Journal of Privacy and Confidentiality (2009) **1**, Number 2, pp. 213-222

Privacy-Preserving Maximum Likelihood Estimation for Distributed Data

Xiaodong Lin* and Alan F. Karr†

Abstract. Recent technological advances enable the collection of huge amounts of data. Commonly, these data are generated, stored, and owned by multiple entities that are unwilling to cede control of their data. This distributed environment requires statistical tools that can produce correct results while preserving data privacy. Privacy-preserving protocols have been proposed to solve specific statistical analysis such as linear regression, clustering, and classification. In this paper, we present methods and protocols for privacy-preserving maximum likelihood estimation in general settings. We discuss both horizontally and vertically partitioned data, and propose procedures that allow participating parties to withdraw from the joint computation. Logistic regression is used to demonstrate our method.

1 Introduction

Although statistical analyses that combine distributed data possess huge potential in knowledge discovery, they can also induce great disclosure risks. Traditionally, combined analyses were performed in centralized data warehouses that collect data from various sources. When confidential, proprietary, or private information is involved, data owners[1] may be reluctant to furnish their data to the central site.

Privacy-preserving data mining has emerged as a promising approach to solving this dilemma. These methods complement traditional techniques for statistical disclosure limitation, which focus on protecting identities of data subjects and values of sensitive attributes within a single database. Underlying them is the concept of secure multiparty computation (SMPC) (Yao, 1986), which deals with computing the value of a function with multiple inputs, in a distributed framework where each participant holds a subset of the inputs. At its best, SMPC ensures that no more information is revealed to a participant than can be inferred from the participant's own inputs and the final output. Numerous SMPC-based privacy-preserving data mining schemes have been proposed in the literature, for decision trees (Lindell and Pinkas, 2000; Du and Zhan, 2002), clustering (Vaidya and Clifton, 2004), association rules (Clifton et al., 2003), feature selection (Verykios et al., 2004), as well as for specific statistical procedures such as linear regression (Karr et al., 2007).

In this paper, we focus on the fundamental statistical problem of maximum likelihood

*Department of Management Science & Information Systems, Rutgers University, 94 Rockafeller Road, Piscataway, NJ, mailto:xiaodonglin@gmail.com Part of this research was conducted during visits to the National Institute of Statistical Sciences.

†National Institute of Statistical Sciences, Research Triangle Park, NC, mailto:karr@niss.org

[1]Which we refer to in this paper as (official statistics) agencies, although they may be companies or other organizations.

(ML) estimation for finite-dimensional parameters. Previous research has addressed only special cases (Fienberg et al., 2007; Lin et al., 2005).

Our setting is the "traditional" one of independent random vectors $X_1, \ldots, X_n$ with common parametric density $\mathbf{X} = f(x;\theta)$, where the associated log-likelihood function can be expressed as

$$\ell(\theta|\mathbf{X}) = \sum_{i=1}^{n} \log f(X_i;\theta), \tag{1}$$

and the ML estimator $\hat{\theta}$ is its maximizer. Below we address both horizontal partitioning, where the X_i are distributed across sites (§2), and vertical partitioning, where each site has some components of all of $X_1, \ldots, X_n$ (§3). Approximate as well as exact methods of computational are considered.

2 Horizontally Partitioned Data

Horizontally partitioned databases contain 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 $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_i \in \mathbb{R}^p$. Assume that there are K agencies and let $\boldsymbol{\ell}_k \subset \{1, \ldots, n\}$ by the data records held be agency k. We assume that the $\boldsymbol{\ell}_k$ are disjoint.

2.1 Exponential Families

The general principle underlying the techniques in Karr et al. (2007) is that any analysis for which the sufficient statistics are additive across agencies can be performed by means of secure summation (Benaloh, 1987). In particular, this occurs when the density of the X_i is from an exponential family:

$$f(x;\theta) = b(x)\exp[a(\theta)^T t(x) - c(\theta)].$$

In this case,

$$\ell(\theta|\mathbf{X}) = \sum_{i=1}^{n} \log b(X_i) + \sum_{i=1}^{n} \left[a(\theta)^T t(X_i) - c(\theta)\right]. \tag{2}$$

The only component of the right-hand side of (2) that depends on the data is

$$\sum_{i=1}^{n} t(X_i) = \sum_{k=1}^{K} \left[\sum_{i \in \boldsymbol{\ell}_k} t(X_i)\right],$$

which can be calculated using secure summation, and then each agency can compute

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

on its own.

2.2 Numerical ML Estimation via Newton-Raphson

When analytical solution is not possible, iterative algorithms are used to compute ML estimators. One of the most popular of these is the Newton-Raphson algorithm for root evaluation, which finds local maxima of the log-likelihood function $l(\theta|\mathbf{X})$ by locating a zero of its derivative. More specifically, assume that the first and second derivatives of the likelihood function with respect to θ exist. Then given an estimator $\hat{\theta}^{(s-1)}$ of θ at step $s-1$, the estimator at step s is

$$\hat{\theta}^{(s)} = \hat{\theta}^{(s-1)} - [D^2\ell(\mathbf{X}|\hat{\theta}^{(s-1)})]^{-1}\nabla\ell(\mathbf{X}|\hat{\theta}^{(s-1)}), \tag{3}$$

where $D^2\ell(\mathbf{X}|\hat{\theta}^{(s-1)})$ is the Hessian matrix of the log-likelihood function evaluated at $\hat{\theta}^{(s-1)}$ and $\nabla\ell(\mathbf{X}|\hat{\theta}^{(s-1)})$ is the gradient.

Let $\theta = (\theta_1, \ldots, \theta_q)$. Then for each $1 \le j \le q$,

$$\nabla_\theta\ell(\mathbf{X}|\hat{\theta}^{(s-1)})(j) = \left(\sum_{k=1}^{K}\sum_{i\in\boldsymbol{\ell}_k}\frac{\frac{\partial f(X_i;\theta)}{\partial\theta_j}}{f(X_i;\theta)}\right)_{\hat{\theta}^{(s-1)}}, \tag{4}$$

and similarly, for each h and j,

$$D^2\ell(\mathbf{X}|\hat{\theta}^{(s-1)})(h,j) = \sum_{k=1}^{K}\sum_{i\in\boldsymbol{\ell}_k}\left(\frac{\frac{\partial^2 f(X_i;\theta)}{\partial\theta_h\partial\theta_j}}{f(X_i;\theta)} - \frac{\frac{\partial f(X_i;\theta)}{\partial\theta_h}\frac{\partial f(X_i;\theta)}{\partial\theta_j}}{f^2(X_i;\theta)}\right)_{\hat{\theta}^{(s-1)}}, \tag{5}$$

Each of these is computable using secure summation, and so is the Newton-Raphson step, using (3).

To understand the security implications of this approach, let

$$L_k(j) = \left(\sum_{i\in\boldsymbol{\ell}_k}\frac{\frac{\partial f(X_i;\theta)}{\partial\theta_1}}{f(X_i;\theta)}\right)_{\hat{\theta}^{(s-1)}}$$

and

$$H_k(h,j) = \sum_{i\in\boldsymbol{\ell}_k}\left(\frac{\frac{\partial^2 f(X_i;\theta)}{\partial\theta_h\partial\theta_j}}{f(X_i;\theta)} - \frac{\frac{\partial f(X_i;\theta)}{\partial\theta_h}\frac{\partial f(X_i;\theta)}{\partial\theta_j}}{f^2(X_i;\theta)}\right)_{\hat{\theta}^{(s-1)}}.$$

Then, the approach uses secure summation to calculate $\sum_{k=1}^{K} L_k$ and $\sum_{k=1}^{K} H_k$, but all that is needed to calculate the Newton-Raphson update is $[\sum_{k=1}^{K} H_k]^{-1}\sum_{k=1}^{K} L_k$. Thus, more information is shared than is necessary.[2]

A reasonable remedy uses the fact that matrix inversion amounts to solution of a system of linear equations. For simplicity, assume that $K = 2$.[3] So, to compute $Z = (H_1 + H_2)^{-1}(L_1 + L_2)$ we need to solve the linear system $(H_1 + H_2)Z = (L_1 + L_2)$.

[2]This same issue is noted in Karr et al. (2007) for regression: secure summation is used to compute X^TX and X^Ty, but all that is really needed is $(X^TX)^{-1}X^Ty$. Worse, a dishonest agency can exploit this by lying about its data.

[3]In fact, this is the most difficult case, because secure summation does not protect information when there are only two agencies.

Our protocol is as follows: agency A generates a $q \times q$ matrix M_1, which is of rank $\lfloor q/2 \rfloor$ and sends M_1 to agency B. Agency B then computes $M_1 H_2$ and $M_1 L_2$, sends them back to agency A, which can then produce the linear system

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

Symmetrically, agency B generates and sends to agency A a matrix M_2 that is of rank $\lfloor q/2 \rfloor$, and can then produce the linear system

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

Direct sharing of either $M_1(H_1 + H_2)$ of $M_1(L_1 + L_2)$ would divulge information. However, if T_1 and T_2 are full rank matrices generated by agencies A and B, respectively, then the systems

$$T_1 M_1(H_1 + H_2)Z = T_1 M_1(L_1 + L_2).$$

and

$$T_2 M_1(H_1 + H_2)X = T_2 M_1(L_1 + L_2).$$

can be combined into a system solvable for Z.

The degree of protection afforded by this protocol depends on the value of q: the larger the better.

There are also issues associated with the iterative nature of the Newton-Raphson algorithm. In particular, numerical inaccuracies associated with differentiation of the log-likelihood function, from matrix multiplication or from matrix inversion that arise at any agency affect computations at all agencies. These can be detected, if not circumvented, by having each agency calculate its proposed value for $\hat{\theta}^{(s)}$, and using secure summation and a secure Boolean operation to terminate the process if any agency's value differs too drastically from the mean.

3 Vertically Partitioned Data

Recall that the data are $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_i = (X_i^1, \cdots, X_i^p)$. In the *vertically partitioned* case, each agency k owns only portion of the variables, but for all n of the data points. Specifically, let $V_k \subset \{1, \ldots, p\}$ be the variables for agency k, and let $\mathbf{X}^k = \{X_i^p : i \in V_k\}$.

3.1 Independent Variables

When the sets $\{X^j : j \in V_1\}, \ldots, \{X^j : j \in V_K\}$ of agency-held variables are independent, maximum likelihood estimation is possible using secure summation. There are, however, at least two reasons why this case is of only limited interest. First, there is no known way securely to verify the independence assumption. And second even if there were, if variables are independent across agencies, the gain from sharing information is meager.

The most extreme case of independence is that the parameterization partitions across agencies: $f(X,\theta) = \prod_{k=1}^{K} f_k(\mathbf{X}^s;\theta_s)$. The log-likelihood function can be written as

$$\ell(\theta|\mathbf{X}) = \sum_{k=1}^{K}\left[\sum_{i=1}^{n}\log f_k(X_i^k;\theta_k)\right],$$

and each agency can calculate its own $\hat{\theta}_k$ and simply share its value.

When the set of variables are independent but the parameterization does not partition, i.e.,

$$f(X_i,\theta) = \prod_{k=1}^{K} f_k(X_i^k;\theta),$$

then

$$\frac{\partial\ell}{\partial\theta} = \sum_{k=1}^{K}\left[\sum_{i=1}^{n}\left(\frac{1}{f_k(X_i^k;\theta)}\frac{\partial f_k(X_i^k;\theta)}{\partial\theta}\right)\right],$$

which can be calculated using secure summation. The Hessian can be treated analogously, and the procedure in §2.2 applies.

3.2 Exponential Family

Next we consider the secure ML estimation for an exponential family model, but without the independence assumption. The likelihood function in still given by (2), and the ML estimator is

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

Assume for simplicity that there are two agencies A and B holding variables 1 and 2, respectively. In order to obtain $\hat{\theta}$, we need to compute $\sum_{i=1}^{n} t(X_i^1, X_i^2)$ securely. Our protocol is as follows: For each data record i,

Step 1 Agency A generates a vector W of length s, one component of which is X_i^1, and the other $s-1$ of which are random, and sends it to B.

Step 2 B computes $t(W_1, X_i^2), \ldots, t(W_s, X_i^2)$ generates a random value ε_i, and calculates $t(W_1, X_i^2) - \varepsilon_i, \ldots, t(W_s, X_i^2) - \varepsilon_i$.

Step 3 Agency A obtains $t(X_i^1, X_i^2) - \varepsilon_i$ from these using 1 out of s oblivious transfer (Di Crescenzo et al., 2000).

When this process is complete, A has $\sum_i [t(X_i^1, X_i^2) - \varepsilon_i]$ and B has $\sum_i \varepsilon_i$, which add to $\sum_{i=1}^{n} t(X_i^1, X_i^2)$.

The risks associated with protocol are, first, agency B's correctly guessing which element of W is X_i^1. The probability of this is $1/s$, which can be reduced using more complex versions of oblivious transfer (Du and Atallah, 2001). Second, agency A obtains $t(X_i^1, X_i^2) - \varepsilon_i$, which represents a risk unless ε_i is sufficiently random.

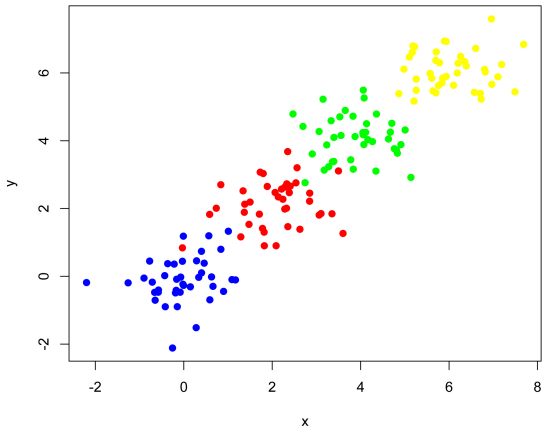

Figure 1: Heterogeneous data, with data points colored by agency.

3.3 General Case: Newton-Raphson

The protocol for exponential families described in §3.2 applies in this case as well.

4 Some (Research and Other) Issues

Here we discuss some of the aspects in secure analysis of distributed data that are not well understood.

4.1 The IID Assumption and Heterogeneous Data

Reverting to horizontally partitioned data, so far we have assumed that the data points, while held at different agencies, are independent and identically distributed (IID) As has been observed elsewhere, the rationale for combined analyses is not compelling in this case. Indeed, essentially all that is gained is the increase in precision resulting from a larger sampler size.

Figure 1 depicts a much more interesting case: the data are heterogeneous across agencies. Each of the four groups was generated through a bivariate normal distribution using the same covariance matrix, but with different means. None of these four groups by itself shows any linear structure, while the combined data clearly possess a linear structure.

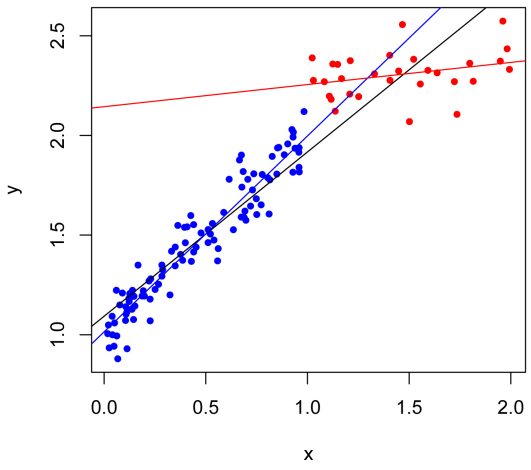

Figure 2: Regression example illustrating the need for opt-out strategies.

It is tempting to think that this can be dealt with simply replacing the model (1) by (recall that k indexes agencies)

$$\ell(\theta|\mathbf{X}) = \sum_{k=1}^{K} \sum_{i \in \boldsymbol{\ell}_k} f_k(X_i; \theta), \tag{6}$$

and proceeding as in §2. The problem is that models of local (within agencies) structure do not merge into a coherent model of global (across agencies) structure. Indeed, in a situation such as that in Figure 1, there do not exist techniques by which the agencies can even know that there is global structure. This is one clear research need.

5 Opt-Out Strategies

A significant disadvantage of privacy-preserving data mining tools in general is that an agency cannot know whether an analysis is too revealing of its data until the analysis has been performed and the results known to all agencies, at which point it is "too late."

To date, only *a priori* opt-out mechanisms are available, based for example on data set sizes (Karr et al., 2007; Sanil et al., 2009). Figure 2 illustrates. There, agency A has 100 blue points, with the corresponding blue regression line, agency B has 30 red points, and the red line is its regression line. The black line represents the linear regression for the combined data. Agency A could have opted out on the basis of having much more

data than agency B, but cannot know that its regression line is so close to the global one without having done the analysis. Related discussion appears in (Karr et al., 2005).

Good tools for characterizing data heterogeneity across agencies would be a key step in the right direction. Fisher information seems to hold some promise: it represents the right abstraction, and can be calculated securely, but the risks are not at all understood.

Acknowledgments

The research underlying this paper was supported by NSF grants EIA–0131884 and SES–0345441 to the National Institute of Statistical Sciences (NISS) and DMS–0112069 to the Statistical and Applied Mathematical Sciences Institute (SAMSI). Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Benaloh, J. (1987). Secret sharing homomorphisms: Keeping shares of a secret secret. In A. M. Odlyzko, ed., *CRYPTO86*, 251–260. Springer–Verlag. Lecture Notes in Computer Science No. 263.

Clifton, C., Kantarcioglu, M., Lin, X., j. Vaidya, and Zhu, M. (2003). Tools for privacy-preserving distributed data mining. *KDD Explorations*, 4(2):28–34.

Di Crescenzo, G., Malkin, T., and Ostrovsky, R. (2000). Single database private information retrieval implies oblivious transfer. *Lecture Notes in Computer Science*, 1807:122–138.

Du, W. and Atallah, M. J. (2001). Secure multi-problem computation problems and their applications: A review and open problems. Technical Report 2001–51, Center for Education and Research in Information Assurance and Security, Department of Computer Science, Purdue University, West Lafayette, IN.

Du, W. and Zhan, Z. (2002). A practical approach to solve secure multi-party computation problems. In *New Security Paradigms Workshop*, 127–135, New York. ACM Press.

Fienberg, S. E., Karr, A. F., Nardi, Y., and Slavkovic, A. (2007). Secure logistic regression with distributed databases. *Bull. Internat. Statist. Inst.* Presented at the 56th Session of the International Statistical Institute, Lisbon, August 2007.

Karr, A. F., Feng, J., Lin, X., Reiter, J. P., Sanil, A. P., and Young, S. S. (2005). Secure analysis of distributed chemical databases without data integration. *J. Computer-Aided Molecular Design*, November, 2005:1–9. Available on-line at `www.niss.org/dgii/technicalreports.html`.

Karr, A. F., Fulp, W. J., Lin, X., Reiter, J. P., Vera, F., and Young, S. S. (2007). Secure, privacy-preserving analysis of distributed databases. *Technometrics*, 49(3):335–345.

Lin, X., Clifton, C., and Zhu, Y. (2005). Privacy preserving clustering with distributed EM mixture modeling. *J. Knowledge and Information Syst.*, 8:68–81.

Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. In *Advances in Cryptology—Crypto2000, Lecture Notes in Computer Science, Volume 1880*, 20–24, New York. Springer–Verlag.

Sanil, A. P., Karr, A. F., Lin, X., and Reiter, J. P. (2009). Privacy preserving analysis of vertically partitioned data using secure matrix products. *J. Official Statist.*, 25(1):125–138.

Vaidya, J. and Clifton, C. (2004). Privacy-preserving data mining: Why, how and what for? *Security and Privacy Magazine*, 2(6):18–27.

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(1):50–57.

Yao, A. C. (1986). How to generate and exchange secrets. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, 162–167.

A Case study: Logistic Regression using Secure MLE

Logistic regression is one of the most commonly used classification technique in machine learning and data mining applications. The models posit parametric form for the distribution $P(Y|X)$, where Y represents class label and $X = (X^1, \cdots, X^p)$ are the explanatory valuables. Logistic regression directly estimates its parameters from the training data. When Y is a binary variable, the conditional distribution has parametric form

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

To estimate the parameters given the data $(X_1, Y_1), \cdots, (X_n, Y_n)$, the log-likelihood is

$$l = \sum_{i=1}^{n} Y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j X_i^j \right) - \log \left(1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_i^j) \right).$$

In order to apply the Newton-Raphson procedure, we use secure summation to compute

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} X_i^j \left[Y_i - \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_i^j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_i^j)} \right],$$

securely, where $X_i^0 = 1$ for all i and

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

Horizontally Partitioned Data. We can compute both the gradient and the Hessian matrix using the secure summation protocol, and perform the parameter updates. We can also use the alternative method there to compute $\hat{\theta}^{(s)} - \hat{\theta}^{(s-1)}$ directly.

Vertically Partitioned Data. In this case, we can use protocol from §3.3. See also Fienberg et al. (2007).