenarios

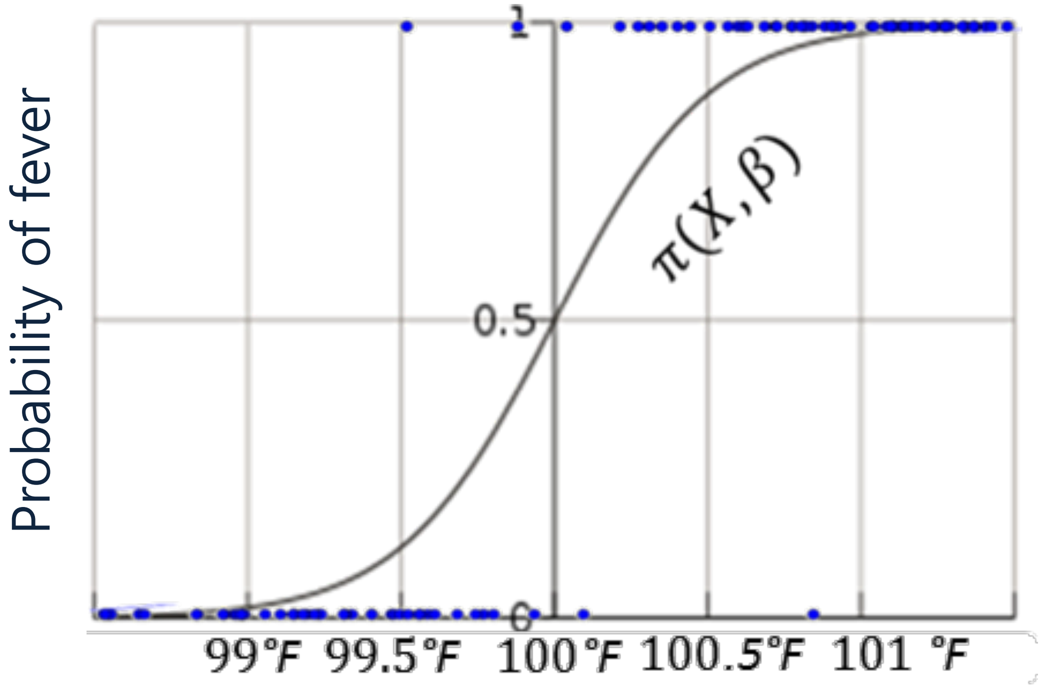
Horizontally distributed data



Vertically distributed data

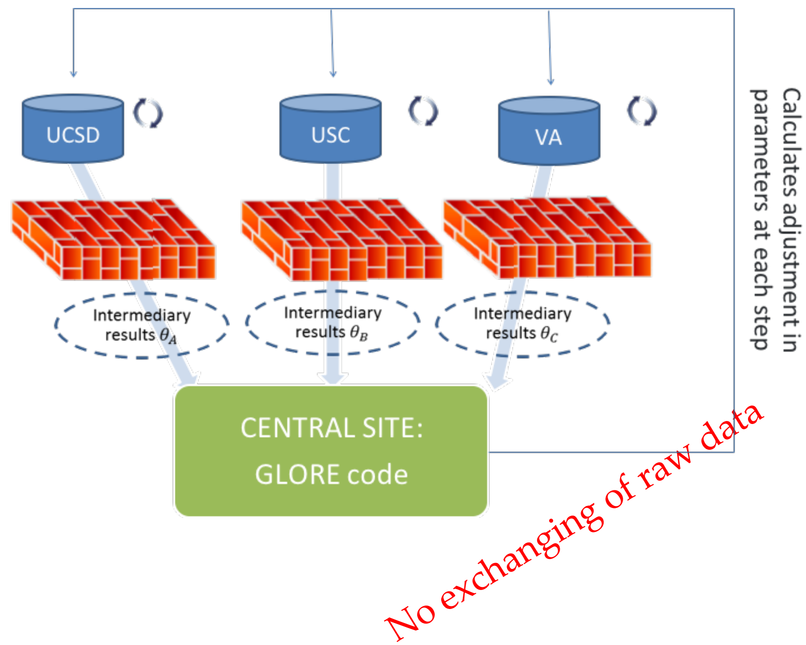


Logistic Regression



Learning a distributed logistic regression

- Support *p-1* features are c onsistent over *k* sites
- In each iteration, intermed iary result of a *pxp* matrix and a *p*-dimensional vecto r are transmitted to the ce ntral site for optimization



Maximum Likelihood Estimation

- **Estimated probability based on observations of a binary response Y and covariates X**
- **Likelihood function based on observed data (centralized)**

$$P(Y = 1|X) = \pi(X, \beta) = \frac{1}{1 + e^{-X\beta}}$$

Binary response

Covariates

Logit function

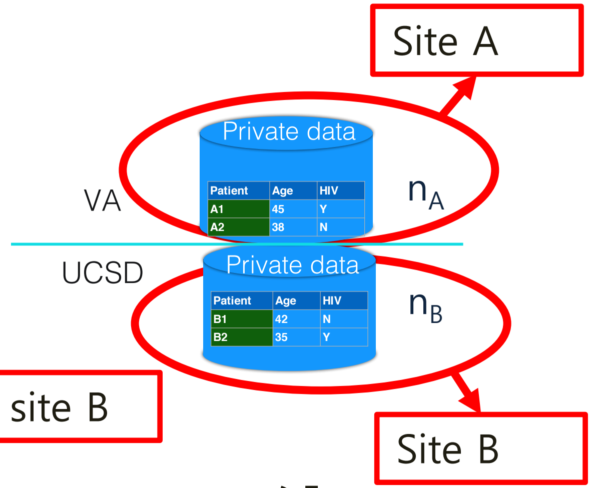
Model parameter

Number of records

$$l(\beta) = \sum_{1}^{n} \left[y_i \log \pi(x_i, \beta) + (1 - y_i) \log\left(1 - \pi(x_i, \beta)\right) \right]$$

Maximum Likelihood Estimation

- **Likelihood function based on observed data (distributed)**



$$P(Y=1|X)=\pi(X,\beta)=\frac{1}{1+e^{-X\beta}}$$

Number of records held by site A

Number of records held by site B

$$l(\beta)=\sum_{1}^{n_A+n_B}\left[y_i\log\pi(x_i,\beta)+(1-y_i)\log\left(1-\pi(x_i,\beta)\right)\right]$$

Maximum Likelihood Estimation

- **Newton-Raphson algorithm for calculation**

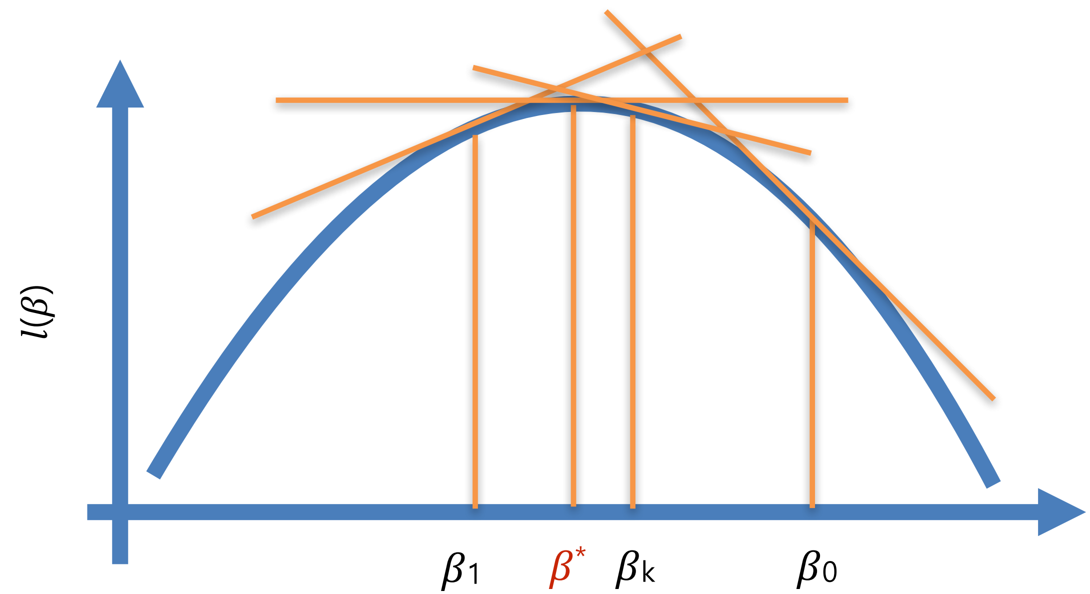
$$P(Y=1|X) = \pi(X,\beta) = \frac{1}{1+e^{-X\beta}}$$

$l(\beta)$ is a concave function

$$\Rightarrow \quad l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i,\beta) + (1-y_i) \log\big(1-\pi(x_i,\beta)\big) \right]$$

$$\beta^{(k+1)} = \beta^{(k)} - \left[\frac{\partial^2 l(\beta^{(k)})}{\partial \beta^{(k)} \partial {\beta^{(k)}}^T} \right]^{-1} \frac{\partial l(\beta^{(k)})}{\partial \beta^{(k)}}$$

Newton-Raphson (NR) Algorithm



$$l(\boldsymbol{\beta}) = \sum_{i=1}^{D} \{\boldsymbol{\beta}^T \sum_{l\in\mathcal{D}_i} \mathbf{z}^l - d_i \log[\sum_{l\in\mathcal{R}_i} \exp(\boldsymbol{\beta}^T \mathbf{z}^l)]\}$$

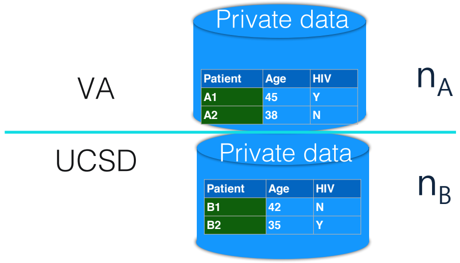
$$\beta^{(k+1)} = \beta^{(k)} - \left[\frac{\partial^2 l(\beta^{(k)})}{\partial\beta^{(k)}\partial\beta^{(k)^T}}\right]^{-1} \frac{\partial l(\beta^{(k)})}{\partial\beta^{(k)}}$$

Distributed Newton-Raphson (NR) Algorithm

$$l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i, \beta) + (1 - y_i) \log\left(1 - \pi(x_i, \beta)\right) \right]$$

Distributed Newton-Raphson (NR) Algorithm

$$l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i, \beta) + (1 - y_i) \log\left(1 - \pi(x_i, \beta)\right) \right]$$



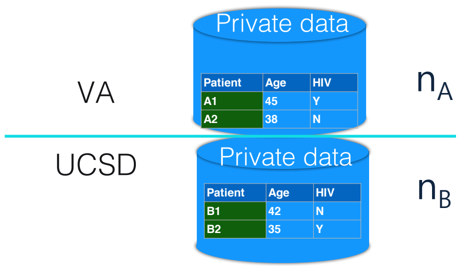
Distributed Newton-Raphson (NR) Algorithm

$$l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i,\beta) + (1-y_i)\log\left(1-\pi(x_i,\beta)\right)\right]$$

$$\begin{aligned}\beta^{(k+1)} &= \beta^{(k)} - \left[\frac{\partial^2 l(\beta^{(k)})}{\partial\beta^{(k)}\partial\beta^{(k)T}}\right]^{-1} \frac{\partial l(\beta^{(k)})}{\partial\beta^{(k)}} \\ &= \beta^{(k)} + \left[\bar{X}^T W(\bar{X},\beta^{(k)})\bar{X}\right]^{-1} \bar{X}^T\left[\bar{Y} - \Pi(\bar{X},\beta^{(k)})\right]\end{aligned}$$

Global variance-covariance matrix

Global prediction outcomes



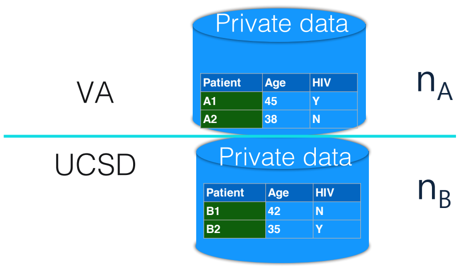
Distributed Newton-Raphson (NR) Algorithm

$$l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i,\beta) + (1-y_i)\log\big(1-\pi(x_i,\beta)\big)\right]$$

$$\beta^{(k+1)} = \beta^{(k)} - \left[\frac{\partial^2 l(\beta^{(k)})}{\partial\beta^{(k)}\partial\beta^{(k)^T}}\right]^{-1} \frac{\partial l(\beta^{(k)})}{\partial\beta^{(k)}}$$
$$= \beta^{(k)} + \left[\bar{X}^T W(\bar{X},\beta^{(k)})\bar{X}\right]^{-1} \bar{X}^T\left[\bar{Y} - \Pi(\bar{X},\beta^{(k)})\right]$$
$$= \beta^{(k)} + \left[\bar{X}_A^T W_A(\bar{X}_A,\beta^{(k)})\bar{X}_A + \bar{X}_B^T W_B(\bar{X}_B,\beta^{(k)})\bar{X}_B\right]^{-1}$$
$$\cdot \{\bar{X}_A^T[\bar{Y}_A - \Pi_A(\bar{X}_A,\beta)] + \bar{X}_B^T[\bar{Y}_B - \Pi_B(\bar{X}_B,\beta)]\}.$$

Local variance-covariance matrix

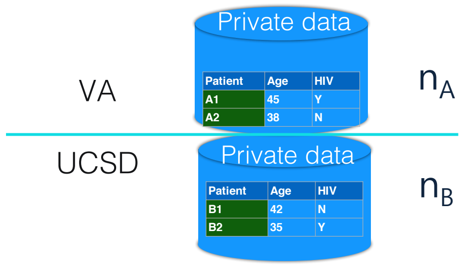
Local prediction outcomes



Distributed Newton-Raphson (NR) Algorithm

$$l(\beta) = \sum_{1}^{n_A+n_B} \left[y_i \log \pi(x_i,\beta) + (1-y_i)\log\left(1-\pi(x_i,\beta)\right) \right]$$

$$\begin{aligned}
\beta^{(k+1)} &= \beta^{(k)} - \left[\frac{\partial^2 l(\beta^{(k)})}{\partial\beta^{(k)}\partial\beta^{(k)T}}\right]^{-1} \frac{\partial l(\beta^{(k)})}{\partial \beta^{(k)}} \\
&= \beta^{(k)} + \left[\bar{X}^T W(\bar{X},\beta^{(k)})\bar{X}\right]^{-1} \bar{X}^T\left[\bar{Y} - \Pi(\bar{X},\beta^{(k)})\right] \\
&= \beta^{(k)} + \left[\bar{X}_A^T W_A(\bar{X}_A,\beta^{(k)})\bar{X}_A + \bar{X}_B^T W_B(\bar{X}_B,\beta^{(k)})\bar{X}_B\right]^{-1} \\
&\quad \cdot \{\bar{X}_A^T[\bar{Y}_A - \Pi_A(\bar{X}_A,\beta)] + \bar{X}_B^T[\bar{Y}_B - \Pi_B(\bar{X}_B,\beta)]\}.
\end{aligned}$$



Local variance-covariance matrix

$$W_A(\bar{X}_A,\beta) = \begin{bmatrix} \pi(x_1,\beta)(1-\pi(x_1,\beta)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \pi(x_{n_A},\beta)(1-\pi(x_{n_A},\beta)) \end{bmatrix},$$

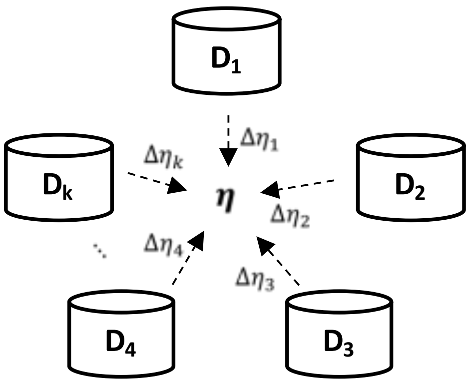
$$W_B(\bar{X}_B,\beta) = \begin{bmatrix} \pi(x_{n_A+1},\beta)(1-\pi(x_{n_A+1},\beta) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \pi(x_{n_A+n_B},\beta)(1-\pi(x_{n_A+n_B},\beta)) \end{bmatrix},$$

$$\Pi_A(\bar{X}_A,\beta) = \begin{bmatrix} \pi(x_1,\beta) \\ \vdots \\ \pi(x_{n_A},\beta) \end{bmatrix}, \text{ and } \Pi_B(\bar{X}_B,\beta) = \begin{bmatrix} \pi(x_{n_A+1},\beta) \\ \vdots \\ \pi(x_{n_A+n_B},\beta) \end{bmatrix}.$$

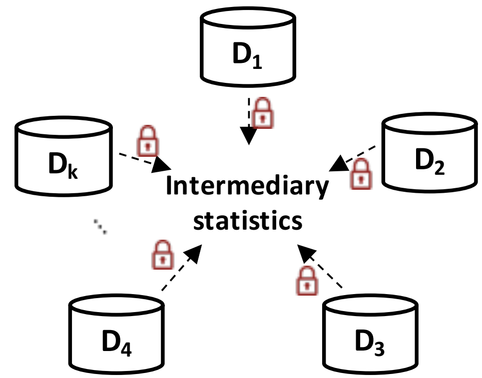
Local prediction outcomes

What remains to be solved?

- Masking the pattern before transmitting



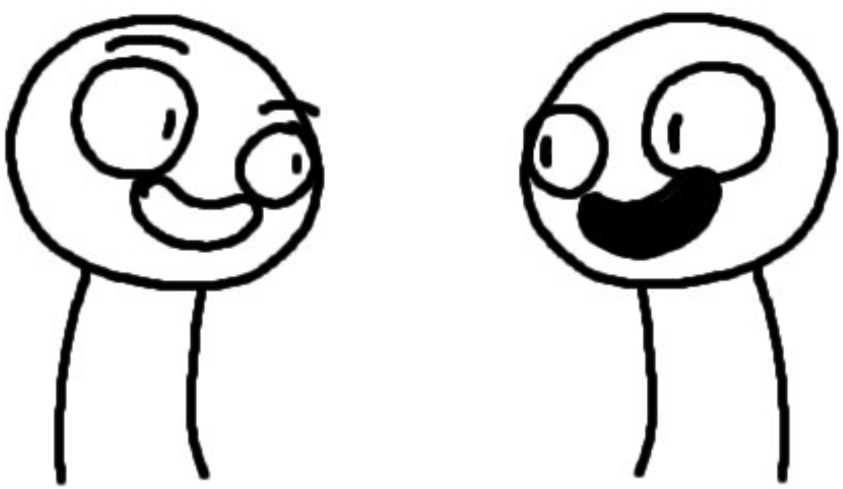
- Using secure primitive to safe guard the communication



Differential Privacy & homomorphic encryption

- A privacy mechanism *A* gives ε-differential privacy if for all neighbouring databases *D*, *D′*, and for any possible output $S \in \text{Range}(A)$, $Pr[A(D) = S] \leq \exp(\varepsilon) \times Pr[A(D') = S]$
  - D and D’ are neighboring databases if they differ on at most one record

- *Homomorphic encryption* is a type of encryption that allows computation conducted on ciphertext, when results are decrypted, map exactly to those of the corresponding computation on the plaintext



Differential private logistic regression

- **We perturb the objective function by adding an additional term $\frac{b^T\beta}{n}$ with b drawn from a Laplacian distribution with mean 0 and standard deviation $\frac{2}{\epsilon}$.**

$$\max_\beta \left[l(\beta) = -\sum_{i=1}^{n} \log(1 + \exp(-y_i \beta^T z_i) - \frac{\lambda}{2}\beta^T\beta - \frac{b^T\beta}{n} \right]$$

$$\beta^{new} = \beta^{old} - \left[l''\left(\beta^{old}\right)\right]^{-1} l'\left(\beta^{old}\right)$$

$$= \beta^{old} + \left(Z^T W^{old} Z + \lambda I\right)^{-1} \left[Z^T\left(Y - \mu^{old}\right) - \lambda\beta^{old} - \frac{b^T}{n}\right]$$

$$Z^T W^{old} Z = \sum_k Z_k^T W_k^{old} Z_k, \; Z^T[Y - \mu^{old}] = \sum_k Z_k^T \left[Y_k - \mu_k^{old}\right], \; b = \sum_k b_k,$$

$$k \in (1, \dots, K)$$

Chaudhuri K, Monteleoni C, Sarwate AD. Differentially Private Empirical Risk Minimization. J Mach Learn Res 2011 Mar;12(Mar):1069–1109. PMID:21892342

Differentially private logistic regression for distributed data

In a distributed setting, objective perturbation can be achieved by

- Gamma Distributed Perturbation Laplacian algorithm (DPLA)
- Gauss Distributed Perturbation Laplacian algorithm (DPLA)
- Laplace Distributed Perturbation Laplacian algorithm (DPLA)

Gergely Ács and Claude Castelluccia. I have a DREAM!: differentially private smart metering. In: Proceedings of the 3rd International Conference on Information Hiding. IH'11. 2011, pp. 118-132.
Goryczka S, Xiong L. A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy. IEEE Trans Dependable Secure Comput 2017 Sep;14(5):463–477. PMID:28919841

Differentially private logistic regression for distributed data

- **Note that the noise added by a single party is not sufficient to ensure DP!**
  - If we add too much noise, the final output will be less valuable.
  - If we add too little noise, it is not enough to protect the privacy.
- **Privacy mechanisms are not designed to provide security of computations.**
  - We need to protect the intermediary results, otherwise, privacy cannot be ensured in a global manner

Algorithm

Win-Win Strategy

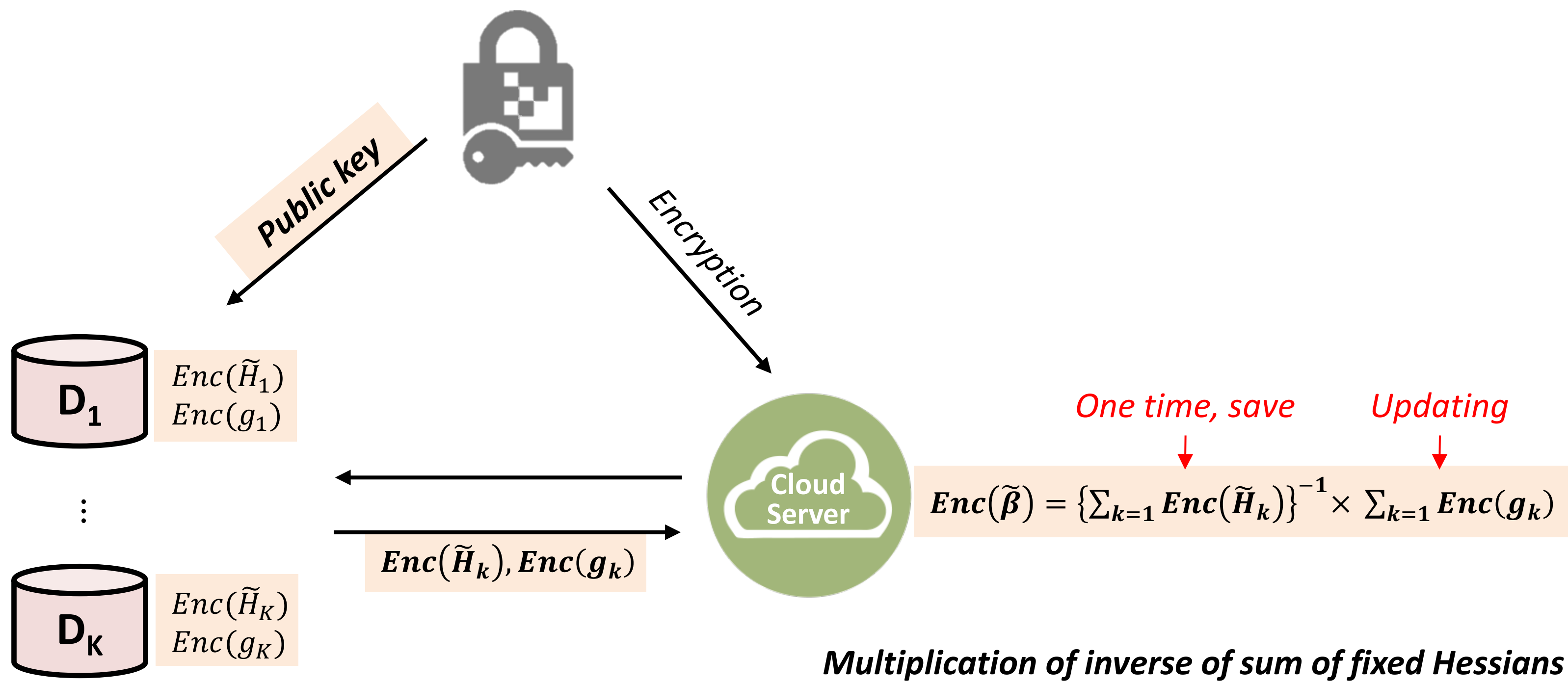
$$\beta^{new} = \beta^{old} - \left[l''\left(\beta^{old}\right)\right]^{-1} l'\left(\beta^{old}\right)$$

$$= \beta^{old} + \Big(\underbrace{\textstyle\sum_k Z_k^T W_k^{old} Z_k + \frac{\lambda}{K} I}_{Hessian\ =\ H}\Big)^{-1} \Big[\underbrace{\textstyle\sum_k \left(Z_k^T\left[Y_k - \mu_k^{old}\right] - \frac{\lambda}{K}\beta^{old}\right) - \left(\sum_k \frac{1}{n} b_k\right)^T}_{Gradient\ =\ g}\Big]$$

- **Homomorphic Encryption with “*Fixed Hessian*”**
  - $\sum_k Z_k^T W_k^{old} Z_k + \frac{\lambda}{K} I = \sum_k H_k \approx \sum_k \frac{1}{4} Z_k^T Z_k + \frac{\lambda}{K} I = \sum_k \bar{H}_k \approx \sum_k diag(\bar{H}_k) = \sum_k \widetilde{H}_k$ $\quad \boldsymbol{Enc(\widetilde{H}_k)}$ One time
  - $Z^T[Y - \mu^{old}] - \lambda\beta^{old} - \frac{1}{n} b = \sum_k Z_k^T \left[Y_k - \mu_k^{old}\right] - \frac{\lambda}{K}\beta^{old} - \sum_k \frac{1}{n} b_k = \sum_k g_k$ $\quad \boldsymbol{Enc(g_k)}$ Iteratively
- **Differential Privacy**
  - β^{new} can be revealed to parties because of the noise
  - HE can be renewed every iteration

Based on DP, we can reduce time complexity and error accumulation of HE

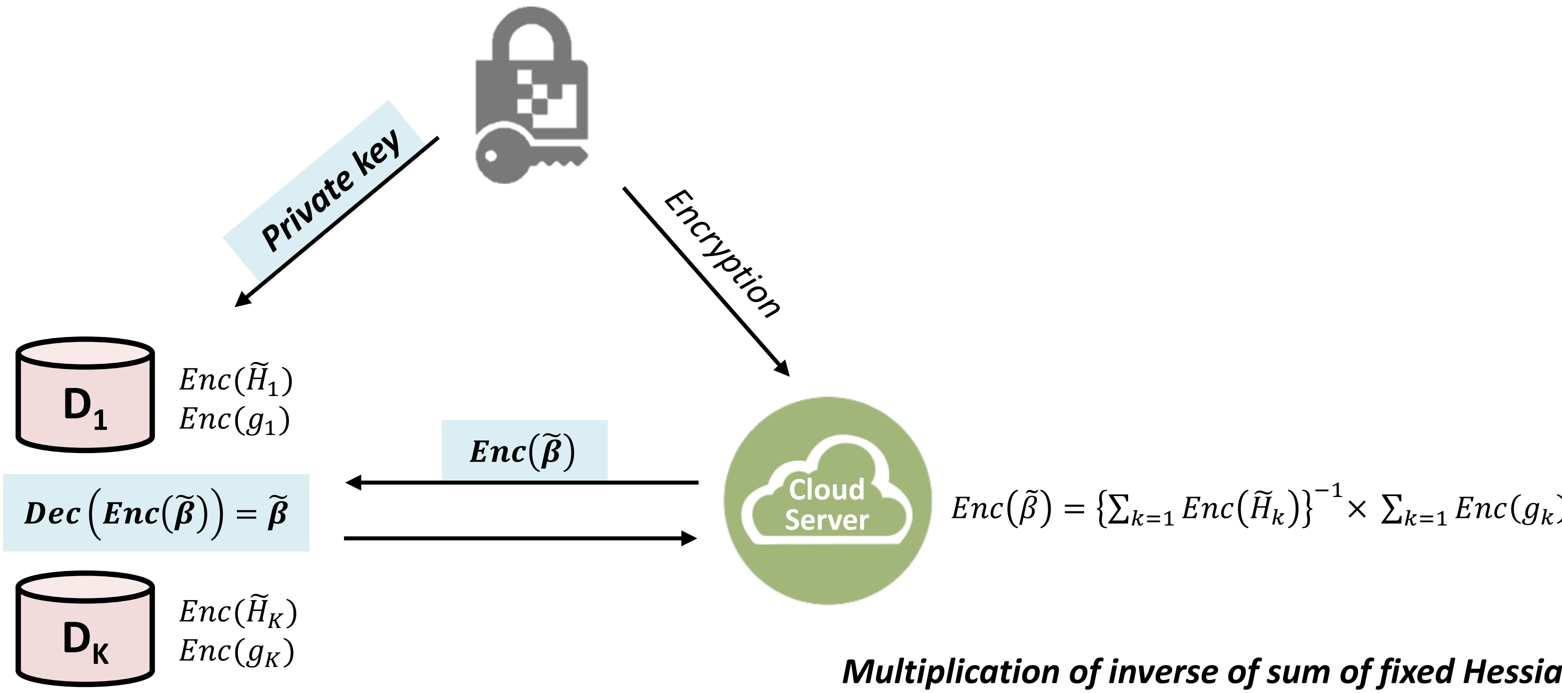
SMC Schemes with HE under Fixed Hessian



Multiplication of inverse of sum of fixed Hessians by sum of gradients

Approximation of fixed Hessian and gradient

SMC Schemes with HE under Fixed Hessian

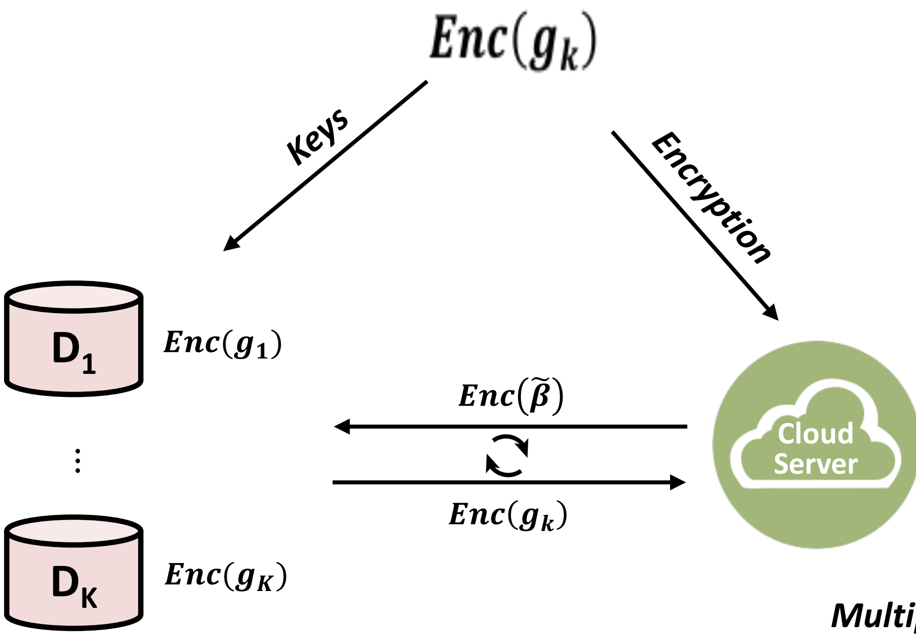


Multiplication of inverse of sum of fixed Hessians by sum of gradients

Approximation of fixed Hessian and gradient

$\widetilde{\beta}$ can be revealed to parties because of the noise.

SMC Schemes with HE under Fixed Hessian



$$Enc(\tilde{\beta}) = \{\underbrace{\textstyle\sum_{k=1} Enc(\tilde{H}_k)}_{\text{Fixed}}\}^{-1} \times \underbrace{\textstyle\sum_{k=1} Enc(g_k)}_{\text{Updating}}$$

Approximation of fixed Hessian and gradient

Multiplication of inverse of sum of fixed Hessians by sum of gradients

A few iterations → Converge

Limitations of Fixed Hessian

$$\sum_k Z_k^T W_k^{old} Z_k + \frac{\lambda}{K} I = \sum_k H_k \approx \sum_k \frac{1}{4} Z_k^T Z_k + \frac{\lambda}{K} I = \sum_k \bar{H}_k \approx \sum_k diag(\bar{H}_k) = \sum_k \widetilde{H}_k$$

- **Simple approximation of Hessian using only its diagonal elements**
- **Valid when the matrix strongly diagonally dominant**
- **Large enough λ to be set**

Largely dependent on λ

- **Better diagonal Hessian approximation**

Diagonal Updating via Quasi-Cauchy Relation

Diagonal Updating via Quasi-Cauchy Relation*

$$\nabla^2 f(x) = \nabla^2 f_A(x) + \nabla^2 f_B(x)$$

where $\nabla^2 f_A(x)$: a diagonal matrix consisting the diagonal entries of the Hessian

$\nabla^2 f_B(x)$: the actual Hessian except that its diagonal entries are all zero

$$\nabla^2 f(x) \approx D = \Psi_1 + \Psi_2 = \Psi_1 + (\theta I + \Psi_3)$$

where Ψ_1: a positive definite diagonal matrix

$$\min \ \frac{1}{2} \|\Psi_3\|_F^2,$$

s.t. $s_i^T\left(\Psi_1 + (\theta I + \Psi_3)\right)s_i = s_i^T y_i$ and Ψ_3 is diagonal

$$D_{i+1} = D_i + \frac{s_i^T y_i - s_i^T \Psi_1 s_i - \theta_i s_i^T s_i}{\operatorname{tr}\left(E_i^2\right)} E_i$$

where $\theta_i = \min\left[1, \frac{s_i^T y_i - s_i^T \Psi_1 s_i}{s_i^T s_i}\right]$ for positive definiteness and $E_i = \operatorname{diag}(s_{i,1}^2, s_{i,2}^2, \dots, s_{i,m}^2)$

* Marjugi and Leong (2013) Diagonal Hessian Approximation for Limited Memory Quasi-Newton via Variational Principle, *Journal of Applied Mathematics*

UC San Diego
School of Medicine

Diagonal Updating via Quasi-Cauchy Relation

Decomposable

$$D_{i+1} = D_i + \frac{s_i^T y_i - s_i^T \Psi_1 s_i - \theta_i s_i^T s_i}{\mathrm{tr}(E_i^2)} E_i, \quad \boxed{\frac{s_i^T y_i - s_i^T \Psi_1 s_i}{\mathrm{tr}(E_i^2)} E_i} - \theta_i \cdot \frac{s_i^T s_i}{\mathrm{tr}(E_i^2)} E_i = \sum_{k=1}^{K} V_{ik} - \theta_i \cdot W_i$$

where $s_i = \beta^{i+1} - \beta^i$

$$y_i = \sum_k \left(Z_k^T \left[Y_k - \mu_k^{i+1} \right] - \frac{\lambda}{K} \beta^{i+1} \right) - \sum_k \left(Z_k^T \left[Y_k - \mu_k^{i} \right] - \frac{\lambda}{K} \beta^{i} \right)$$

$$V_{ik} = \frac{s_{ik}^T y_i - s_i^T \Psi_1 s_i}{\mathrm{tr}(E_i^2)} E_i, \; W_i = \frac{s_i^T s_i}{\mathrm{tr}(E_i^2)} E_i$$

- For positive definiteness
  - $\theta_i = \min\left[1, \frac{s_i^T y_i - s_i^T \Psi_1 s_i}{s_i^T s_i}\right]$
  - Comparison within ciphertext is not easy, so we used one more round of iteraction

One more step is added every iteration.

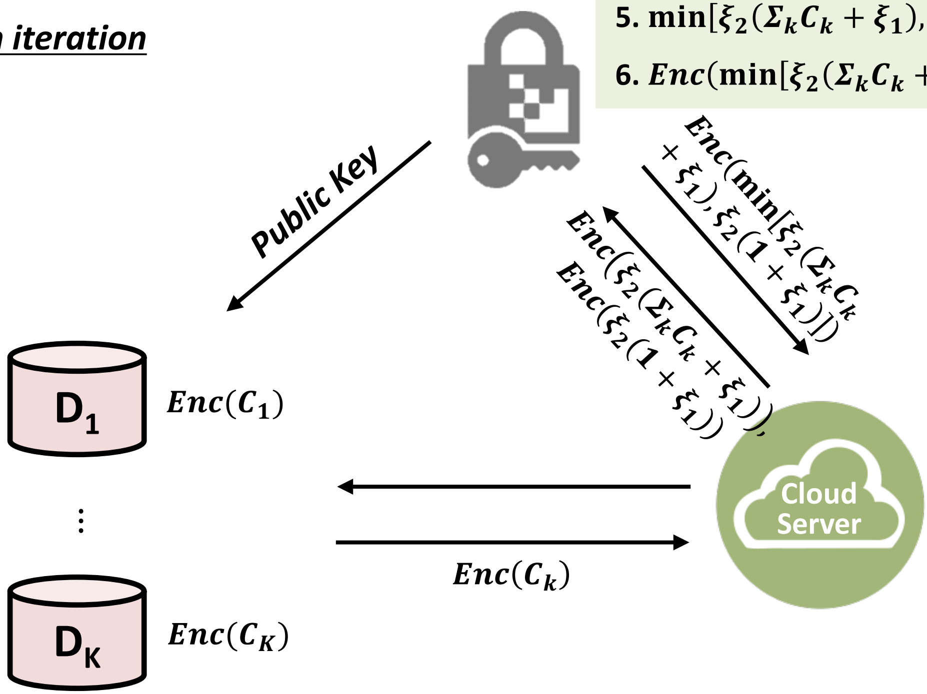
Positive Definiteness

Decomposable

$$\theta_i = \min\left[1, \boxed{\frac{s_i^T y_i - s_i^T \Psi_1 s_i}{s_i^T s_i}}\right] = \min[1, \Sigma_k C_{ik}]$$

where $C_{ik} = \dfrac{s_i^T y_{ik}}{s_i^T s_i} - \dfrac{s_i^T \Psi_1 s_i}{K s_i^T s_i}$

i-th iteration



1. $Enc(\Sigma_k C_k) = \Sigma_k Enc(C_k)$

2. Random number ξ_1, ξ_2 **generation**

3. $Enc(\xi_2(\Sigma_k C_k + \xi_1)), Enc(\xi_2(1+\xi_1))$

4. $Dec(\xi_2(\Sigma_k C_k + \xi_1)) = \xi_2(\Sigma_k C_k + \xi_1)$

$Dec(\xi_2(1+\xi_1)) = \xi_2(1+\xi_1)$

5. $\min[\xi_2(\Sigma_k C_k + \xi_1), \xi_2(1+\xi_1)]$

6. $Enc(\min[\xi_2(\Sigma_k C_k + \xi_1), \xi_2(1+\xi_1)])$

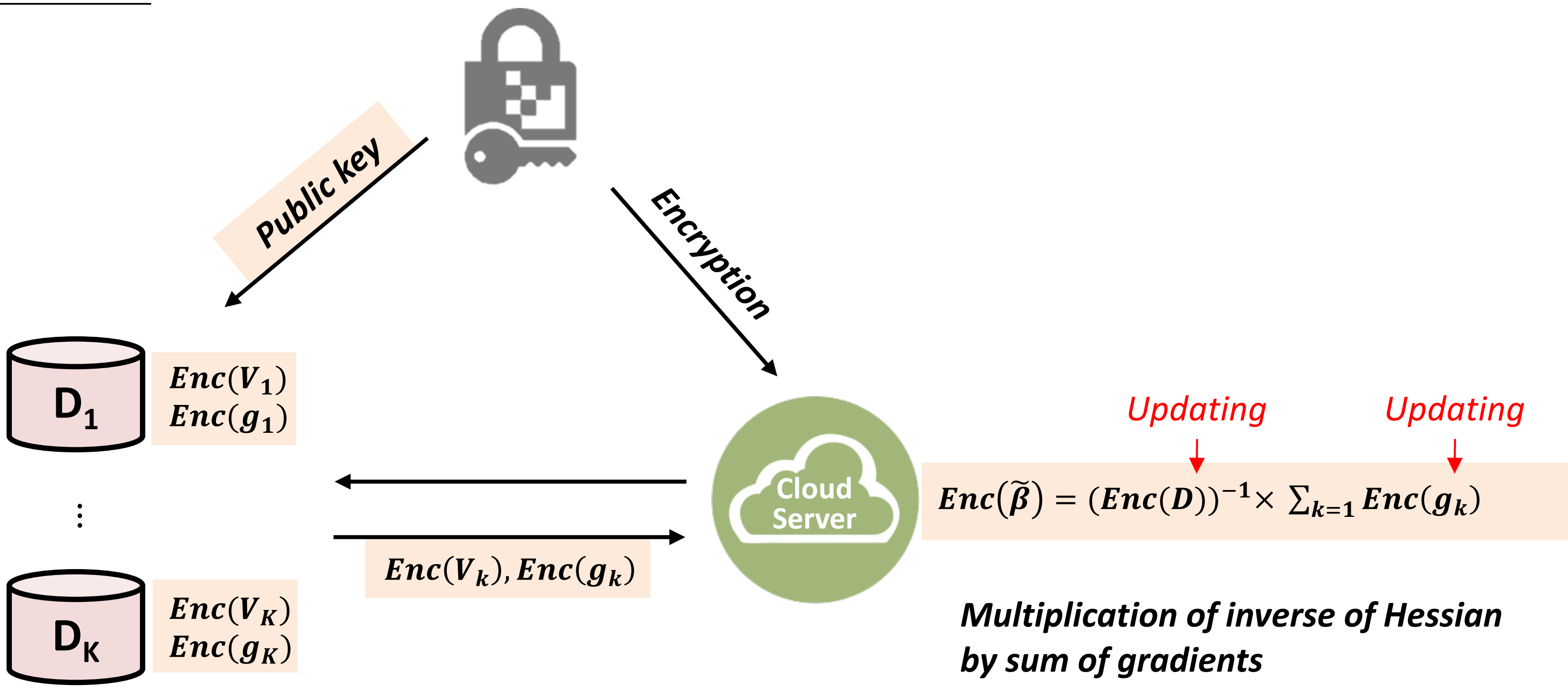
7. $Enc(\min[\Sigma_k C_k, 1])$

$= Enc(\min[\xi_2(\Sigma_k C_k - \xi_1), \xi_2(1+\xi_1)])$

$/Enc(\xi_2^{-1}) - Enc(\xi_1)$

SMC Schemes with HE under Updating Hessian

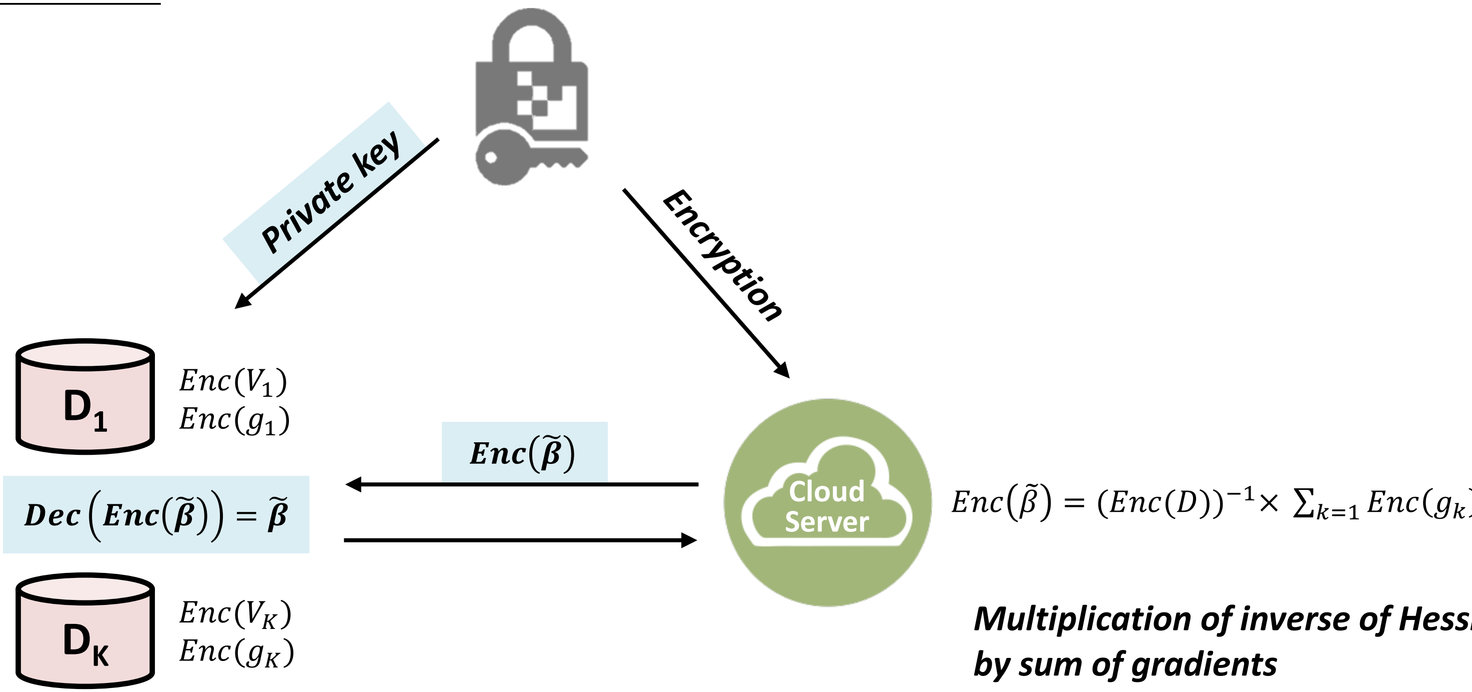
i-th iteration



Approximation of Hessian and gradient

SMC Schemes with HE under Updating Hessian

i-th iteration

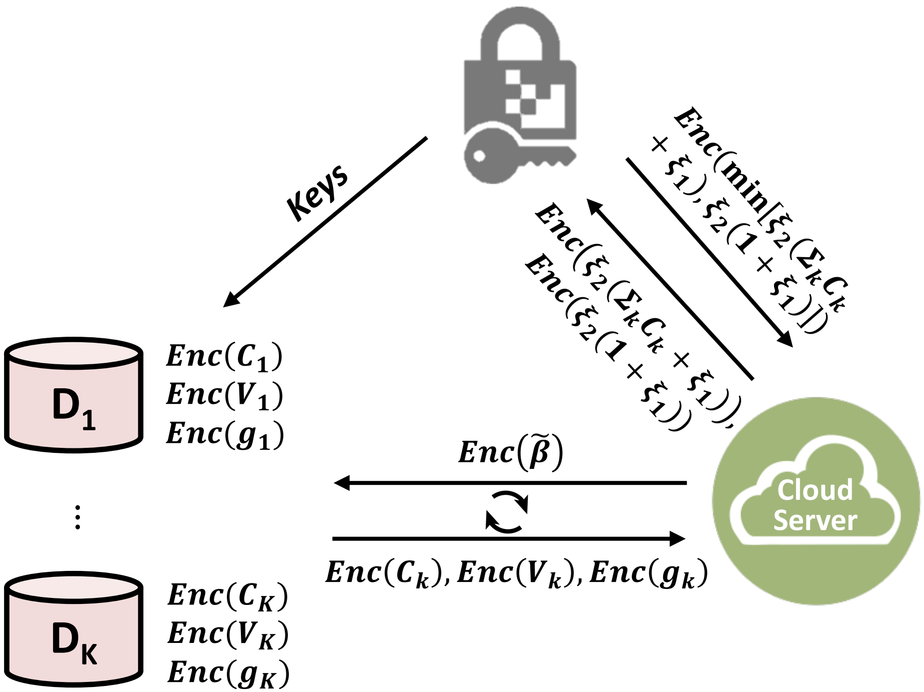


Multiplication of inverse of Hessian by sum of gradients

Approximation of Hessian and gradient

$\tilde{\beta}$ can be revealed to parties because of the noise.

SMC Schemes with HE under Updating Hessian



Updating ↓ Updating ↓

$$Enc(\tilde{\beta}) = (Enc(D))^{-1} \times \sum_{k=1} Enc(g_k)$$

Multiplication of inverse of Hessian by sum of gradients

Approximation of Hessian and gradient

A few iterations → Converge

Trade-Off between Fixed Hessian and Updating Hessian

	Fixed Hessian	Updating Hessian
The number of Iterations	$\uparrow$	$\downarrow$
Time per iteration	$\downarrow$	$\uparrow$

- **Fixed Hessian**
  - Iteration numbers are too dependent on λ which is also depending on data.
  - The number of iterations can be more than 100 when not big enough λ.
- **Updating Hessian**
  - Iteration numbers are quite robust on λ.
  - Inverse diagonal Hessian is not that much expensive ($\because$ vector computation).

Dataset: Death in hospital

- **PhysioNet Challenge 2012 [MIMIC II database]**
  - Dataset comprised of 4000 patient stays in the ICU lasting at least 2 days for predicting mortality.
  - The data were formatted as time-stamped measurements for 37 distinct variables.
  - Four static variables (age, gender, height, and initial weight) are also present.

➢ Number of patients: 4000, Number of features: 41

Data Preprocessing

Percentage of patients for whom at least one measurement was available during the first 48 ICU hours

Measurement	%	Measurement	%
ABP (Arterial blood pressure)		Heart rate	98.4
Invasive (diastolic, mean, systolic)	98.4	K (Serum potassium)	97.9
Non-invasive (diastolic)	87.3	Lactate	54.8
Non-invasive (mean)	87.2	Mg (Serum magnesium)	97.5
Non-invasive (systolic)	87.6	Mechanical ventilation	63.1
Albumin	40.5	Na (Serum sodium)	98.2
ALP (Alkaline phosphatase)	42.4	PaCO2	75.4
ALT (Alkaline transaminase)	43.4	PaO2	75.4
AST (Aspartate transaminase)	43.4	pH	75.9
Bilirubin	43.4	Platelets	98.3
BUN (Blood urea nitrogren)	98.4	Respiration rate	27.7
Cholesterol	7.9	SaO2	44.7
Creatinine	98.4	Temperature	98.4
FiO2 (Fractional inspired oxygen)	67.6	Troponin-I	4.7
Glasgow Coma Score (GCS)	98.4	Troponin-T	21.9
Glucose	97.5	Urine output	97.4
HCO3 (Serum bicarbonate)	98.2	WBC (White blood cell count)	98.2
HCT (Hematocrit)	98.4	Weight	67.7

1. Compute min, max, mean, first value, last value as a way to represent time-series features

2. Missing values are replaced by the mean value of a feature.

Number of patients: 4,000, Number of features: 189

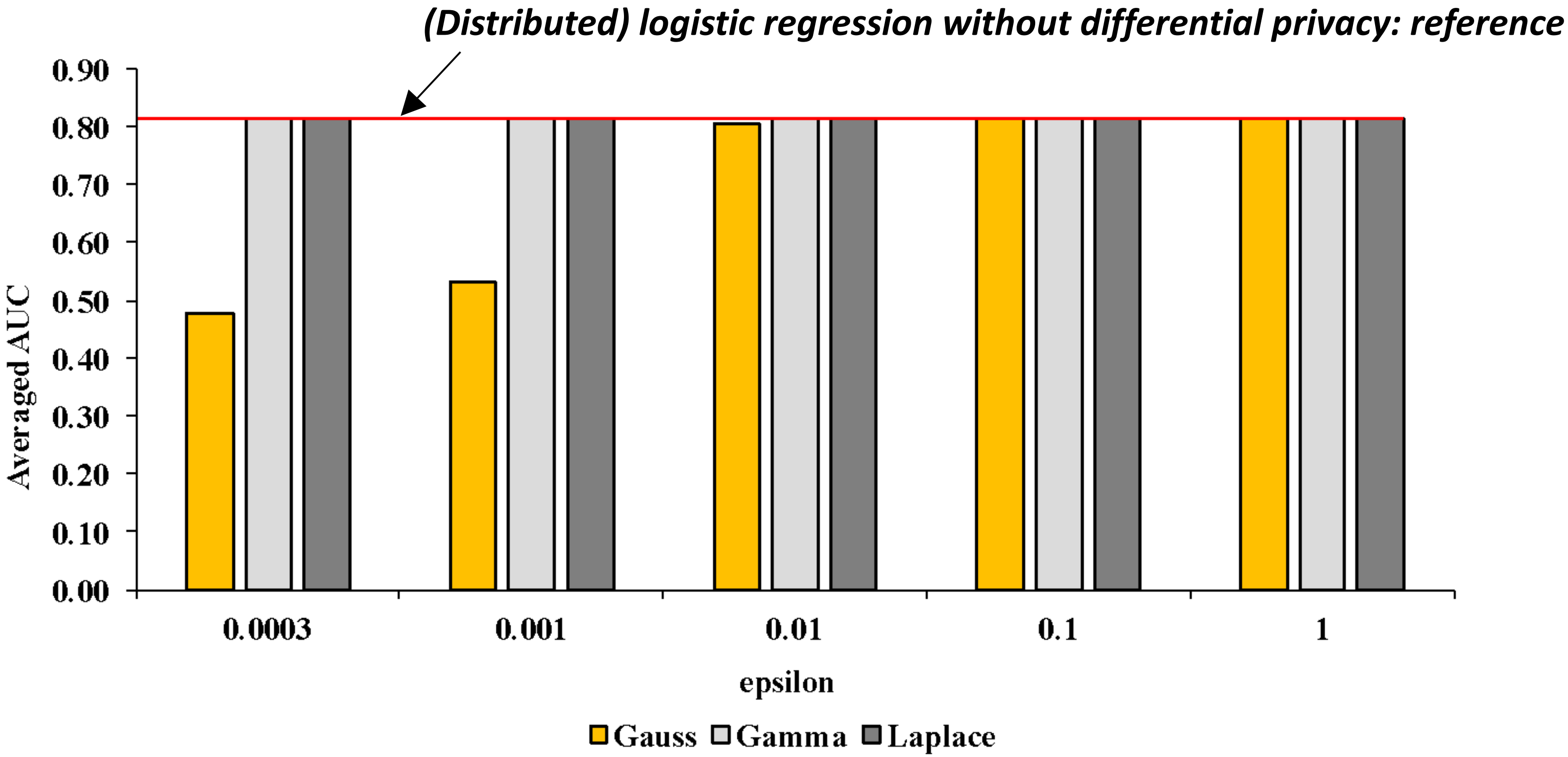
Experiment

- **Models**
  1) (Distributed) model without differential privacy

     For $b \sim Lap(0, \sqrt{2}/\epsilon)$ with $2/\epsilon$ standard deviation
  2) Model with Gamma DLPA and HE
  3) Model with Gauss DLPA and HE
  4) Model with Laplace DLPA and HE
- **Scenario**
  - 3 sites with equal sizes
- **Comparison**
  - 10 repetitions of 4-fold CV
    - AUC
    - Mean of coefficients
    - Standard deviation of coefficients

Reference Result without HE

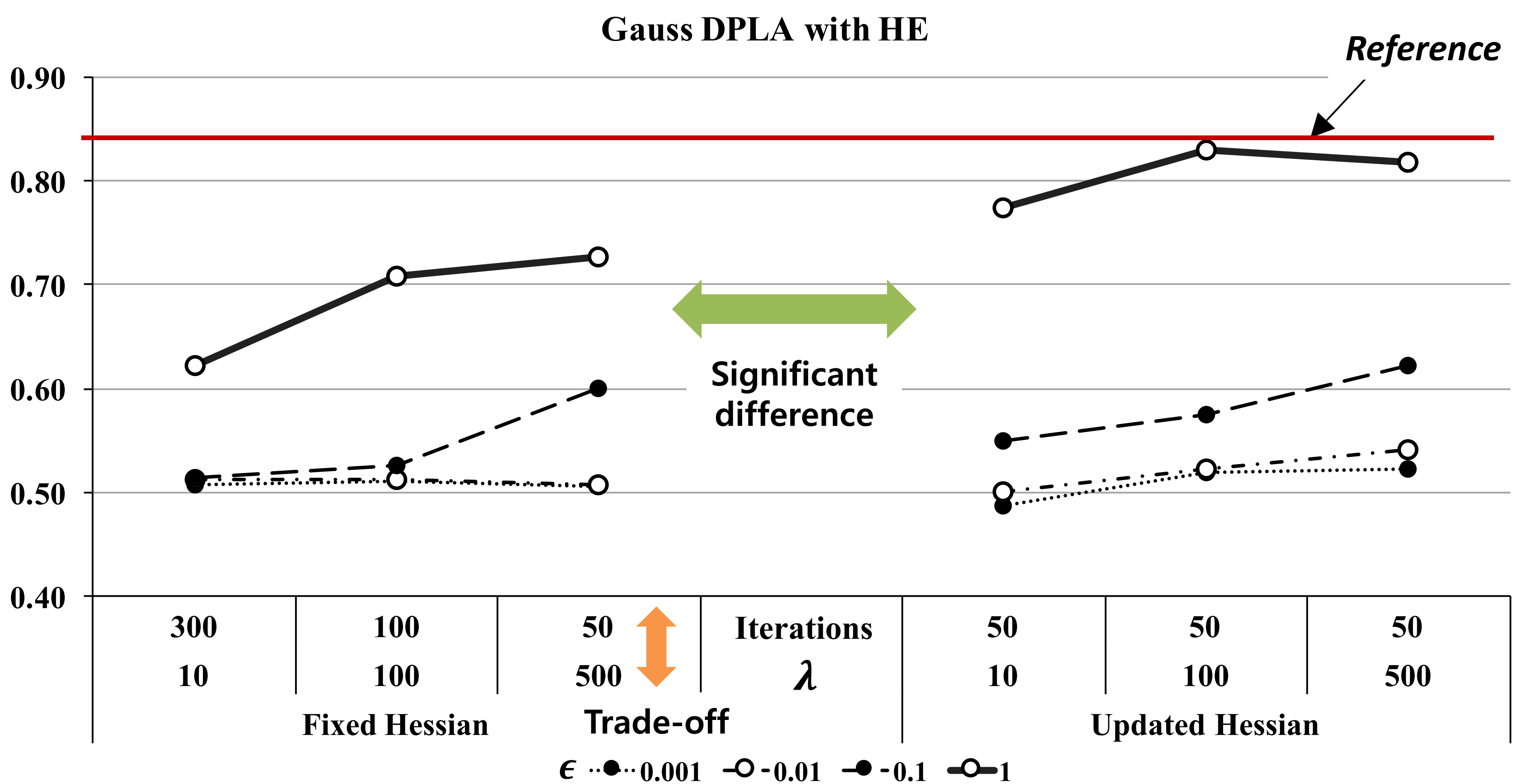
- **Averaged AUC on plaintext**



Gauss << Laplace ≈ Gamma

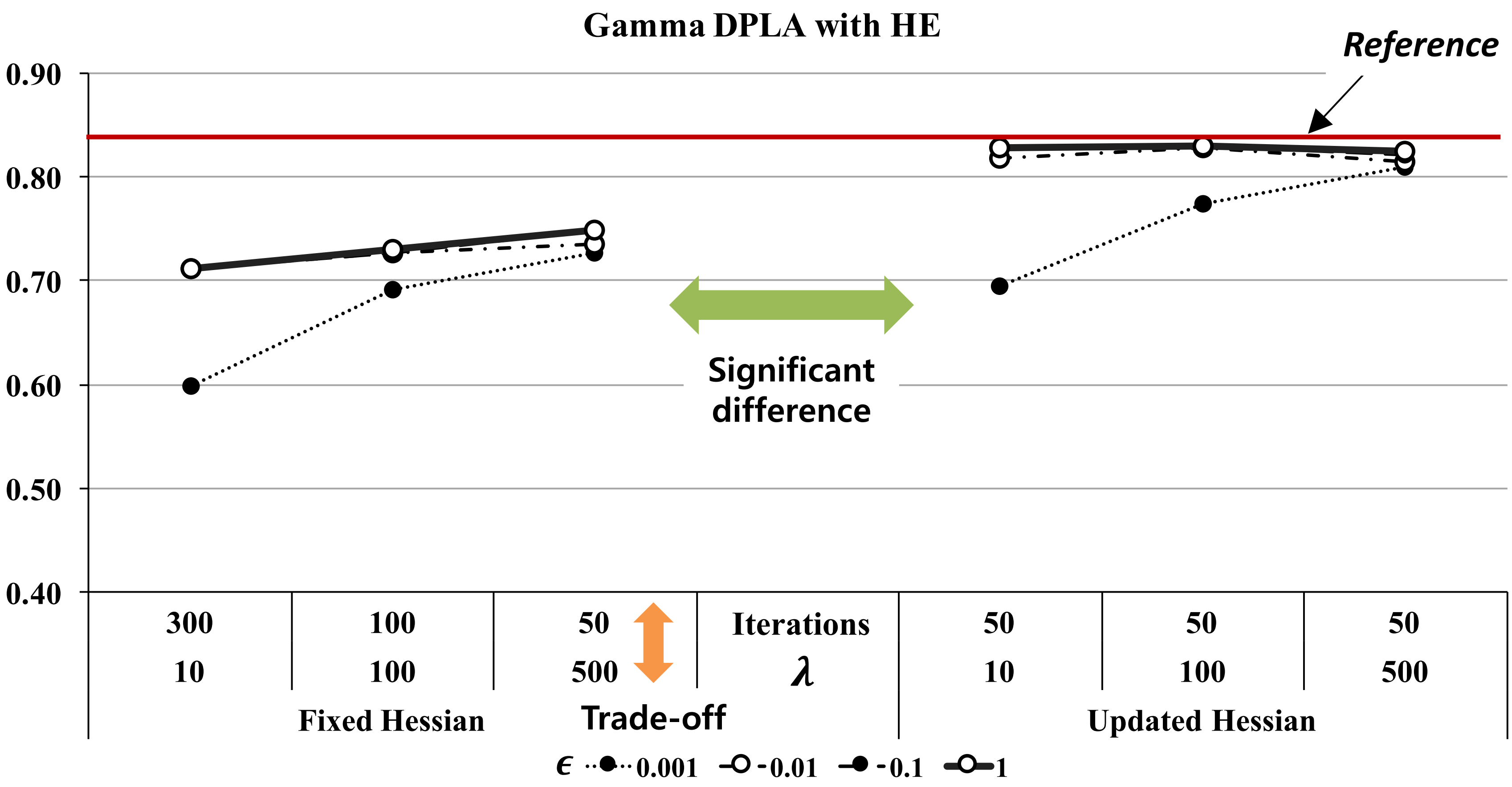
Gauss DPLA with HE

Budget: ϵ/iterations



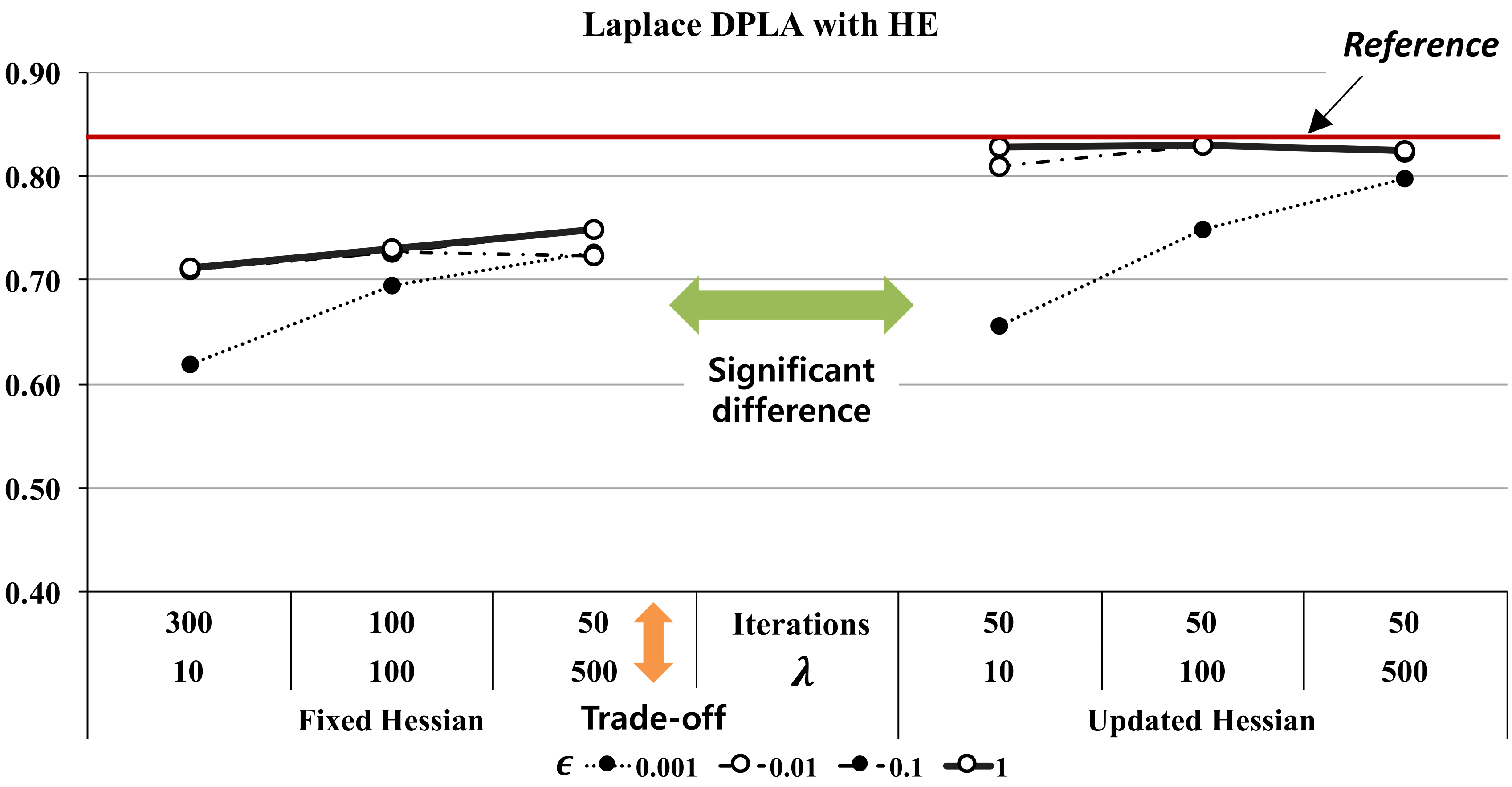
Gamma DPLA with HE

Budget: ϵ/iterations



Laplace DPLA with HE

Budget: ϵ/iterations



Same tendency: Gauss << Laplace ≈ Gamma

UC San Diego School of Medicine

Time Complexity

	Fixed Hessian	Updating Hessian
The number of Iterations	↑	↓
Time per iteration	↓	↑

	λ		
	10	100	500
	Fixed Hessian		
Iterations	50	100	300
Time (s)	**26.33**	49.36	**142.39**
	Trade-off depending on the number of iterations		
	Updated Hessian		
Iterations	50	50	50
Time (s)	**131.72**	**132.23**	**131.22**

We confirmed win-win strategy!

UC San Diego Division of Biomedical Informatics

UC San Diego School of Medicine

Acknoledgement

- Junghye Lee
- Miran Kim
- Shuang Wang
- Robert El-Kareh
- Lucila Ohno-Machado

- NIGMS grant

  R01GM118609

Q&A