

# Privacy Preserving Distributed Maximum Likelihood Estimation

Alan F. Karr<sup>\*</sup> and Xiaodong Lin<sup>†</sup>

## Abstract

In the environment of distributed computation, when data privacy is a concern, traditional statistical models cannot be applied directly. Thus, there is a need to develop statistical tools that can obtain proper analysis results while preserving data privacy. Individual privacy preserving statistical analysis protocols have been proposed for specific models recently. In this paper, we developed methods and protocols for privacy preserving maximum likelihood estimation in very general settings. These methods can be used in any statistical models that utilize maximum likelihood for parameter estimation. We developed models for both horizontally and vertically partitioned data, and proposed procedures that give participating parties the choice to withdraw from the joint computation. Our approach is illustrated through a detailed example on logistic regression.

## 1 Introduction

Although statistical analysis over combined data possesses huge potentials in knowledge discovery, it can also induce great disclosure risks. Traditionally, combined analysis runs on centralized data warehouses that collect data from various agencies. When confidential, proprietary, or private information is involved, data owners maybe be reluctant to disclose their data to the central site. Privacy preserving data mining has emerged as a promising approach to solving this dilemma.

Numerous methods have been proposed in recent years to address this challenge. There are two main categories of approaches. The first one uses data distortion techniques such

---

<sup>\*</sup>Karr is director, National Institute of Statistical Sciences, Research Triangle Park, NC 27709, karr@niss.org

<sup>†</sup>Lin is assistant professor, Department of Mathematical Sciences, 839 Old Chemistry Building, University of Cincinnati, Cincinnati, OH, 45221, linxd@math.uc.edu. Part of this research was conducted when he was conduct at the National Institute of Statistical Sciences, Research Triangle Park, NC 27709

as additive noise (Agrawal 2000, Zhu and Liu 2004), data swapping (Gotman 2007) and randomized response (Du 2004) to disguise the original data. Practitioners using these techniques need to balance the data utility of the distorted data with privacy preservation (Karr et al. 2005). Similar issues have been addressed in the statistical disclosure limitation literature, where centralized data is released subject to data confidentiality constraints. The other category of approaches are based on the secure multi-party Computation (SMC) techniques (Yao 1986). Briefly, an SMC problem deals with computing certain function on multiple inputs, in a distributed framework where each participant holds a subset of the inputs. SMC ensures that no more information is revealed to a participant in the computation than what can be inferred from the participant’s own inputs and the final output. Numerous SMC-based privacy-preserving data mining schemes have been proposed recently. Lindell and Pinkas(2000) used SMC to build decision trees over the horizontally partitioned data. Du and Zhan(2003) proposed a SMC-based solution to build decision trees over the vertically partitioned data. Also based on SMC, Vaidya and Clifton proposed the solutions to the clustering problem (2003) and the association rule mining problem for vertically partitioned data (2002). Alan et al. (2005) presented privacy preserving linear regression tools for both horizontally and vertically partitioned data. Other privacy preserving methods for classification, clustering, and feature selection have also appeared in recent literature (Verykios et al. 2004 and Clifton et al. 2003).

Data distortion techniques for privacy preserving distributed computation can usually be applied to different statistical models. However, this is compensated by the loss of data utility. Approaches based on SMC are more accurate but they typically aim at a specific methodology. Our goal in this paper is to develop more general SMC-based approaches that can be applied to different statistical models.

One of the key issues in statistical modeling is parameter estimation. For example, regression problems are essentially the estimation of the regression coefficients; model based clustering techniques rely on the estimation of parameters in a mixture model. One of the most popular approaches for parameter estimation is the Maximum Likelihood Estimation (MLE). Problem specific privacy preserving MLE methods have been developed for Finite Mixture Model (Lin et al. 2005) and Logistic regression (Fienberg et al. 2007). If we can derive general privacy preserving procedures for MLE, much effort on developing ad hoc approaches for privacy preserving statistical analysis can be saved.

Therein, we address the problem of privacy preserving maximum likelihood estimation for distributed data. Consider a random sample  $\mathbf{x}^n = \{x_1, \dots, x_n\}$ , while  $x_i$  follows  $f(X; \theta)$ .

The log likelihood function can be expressed as

$$l(\theta|\mathbf{x}^n) = \sum_{i=1}^n \log f(x_i; \theta).$$

The maximum likelihood estimator  $\hat{\theta}$  is defined as the maximizer for  $l(\theta|\mathbf{x}^n)$ . When data  $\mathbf{x}^n$  is distributed across different sites, our goal is to obtain the  $\hat{\theta}$  without sharing or pooling the data. In this paper, we will propose privacy preserving maximum likelihood solutions for both the horizontally and vertically partitioned cases. Several novel protocols are developed along the way.

The paper is organized as follows. First we consider the situation when data is horizontally partitioned. Assume that the data generating distribution is from the exponential family. The optimization problem is reduced to compute a sum securely, which can be solved by using the secure summation protocol. We then consider the general case for horizontally partitioned data where the MLE is estimated through Newton-Raphson procedure. For an iterative algorithm, the sharing of intermediate partial sum and partial covariance matrix can lead to undesirable information disclosure. To address this problem, we developed a new secure protocol for matrix multiplication and inversion, through which a new secure MLE procedure that reveals minimum intermediate results can be derived. In Section 3 we focus on cases where data are vertically partitioned. When the covariates are assumed to be independent, this problem can be solved in a similar fashion as the horizontally partitioned case using secure summation protocol. We then propose a protocol based on oblivious transfer (Du and Atallah 2001), through which any function depending on covariates that are distributed into different agencies can be computed securely. This protocol is used to build procedures for privacy preserving MLE. Section 4 discusses the validity of the usual assumption in MLE that the data are independent and identically distributed and proposes remedies for pattern identification with privacy concern. The secure procedure that enables participating parties to withdraw is proposed in Section 5. The paper concludes with a logistic regression example in Section 6 and closing remarks in Section 7.

## 2 Secure MLE for horizontally partitioned data

The *Horizontally partitioned* databases comprise the same numerical attributes for disjoint sets of data subjects. For example, several state or local school districts may want to combine their students' data to improve the precision of analyses for the general student population. Denote the combined data as  $\mathbf{x}^n = x_1, \dots, x_n$ , where  $x_i \in \mathcal{R}^p$ ,  $1 \leq i \leq n$  and the data is horizontally partitioned across  $m$  agencies. Each agency has a portion of data records from

$\mathbf{x}^n$  that contains the same data attributes.

## 2.1 Special case: exponential family

We first discuss the case when the random samples are generated from the exponential family of distributions. Namely, the density function has the form  $f(x) = b(x)\exp\{a(\theta)^T t(x) - c(\theta)\}$ . Given the random samples of size  $n$ ,  $\mathbf{x}^n$ , the log likelihood function is

$$l(\theta; \mathbf{x}^n) = \sum_{i=1}^n \log b(x_i) + \sum_{i=1}^n \{a(\theta)^T t(x_i) - c(\theta)\}. \quad (1)$$

Since the optimization is on  $\theta$ , the MLE is

$$\hat{\theta} = \arg \max_{\theta} a(\theta)^T \sum_{i=1}^n t(x_i) - nc(\theta).$$

The only quantity in the right hand side that depends on data values is  $\sum_{i=1}^n t(x_i)$ . When the number of agencies  $m$  is greater than 2, this quantity can be computed jointly using the secure summation protocol and shared among the participating agencies. Then the MLE  $\hat{\theta}$  can be computed locally.

## 2.2 Secure MLE based on Newton Raphson procedure

When analytically solution is hard to obtain, iterative procedures are usually used to compute MLE. One of the most popular techniques is the Newton Raphson iteration for root evaluation, which finds the local maxima of the log likelihood function  $l(\theta; \mathbf{x}^n)$ .

The typical Newton Raphson procedure at step  $s$  goes as follows. Assume that we have the estimator of  $\theta$  from step  $s - 1$ , denoted as  $\theta^{(s-1)}$  and that the first and second derivatives of the likelihood function exist. The estimator in the  $s$ th step is

$$\theta^{(s)} = \theta^{(s-1)} - (D^2 l(\mathbf{x}^n; \theta^{(s-1)}))^{-1} \nabla l(\mathbf{x}^n; \theta^{(s-1)}),$$

where  $D^2 l(\mathbf{x}^n; \theta^{(s-1)})$  is the Hessian matrix of the likelihood function evaluated at  $\theta^{(s-1)}$  and  $\nabla l(\mathbf{x}^n; \theta^{(s-1)})$  is the gradient.

### 2.2.1 Newton Raphson using secure summation protocol

Our goal now is to obtain the parameter update securely. Assume  $\theta = \{\theta_1, \dots, \theta_k\}$ ,

$$\begin{aligned}\nabla_{\theta} l(\mathbf{x}^n; \theta^{(s-1)}) &= \left( \frac{\partial l}{\partial \theta_1}, \dots, \frac{\partial l}{\partial \theta_k} \right) \\ &= \left( \sum_{i=1}^n \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_1}}{f(x_i; \theta)}, \dots, \sum_{i=1}^n \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_k}}{f(x_i; \theta)} \right)_{\theta^{(s-1)}}.\end{aligned}$$

Locally, we can compute  $L_j$ ,  $1 \leq j \leq m$ , where

$$L_j = \left( \sum_{i=1}^{n_j} \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_1}}{f(x_i; \theta)}, \dots, \sum_{i=1}^{n_j} \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_k}}{f(x_i; \theta)} \right)_{\theta^{(s-1)}}.$$

Similarly, we can compute each element for the Hessian matrix of the likelihood function at the current step locally, and obtain the Hessian matrix  $H_j$  at the local level, where the element of  $H_j$  is

$$H_j(h, l) = \sum_{i=1}^{n_j} \left( \frac{\frac{\partial^2 f(x_i; \theta)}{\partial \theta_h \partial \theta_l}}{f(x_i; \theta)} - \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_h} \frac{\partial f(x_i; \theta)}{\partial \theta_l}}{f^2(x_i; \theta)} \right)_{\theta^{(s-1)}},$$

where  $n_j$  is the data size of agency  $j$ . The iteration step for Newton-Raphson becomes

$$\theta^{(s)} = \theta^{(s-1)} - \Delta^{(s-1)}, \quad \text{where } \Delta^{(s-1)} = \left( \sum_{j=1}^m H_j \right)^{-1} \left( \sum_{j=1}^m L_j \right).$$

The  $H_j$  and  $L_j$  can be computed at each agency locally, and we use secure summation protocol to obtain the parameter updates.

### 2.2.2 security analysis

In the current approach, what shares among agencies are the value of  $\sum_{j=1}^m H_j$  and  $\sum_{j=1}^m L_j$  at each iteration. When  $m > 2$ , secure summation protocol guarantees that no individual  $H_j$  and  $L_j$  are revealed to the other agencies. However, the minimum information that we need for parameter updating is  $(\sum_{j=1}^m H_j)^{-1} (\sum_{j=1}^m L_j)$  and the sharing of  $\sum_{j=1}^m H_j$  and  $\sum_{j=1}^m L_j$  among the agencies during each iteration step reveals abundant information. We proposed the following protocol to address this issue.

### 2.2.3 Second protocol

Without loss of generality, assume that there are two participating agencies (Note that the case of  $m = 2$  is in fact the most difficult one, as the secure summation protocol fails, and the previous method does not work). Our task is to compute  $(H_1 + H_2)^{-1}(L_1 + L_2)$  and preserve data privacy. Denote  $X = (H_1 + H_2)^{-1}(L_1 + L_2)$ , the problem is equivalent to solving the linear equation system  $(H_1 + H_2)X = (L_1 + L_2)$ .

Our protocol goes as the following. Site one generates a  $k$  by  $k$  matrix  $M_1$ , which is of rank  $k/2$  (here we assume that  $k > 2$ , and without loss of generality, assume  $k$  is an even number). Site one sends  $M_1$  to site two. Site two then computes  $M_1H_2$  and  $M_1L_2$ , sends them back to site one. Now site one can produce the linear equation system

$$M_1(H_1 + H_2)X = M_1(L_1 + L_2).$$

Symmetrically, site two generates  $M_2$  that is of rank  $k/2$  and sends to site one. Following similar steps, site two can produce the linear equation system

$$M_2(H_1 + H_2)X = M_2(L_1 + L_2).$$

Our goal is to combine these two linear equation systems and solve for  $X$ . However, if we share  $M_1(H_1 + H_2)$  with site two, it can then compute  $H_1$ . To solve this problem, we generate a full rank matrix  $T_1$ , the linear equation system produced by site one becomes

$$T_1M_1(H_1 + H_2)X = T_1M_1(L_1 + L_2).$$

Similarly we generate  $T_2$  at site two to obtain

$$T_2M_1(H_1 + H_2)X = T_2M_1(L_1 + L_2).$$

Combining these two linear equation system will produce the same solution of  $X$  as before.

### 2.2.4 Security analysis

Site one sends to site two  $M_1$ ,  $M_2H_1$ ,  $M_2L_1$ ,  $T_1M_1(H_1 + H_2)$  and  $T_1M_1(L_1 + L_2)$ . Site one can check that the rank of  $M_2$  is  $k/2$ . When  $k > 2$ ,  $H_1$  and  $L_1$  are not revealed. Since site two does not know  $T_1$ , sharing of  $T_1M_1(H_1 + H_2)$  will not reveal the value of  $H_1$ . Similar arguments can be made on the other components. Clearly, one obvious disadvantage of this protocol is that when  $k$  is 1 or 2, the protocol breaks down since  $H_1$ ,  $H_2$ ,  $L_1$  and  $L_2$  can be learned, and the degree of privacy protection for this protocol depends on how large  $k$  is.

### 2.2.5 Error propagation

Our secure procedure involves matrix multiplications and inversion which are computed locally. Due to the varieties of computation environment, results with errors and differences on precision are expected. In our approaches, each site computes their own  $\Delta^{(s-1)}$  at step  $s$ , and they jointly obtain the parameter update  $\theta^{(s)}$  thereafter. If say the  $\theta^{(s)}$  computed by site  $k$  deviates from the true value, this error will propagate to the next step when used to compute  $L_k$  and  $H_k$ . To solve this problem, we propose to perform the following error checking procedures before the end of each iteration step:

1. Step one, for site 1 to  $m$ , compute  $\theta^{(s)}$ , denoted as  $\theta_j^{(s)}$ ,  $1 \leq j \leq m$ .
2. Step two, start with site one, using secure summation protocol to compute  $\sum_{j=1}^m \theta_j^{(s)}$ . Also compute  $m\theta_1^{(s)}$ .
3. Step three, use secure boolean procedure to check if  $|\sum_{j=1}^m \theta_j^{(s)} - m\theta_1^{(s)}| < \tau$ , where  $\tau$  is a threshold that all the participants agree. If the result is 1, namely the difference is less than the threshold, we go to the next round. Otherwise we choose to stop or redo the analysis.

## 3 Vertically partitioned case

Assume that  $\mathbf{x}^n = \{x_1, \dots, x_n\}$ , where  $x_i = (x_i^1, \dots, x_i^p)$ ,  $1 \leq i \leq n$ . In the vertically partitioned case, each agency owns only portion of the variables for  $x_i$ . We assume that the functional forms of the joint density and the corresponding Hessian Matrix and Gradient are known to the participating parties.

### 3.1 Independent variable case

For the simplest case, assume that  $f(x_i, \theta) = \prod_{s=1}^t f_s(x_i^s; \theta_s)$ . The log likelihood function can be written as

$$l = \sum_{i=1}^n \log \prod_{s=1}^t f_s(x_i^s; \theta_s) = \sum_{s=1}^t \left[ \sum_{i=1}^n \log f_s(x_i^s; \theta_s) \right].$$

Clearly the right hand side can be optimized locally to obtain the ideal  $\theta_s$ .

Now assume that  $f(x_i, \theta) = \prod_{s=1}^t f_s(x_i^s; \theta)$ . Namely, each component density function involves the whole parameter vector  $\theta$ . The log likelihood function becomes

$$l = \sum_{s=1}^t \left[ \sum_{i=1}^n \log f_s(x_i^s; \theta) \right].$$

Taking the first derivative of this function with respect to  $\theta$  we obtain

$$\frac{\partial l}{\partial \theta} = \sum_{s=1}^t \left\{ \sum_{i=1}^n \left[ \frac{1}{f_s(x_i^s; \theta)} \frac{\partial f_s(x_i^s; \theta)}{\partial \theta} \right] \right\}.$$

This quantity can be written as  $\sum_{j=1}^m L_j$ , where  $L_j$  can be computed locally at agency  $j$ . Similarly, the Hessian matrix  $H_j$  can be computed locally. Thus the update for Newton Raphson iteration at step  $s$  is

$$\theta^{(s)} = \theta^{(s-1)} - \left( \sum_{j=1}^m H_j \right)^{-1} \left( \sum_{j=1}^m L_j \right).$$

Similar procedures in horizontally partitioned case applies.

### 3.2 General case: exponential family

Now we consider the secure MLE evaluation for the exponential family without the independence assumption over the covariates. The likelihood function in this case still has the form (1), and the MLE is

$$\hat{\theta} = \arg \max_{\theta} a(\theta)^T \sum_{i=1}^n t(x_i) - nc(\theta).$$

WLG, assume that there are two agencies A and B. Agency A holds  $(x_i^1, \dots, x_i^k)$ , and agency B holds  $(x_i^{k+1}, \dots, x_i^p)$ ,  $1 \leq i \leq n$ . In order to obtain  $\hat{\theta}$ , we need to compute  $\sum_{i=1}^n t(x_i)$  securely. Assume that the form of function  $t$  is known to both agencies A and B.

*Protocol three. Secure two party functional computation for vertical partitioned data*

Given the vertically partitioned data:  $(x_i^1, \dots, x_i^k)$  for Agency A and  $(x_i^{k+1}, \dots, x_i^p)$  for Agency B. Compute  $t(x_i^1, \dots, x_i^k; x_i^{k+1}, \dots, x_i^p)$  without sharing original data between the two agencies.

For simplicity, consider the case where A has one attribute  $x_1$  and B had one attribute  $x_2$ . The general case can be readily obtain following the same procedure.

For  $i = 1$  to  $n$ ,

- Step one. Agency A generate a vector of length  $s$ , among which the  $k$ th item  $x_{1,i}^k = x_{1,i}$ . The other  $s - 1$  items are random copies.
- Step two. A sends these  $s$  copies to B, B computes  $t(x_{1,i}^1, x_{2,i}), \dots, t(x_{1,i}^s, x_{2,i})$ . Denote these by  $t^1, \dots, t^s$ . B generates a random value  $\epsilon_i$  ( if the values of function  $t$  is multi-

dimensional, generate a random vector instead). Define  $g_i^1 = t^1 - \epsilon_i, \dots, g_i^s = t^s - \epsilon_i$ .

- Step three. Agency A obtains  $g_i^k$  using 1 out of  $s$  oblivious transfer.

End for.

After the loop, Agency A has  $g_i^k$  for all  $i$  thus  $\sum_{i=1}^n g_i^k$  and Agency B has  $\epsilon_i$  for all  $i$  thus  $\sum_{i=1}^n \epsilon_i$ . The sum of these two items give  $\sum_{i=1}^n t(x_{1,i}, x_{2,i})$ .

Note that this protocol can be used for the computation of any known function  $t$  over the covariates, while the data are vertically partitioned.

### 3.3 Security and performance analysis for protocol three

Agency A sends  $s$  copies to Agency B, the  $k$ th item is the desired value. Agency B has  $1/s$  chances of guessing right (more secure versions of oblivious transfer are available, see Du and Atallah (2001)). Agency A obtains  $g_i^k = t^k - \epsilon_i$ . Since  $\epsilon_i$  is a random value generated by Agency B, A does not know the real value of  $t^k$ , thus the value of  $x_{2,i}$  is not revealed. At the end of the loop the sum of  $\sum_{i=1}^n g_i^k$  and  $\sum_{i=1}^n \epsilon_i$  are shared, but not the individual value. The protocol is not symmetric, which is mainly due to the fact that the 1 out of  $N$  oblivious transfer protocol is asymmetric.

This approach not only achieves a desire level of privacy protection for data from both parties, it is also efficient. The total communication cost equals  $n * s + n * L(s)$ .  $L(s)$  is the communication cost for 1 out of  $s$  oblivious transfer, which is achievable in linear time with respect to  $s$ .

### 3.4 General case: Newton-Raphson

Now consider the general case for the vertically partitioned data. The estimator for  $\theta$  at the  $s$  step is

$$\theta^{(s)} = \theta^{(s-1)} - (D^2 l(\mathbf{x}^n; \theta^{(s-1)}))^{-1} \nabla l(\mathbf{x}^n; \theta^{(s-1)}),$$

Where

$$\nabla_{\theta} l(\mathbf{x}^n; \theta^{(s-1)}) = \left( \sum_{i=1}^n \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_1}}{f(x_i; \theta)}, \dots, \sum_{i=1}^n \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_k}}{f(x_i; \theta)} \right)_{\theta^{(s-1)}}. \quad (2)$$

and

$$D^2(h, l) = \sum_{i=1}^n \left( \frac{\frac{\partial^2 f(x_i; \theta)}{\partial \theta_h \partial \theta_l}}{f(x_i; \theta)} - \frac{\frac{\partial f(x_i; \theta)}{\partial \theta_h} \frac{\partial f(x_i; \theta)}{\partial \theta_l}}{f^2(x_i; \theta)} \right)_{\theta^{(s-1)}}.$$

Under the assumption that both agencies A and B know the functional form of  $\nabla_{\theta} l(\mathbf{x}^n; \theta^{(s-1)})$  and  $D^2(h, l)$ , protocol three can be used to compute their values securely. Thus the agencies

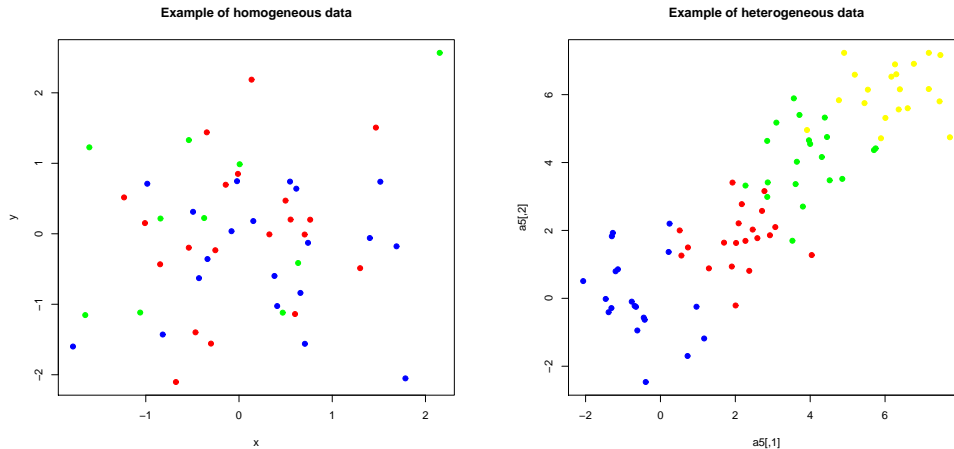


Figure 1: Comparing homogeneous data and heterogeneous data.

can jointly obtain the stepwise update for  $\theta^{(s)}$ .

## 4 The i.i.d. assumption for the random sample

So far we have assumed that the data from different sites are i.i.d. samples from the same distribution  $f(x|\theta)$ . This, however, is probably not the most interesting case. In the left plot of Figure (??), the four groups of data are all generated from the same bivariate normal distribution. Pooling the data merely gains over precision of the estimates. The more interesting case is illustrated in the right hand side plot of Figure (??), which shows a heterogeneous dataset. None of these four groups shows any linear trend, while the combined data clearly possesses a upward linear pattern. If we do not perform combined analysis, this linear trend can not be identified. In the latter case, the global data is clearly correlated while independency can possibly be claimed after a linear tranformation of the two variables. More generally, we consider the model

$$g(x^1, x^2, \dots, x^l) = \epsilon,$$

where  $\epsilon \sim f(.|\theta)$  and  $x^1$  can be just one of the dependent variables, or the response variable usually denoted as  $y$ . The  $f(.|\theta)$  is known to all the participating parties and so is the parametric form of  $g$ . The function  $g$  belongs to a class of functions  $\mathcal{G}$ . Thus we can write out the log likilihood function

$$l(.|\theta) = \sum_{i=1}^n \log f(g(x_i^1, \dots, x_i^l)).$$

The goal is to identify the optimum  $g \in \mathcal{G}$  that maximizes the likelihood (penalized or pseudo likelihood when different loss functions are used). For a simple linear regression model, the function  $g$  has the form of

$$g = x^1 - \sum_{s=2}^l \beta^s x^s,$$

and the log likelihood is

$$l = (2\pi\sigma^2)^{-n/2} - \frac{\sum_{i=1}^n (x_i^1 - \sum_{s=2}^l \beta^s x_i^s)^2}{2\sigma^2}.$$

The zeros of this likelihood function can be found explicitly. Thus the previously developed protocols can be applied to estimate  $\theta$  as well as those parameters in the function  $g$ . Note that the data  $\mathbf{x}^n$  need not to be independent, which fits the setting of heterogeneous data setting well.

## 5 Opt. out strategy and simulations

The lack of options to withdraw from the combined analysis is one of the main disadvantages for existing privacy preserving data mining tools. As show in the vertically partitioned linear regression case (Alan et al. 2007), parties that possess more variables are likely to lose more private information. Consider the extreme case that one party has one variable and the other party has 100, most of the inferential power is provided by the second party, and it may want to withdraw from the computation when observing this fact. To make privacy preserving distributed computing tools attractive to practioners, there is a need to build modules that provide participating parties the choice to opt. out. Obviously, the number of data points  $n_j$  and the number of variables  $l_j$ , for the  $j$ th party can be used as such a measure.

The proportions of data owned by the agencies are important factors. This is illustrated clearly in Figure (??). Other factors include the difference of  $R^2$  between the global model and the local model, as well as the differences between the estimated  $\beta$ . For instance, if party A realizes that its parameter estimates are very close to the global estimate, it can not gain much from the global analysis. Also, the other parties may be able to find that party A's local model is very close to the global model. This is a potential privacy threat for party A.

We propose to use the Sum of Square Error and Mean Square Error as opt. out measures for horizontally partitioned data. First, at each site  $j$ ,  $1 \leq j \leq m$ , check if  $SSE_{local}/SSE_{global} < \delta_1$ , where  $\delta_1$  is a user specified threshold. If the statement is true, site  $j$  chooses to opt. out. The reason for this check is that when the local SSE is small

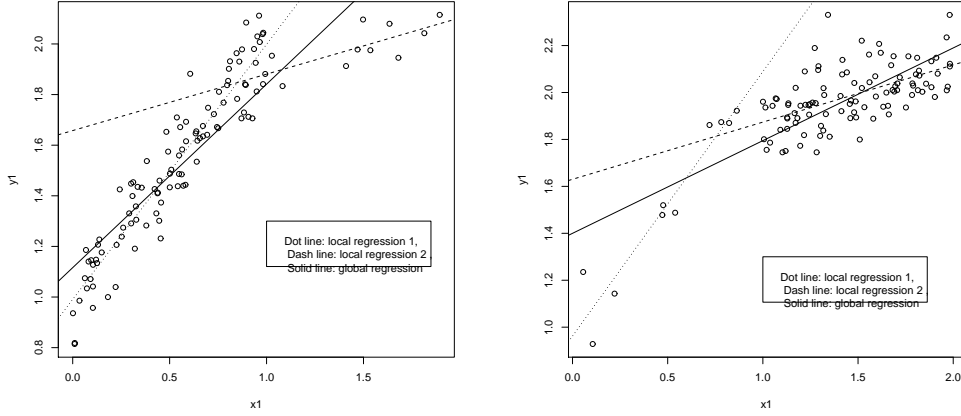


Figure 2: Regression example for opt. out strategy.

comparing to the global SSE, the global model explains the local data well, thus the local model can be at risk. Otherwise, check (2) if  $MSE_{local}/MSE_{global} < \delta_2$ . When a particular party possess a majority portion of the data, though the  $SSE_{local}/SSE_{global}$  ratio is big, it may due to the fact that the data size is large, and a larger sample size increases the risk factor.

More generally, we propose to use the Fisher information as our opt. out measure. The idea is that the Fisher information is the amount of information that an observable random variable  $X$  carries about an unknown parameter  $\theta$  upon which the likelihood function of  $X$  depends. In the MLE setting, for a particular set of parameters of interest, the closer the local Fisher information is to the global Fisher information, the less gain and more risk the local party will have.

The Fisher information matrix for  $k$  parameters  $\theta$  has element

$$(\mathcal{I}(\theta))_{qh} = -\mathbf{E}\{l''(\theta)\} = -\mathbf{E} \left[ \sum_{i=1}^n \frac{\partial^2}{\partial \theta_q \partial \theta_h} \log f(X_i; \theta) \right],$$

where  $1 \leq q; h \leq k$ . This quantity is usually approximated by the observed fisher information matrix with element

$$(\mathbf{J}(\theta))_{qh} = -\sum_{i=1}^n \frac{\partial^2}{\partial \theta_q \partial \theta_h} \log f(X_i; \theta).$$

At each step, we can calculate and share among the agencies the  $\mathbf{J}$ . Each party also calculate its own  $\mathbf{J}$  using the local data. If the difference between these two information matrix is small, then the party should consider the option of withdrawing from this joint computation.

## 6 Case study: Privacy preserving distributed logistic regression using secure MLE

Logistic regression is one of the most commonly used classification technique in machine learning and data mining applications. Logistic Regression assumes a parametric form for the distribution  $P(Y|X)$ , where  $Y$  is discrete valued representing class label and  $X = \langle X^1, \dots, X^l \rangle$  are the explanatory valuables. Logistic regression directly estimates its parameters from the training data. When  $Y$  is a binary variable, the conditional distribution has the following parametric form

$$P(Y = 0|X) = \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j X^j)}$$

and  $P(Y = 1|X) = 1 - P(Y = 0|X)$ . To estimate the parameters given the training data  $(y_1, x_1), \dots, (y_n, x_n)$ , the log-likelihood is

$$\begin{aligned} l &= \sum_{i=1}^n y_i \log \left( 1 - \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j)} \right) + (1 - y_i) \log \frac{1}{1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j)} \\ &= \sum_{i=1}^n y_i \left( \beta_0 + \sum_{j=1}^l \beta_j x_i^j \right) - \log \left( 1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j) \right). \end{aligned}$$

In order to use the Newton-Raphson procedure, we need to securely compute

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^n x_i^j \left[ y_i - \frac{\exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j)}{1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j)} \right], \quad 0 \leq j \leq l,$$

where we  $x_i^0 = 1$  for all  $i$  and

$$\frac{\partial^2 l}{\partial \beta_j \partial \beta_k} = \sum_{i=1}^n x_i^j \left[ -\frac{\exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j) x_i^k}{(1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j))^2} \right], \quad 1 \leq j, k \leq l.$$

### Horizontally partitioned case

When data  $\mathbf{x}^n$  are horizontally partitioned across different sites, at step  $s$ ,

$$\theta^{(s)} = \theta^{(s-1)} - \Delta^{(s-1)},$$

where  $\Delta^{(s-1)}$  is defined in Section 2.2. We can compute both the gradient and the Hessian matrix using the secure summation protocol, and perform the parameter updates. We can also use protocol two to compute  $\Delta^{(s-1)}$  directly.

### Vertically partitioned case

In the vertical partitioned case, we can use protocol three to compute  $\Delta^{(s-1)}$  jointly.

### Opt. out

The observed fisher information matrix in this case has the following form

$$\mathbf{J}_{jk} = \left( \sum_{i=1}^n x_i^j \left[ \frac{\exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j) x_i^k}{(1 + \exp(\beta_0 + \sum_{j=1}^l \beta_j x_i^j))^2} \right] \right)_{jk}.$$

We can use secure summation to compute the global Fisher information matrix  $\mathbf{J}_{global}$ , each site can compute  $\mathbf{J}_{local}$  locally. If  $\|\mathbf{J}_{global} - \mathbf{J}_{global}\|_F < \delta_3$  for some threshold  $\delta_3$ , then this particular party may choose to opt. out.

## 7 Appendix

**Protocol: Secure summation.**

**Protocol: 1 out of N oblivious transfer.**

## References

- [1] Agrawal, D. and Aggarwal, C. C. (2001). On the design and quantification of privacy preserving data mining algorithms. *In Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principals of Database Systems*, 247-255.
- [2] Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. *In Proceedings of the 2000 ACM SIGMOD Conference on Management of Data*, 439-450.
- [3] Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. and Zhu, M. (2003). Tools for Privacy Preserving Distributed Data Mining. *ACM SIGKDD Explorations*, **4**, No.2.
- [4] Du, W. and Zhan, Z. (2003). Using Randomized Response Techniques for Privacy-Preserving Data Mining. *Proceedings of The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 505-510.
- [5] Evfimievski, A. (2002). Randomization in Privacy-Preserving Data Mining. *ACM SIGKDD Explorations*, **4**, No. 2, 43-49.

- [6] Fienberg, S. E., Karr, A. F., Nardi, Y. and Slavkovic, A. (2007). Secure Logistic Regression with Distributed Databases. *Bulletin of the International Statistical Institute Meetings*, to appear.
- [7] Goldreich, O., Micali, S. and Wigderson, A. (1987) How to play any mental game - a completeness theorem for protocols with honest majority. *19th ACM Symposium on the Theory of Computing* , 218-229.
- [8] Gomatam, S., Karr, A. F., and Sanil, A. P. (2007) Data swapping as a decision problem, *Journal of Official Statistics*, to appear.
- [9] Karr, A. F., Lin, X., Sanil, A. P., and Reiter, J. P. (2005) Secure Regression on Distributed Databases. *Journal of Graphical and Computational Statistics*. **14**, 263 - 279.
- [10] Karr, A. F., Kohonen, C. N., Oganian, A., Reiter, J. P. and Sanil, A. P. (2006). A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician* **60**, No. 3, 224-232.
- [11] Karr, A. F., Lin, X., Sanil, A. P. and Reiter, J. P. (2007). Privacy Preserving Analysis of Vertically Partitioned Data Using Secure Matrix Products. Revised for *Journal of Official Statistics*.
- [12] Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. *Advances in Cryptology CRYPTO 2000, Springer-Verlag, Berlin*,36-54.
- [13] Lin, X., Clifton, C, and Zhu, Y. (2005) Privacy Preserving Clustering with Distributed EM Mixture Modeling. *Journal of Knowledge and Information Systems*. **8** 68-81.
- [14] Zhu, Y. and Liu, L. (2004). Optimal Randomization for Privacy Preserving Data Mining, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 761-766.
- [15] Vaidya, J. S. and Clifton, C. (2002). Privacy preserving association rule mining in vertically partitioned data, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 639-644.
- [16] Vaidya, J. and Clifton, C. (2003). Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 206-215.

- [17] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y. and Theodoridis, Y. (2004). State-of-the-art in Privacy Preserving Data Mining, *ACM SIGMOD Record*, **3**, No. 1, 50-57.
- [18] Yao, A. C. (1986). How to generate and exchange secrets, *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science* , 162-167.