

Valid Statistical Analysis for Logistic Regression with Multiple Sources

Stephen E. Fienberg¹, Yuval Nardi¹, and Aleksandra B. Slavković²

¹ Carnegie Mellon University, Pittsburgh, PA 15213
{fienberg,yuval}@stat.cmu.edu

² Pennsylvania State University, University Park, PA 16802
sesa@stat.psu.edu

Abstract. Considerable effort has gone into understanding issues of privacy protection of individual information in single databases, and various solutions have been proposed depending on the nature of the data, the ways in which the database will be used and the precise nature of the privacy protection being offered. Once data are merged across sources, however, the nature of the problem becomes far more complex and a number of privacy issues arise for the linked individual files that go well beyond those that are considered with regard to the data within individual sources. In the paper, we propose an approach that gives full statistical analysis on the combined database without actually combining it. We focus mainly on logistic regression, but the method and tools described may be applied essentially to other statistical models as well.

Keywords: Distributed Databases, Horizontal Partitioned Data, Log-linear models, Privacy Preserving Data Mining, Secure Logistic Regression, Vertical Partitioned Data.

1 Introduction

Following the events of September 11, 2001, there has been heightened attention in the United States and elsewhere to the use of multiple government and private databases for the identification of possible perpetrators of future attacks, as well as an unprecedented expansion of federal government data mining activities, many involving databases containing personal information. There have also been claims that prospective data mining could be used to find the “signature” of terrorist cells embedded in larger networks. Fienberg [1,2] describes some proposals for the search of multiple databases which supposedly do not compromise possible pledges of confidentiality to the individuals whose data are included. One example is the concept of selective revelation associated with the now abandoned Total Information Awareness (TIA) security program.

Considerable effort has gone into understanding issues of privacy protection of individual information in single databases, and various statistical solutions have been proposed depending on the nature of the data, the ways in which the database will be used and the precise nature of the privacy protection being offered. Many data mining algorithms attempt to mine multiple distributed

databases and this was the goal for TIA. For an assessment of the role of datamining in terrorism prevention and related privacy issues see the recently-released report of the Committee on Technical and Privacy Dimensions of Information for Terrorism Prevention and Other National Goals at the National Research Council [3].

Once data are merged across sources, however, the nature of the confidentiality problem becomes far more complex and a number of privacy issues arise for the linked individual files that go well beyond those that are considered with regard to the data within individual sources. Mining for individual data or for identifiable groups of individuals is inherently disclosive and the privacy of those whose data are sought clearly cannot be protected! But such data mining can also compromise the data of others in the databases being searched. We have some hope of privacy protection when the goal is the production of the results of some statistical calculation, such as a regression analysis.

In working with multiple databases, we can conceptualize the existence of a single combined database containing all of the information for the individuals in the separate databases and for the union of the variables. In this paper, we propose an approach that gives full statistical analysis on this combined database without actually combining information sources. We focus mainly on logistic regression, but the method and tools described may be applied essentially to other statistical models as well. In Section 2, we briefly review the relevant privacy-preserving data mining (PPDM) and statistical disclosure limitation (SDL) literatures, state the general problem and provide an overview of binary logistic regression. Section 3 describes two protocols for secure logistic regression used when dealing with horizontally or vertically partitioned databases. We conclude by discussing our proposed protocols, privacy leakage problems and other ongoing work.

2 Background and Problem Formulation

Suppose that there are several parties collecting data separately and privately on overlapping sets of individuals or entities and involving overlapping sets of variables. We can use the designation ‘parties’ to stand for statistical or other government agencies, competing corporations, or any other organizations engaged in data collection. We assume that the parties desire to keep the information in their separate databases private and do not wish to share the databases with any other party. Each party is interested in performing some statistical analysis using the database in its possession in order to learn more about the underlying population from which it has drawn its database. Each party recognizes that the desired statistical analysis would enjoy greater statistical accuracy if it were able to carry out the relevant calculations using a hypothetical pooled (combined) database (made out of all of the parties’ databases). But data integration may lead to privacy breaches. Therefore, our goal is to establish new methods for analyzing the pooled data without actually combining the databases.

Privacy-preserving data mining (PPDM) is a class of algorithms used to extract (mine) information, but at the same time, maintain privacy (see [4,5]).

Emphasis is often on the algorithms rather than full statistical analyses. Related statistical disclosure limitation (SDL) techniques aim to preserve confidentiality but in contrast to PPDM techniques also aim to provide access to useful statistical data. The idea is that statistical inference should be the same whether one is using the original complete dataset, or an output dataset resulting from the original dataset and the SDL techniques.

Both the PPDM and SDL literatures have addressed problems related to partitioned databases. The technique used depends on how the database is partitioned. When the parties have exactly the same variables but for different data subjects, we call the situation (pure) *horizontally partitioned data*. At the other extreme, when the parties hold disjoint sets of attributes for the same data subjects we call the situation (pure) *vertically partitioned data*. These two pure cases have gained much attention recently. For results concerning horizontally partitioned data, see [6] (log-linear based logistic regression), [7] (adaptive regression splines), [8] (regression) and [9] (regression, data integration, contingency tables, maximum likelihood, Bayesian posterior distributions; regression for vertically partitioned data). Also see [10,11] for mining of association rules, and [12,13] for privacy-preserving support vector machine (SVM) classification, for both the horizontally and vertically partitioned data.

Sanil et al. [14,15] describe two different perspectives for performing linear regression on vertically partitioned databases. The work in [14] relies on quadratic optimization to solve for the regression coefficients $\hat{\beta}$ but it has two main problems. This method relies on the often unrealistic assumption that the party holding the response attribute is willing to share it with the other parties, and the method releases only limited diagnostic information. In [15] the authors use a form of secure matrix multiplication to calculate off-diagonal blocks of the full-data covariance matrix. An advantage of this approach is that rather complete diagnostic information can be obtained with no further loss of privacy. Analyses similar to ordinary regression (e.g., ridge regression) work in the same manner. Du and colleagues [16,17] describe similar, but less complete, approaches.

This work is related to the literature on secure multi-party computation (SMC). Over the past twenty years computer scientists have developed a number of efficient algorithms to securely evaluate a function whose inputs are distributed among several parties, known as secure multi-party computation protocols [18,19]. We make repeated use of these algorithms. Specifically, we use the *secure summation protocol*—a secure algorithm to compute a sum without sharing distributed inputs [20], and a *secure matrix multiplication*—a secure way to multiply two private matrices. Finally, we assume that the parties involved are *semi-honest*, i.e., (1) they follow the protocol and (2) they use their true data values. But parties may retain values from intermediate computations.

Logistic regression is a form of multivariate regression used for the analysis of binary outcomes. It is one of the most widely used statistical methods in biomedicine, genetics, the social sciences, business and marketing. It can be used to classify and predict, in a similar fashion to linear discriminant analysis, and is closely related to neural networks and support vector machines described in

data mining and machine learning literatures. In this paper, we draw from both PPDM and SDL paradigms, and address the problem of performing a “secure” logistic regression when the design matrix is distributed among multiple sources.

2.1 Partitioned Database Types

In this section we consider horizontally and vertically partitioned databases while more involved partitioning schemes are briefly discussed in Section 4. We assume that K parties (designated by A_1, \dots, A_K) with $K \geq 2$ are involved. Note, however, that the case with $K = 2$ is often trivial for security purposes. Horizontally partitioned data is the case in which agencies share the same fields but not the same individuals, or subjects. Assume the data consist of a design matrix X and a response vector Y , such that:

$$X = [X'_1, X'_2, \dots, X'_K]' \text{ and } Y = [Y'_1, Y'_2, \dots, Y'_K]', \quad (1)$$

where apostrophe ($'$) denotes transpose. Here, X_k , for $k = 1, \dots, K$, is the database held privately by party A_k , and Y_k is its vector of responses. We let n_k denote the number of individuals that belong to party A_k , and let $N = \sum_{k=1}^K n_k$ be the overall sample size. Each X_k is an $n_k \times p$ matrix and we assume that the first column of each X_k matrix is a column of 1's. We will refer to X and Y as the “global” predictor matrix and the “global” response vector respectively. For horizontally partitioned databases it is assumed that parties all have the same variables, and that no parties share observations. Also, the attributes need to be in the same order.

In vertically partitioned data, parties all have the same subjects, but different attributes. Assume the data look like the following:

$$[Y, X] = [Y, X_1, \dots, X_K], \quad (2)$$

where X_k is the matrix of a distinct number of independent variables on all N subjects, and Y is the vector of responses. We assume that Y is held by only one party, say party A_1 . We let p_k be the number of variables for party A_k , $k = 1, \dots, K$. Note that each X_k is an $N \times p_k$ matrix (except for X_1 , which is an $N \times (1 + p_1)$) and we assume that the first column of the X_1 matrix is a column of 1's. For vertically partitioned databases it is further assumed that parties all have the same observations, and that no parties share variables. In order to match up a vertically partitioned database, all parties must have a global identifier, such as social security number.

2.2 Logistic Regression

Researchers and data analysts use binary logistic regression for modeling binary outcomes, for example, to predict a membership in a group, e.g., does a person have a high-risk credit score or a low-risk credit score given her payment history, income, and gender?

Let Y_1, \dots, Y_N be independent Bernoulli variables whose means $\pi_i = E(Y_i)$, depend on some covariates $x_i \in \mathbb{R}^{p+1}$, through the relationship

$$\text{logit}(\pi_i) = \sum_{j=0}^p x_{ij}\beta_j = x_i'\beta, \quad (3)$$

where $\text{logit}(\pi) = \log[\pi/(1 - \pi)]$, and the x_i 's make up the N rows of the design matrix X , whose first column is unity.

In logistic regression, the vector of coefficients, or β , is of interest. Since we cannot compute the estimate of β in closed form, we traditionally use Newton-Raphson or a related iterative method (see [21]), to find a value of β that maximizes the log-likelihood:

$$l(\beta) = \sum_j \left(\sum_i y_i x_{ij} \right) \beta_j - \sum_i n_i \log \left[1 + \exp(x_i' \beta) \right]. \quad (4)$$

Here j runs over the set of predictors, and i runs over the different number of “settings” of the covariates. The number of observations corresponding to setting x_i is denoted by n_i (note that it is different than n_k in Section 2.1). In case of continuous predictors, we have $n_i = 1$.

At each iteration of Newton-Raphson algorithm, we calculate the new estimate of $\hat{\beta}$ by

$$\hat{\beta}^{(s+1)} = \hat{\beta}^{(s)} + (X'W^{(s)}X)^{-1}X'(y - \mu^{(s)}), \quad (5)$$

where $W^{(s)} = \text{diag}(n_i \pi_i^{(s)}(1 - \pi_i^{(s)}))$, $\mu_i^{(s)} = n_i \pi_i^{(s)}$ and $\pi_i^{(s)}$ is the probability of a “success” for the i^{th} observation in the iteration s , $i = 1, \dots, N$. The algorithm stops when the estimate converges. Note that we require an initial estimate of $\hat{\beta}$. For the complete statistical analysis, finding the coefficients of a regression equation is not sufficient; we need to know whether the model has a reasonable fit to the data. We typically look at the residuals and fit statistics such as Pearson’s χ^2 and the likelihood-ratio deviance statistics.

Next we describe how to use secure matrix sharing techniques and apply them to the logistic regression setting over distributed databases.

3 Secure Logistic Regression

Fienberg et al. [6] described “secure” logistic regression for horizontally partitioned databases when all variables are categorical. They discuss the advantages of the log-linear approach versus the regression approach in the fully categorical case where the minimal sufficient statistics are marginal totals and logistic regression is effectively equivalent to log-linear model analysis (e.g., see [21,22]).

Here we focus on binary logistic regression in the case of horizontally and vertically partitioned databases but with *quantitative* covariates using secure multi-party computation. We draw from [6] for the horizontal case presented here and suggest necessary modifications. We continue to work on the problem of vertically partitioned data in the fully categorical data setting but do not report on any results here.

3.1 Logistic Regression over Horizontally Partitioned Data

We now turn to a general approach for logistic regression for a horizontally partitioned database using ideas from secure regression (e.g. see [8]). In ordinary linear regression, the estimate of the vector of coefficients is

$$\hat{\beta} = (X'X)^{-1}X'y. \quad (6)$$

To find the global $\hat{\beta}$ vector, party A_k calculates their own $X'_k X_k$ and $X'_k y_k$ matrices. The sum of these respective matrices are the global $X'X$ and $X'y$ matrices. Since direct sharing of these matrices results in full disclosure, the parties need to employ some other method such as secure summation to preserve privacy. In this secure summation process, the first party adds a random matrix to its data matrix. The remaining agencies add their raw data to the updated matrix until in the last step the first party subtracts their added random values and shares the global matrices.

Next we apply the secure summation approach to the logistic regression analysis, and implement the secure Newton-Raphson algorithm. We can choose an initial estimate for the Newton-Raphson procedure in two ways: (i) the parties can discuss and share an initial estimate of the coefficients, or (ii) we can compute initial estimates using ordinary linear regression of the responses and predictors using secure regression computations. In order to update β , we need the parts shown in (5). We can break the last term on the right-hand side into two parts: the $(X'W^{(s)}X)^{-1}$ matrix and the $X'(y - \mu^{(s)})$ matrix. At each iteration of Newton-Raphson, we update the π vector, and thus update the W matrix and the vector μ . It follows that

$$X'W^{(s)}X = (X_1)'(W_1)^{(s)}X_1 + (X_2)'W_2^{(s)}X_2 + \cdots + (X_K)'W_K^{(s)}X_K, \quad (7)$$

$$X'(y - \mu^{(s)}) = X_1(y_1 - \mu_1^{(s)}) + X_2(y_2 - \mu_2^{(s)}) + \cdots + X_K(y_K - \mu_K^{(s)}), \quad (8)$$

where $\mu_k^{(s)}$ is the vector of $(n_k)_l(\hat{\pi}_k)_l$ values and $W_k^{(s)} = \text{diag}((n_k)_l(\hat{\pi}_k)_l(1 - (\hat{\pi}_k)_l))$ for party k , $k = 1, \dots, K$, $l = 1, \dots, n_k$ and for iteration, s . Note, however, since we are dealing here with only continuous explanatory variables that $n_k = (n_k)_l$. Then for each iteration of Newton-Raphson, we find the new estimate of β by using secure summation.

One major drawback of this method is that we have to perform secure matrix sharing for every iteration of the algorithm; every time it runs, we also have to share the old $\hat{\beta}$ vector with all of the parties so they may calculate their individual pieces. When all variables are categorical, this method involves more computation than using the log-linear model approach to logistic regression, where only the relevant marginal totals must be shared (once) among the parties. In the more general setting, we also have no simple way to check on potential disclosure of individual level data and thus we are providing security only for the parties and not necessarily for the individuals in their databases, e.g., see discussion in [8] for the linear regression secure computation problem.

Diagnostics. One way to assess fit is to use various forms of model diagnostics such as residuals, but this potentially increases the risk of disclosure. As Fienberg et al. [6] proposed in their log-linear model approach, we can compare log-likelihood functions of the larger model and the more parsimonious model. We can rewrite the log-likelihood equation from (4) in terms of the K parties and use secure summation to find this value

$$\sum_{k=1}^K \sum_{j=1}^{n_k} \{(y_k)_j \log((\pi_k)_j) + (1 - (y_k)_j) \log(1 - (\pi_k)_j)\}, \quad (9)$$

as well the Pearson's χ^2 or likelihood-ratio deviance statistics:

$$X^2 = \sum_{k=1}^K \sum_{j=1}^{n_k} \left(\frac{(y_k)_j - (n_k)_j (\pi_k)_j}{\sqrt{(n_k)_j (\pi_k)_j (1 - (\pi_k)_j)}} \right)^2 \quad (10)$$

and

$$G^2 = 2 \sum_{k=1}^K \sum_{j=1}^{n_k} \left[(y_k)_j \log \left(\frac{(y_k)_j}{(\hat{\mu}_k)_j} \right) + ((n_k)_j - (y_k)_j) \log \left(\frac{(n_k)_j - (y_k)_j}{(n_k)_j - (\hat{\mu}_k)_j} \right) \right]. \quad (11)$$

If the change in the likelihood is large with respect to a chi-square statistic with appropriate degrees of freedom, we can reject the null hypothesis and conclude that the simpler model provides a better fit to the data.

3.2 Logistic Regression over Vertically Partitioned Data

For vertically partitioned data held by K parties, (A_1, \dots, A_K) , we have $X = [X_1, X_2, \dots, X_K]$, where each X_k is an $N \times p_k$ matrix, except for X_1 , which has $1 + p_1$ columns (one for the intercept). The parameter β has a similar block structure. Thus we can rewrite equation (3) as

$$\text{logit}(\pi_i) = \sum_{k=1}^K x'_{i,k} \beta_k, \quad (12)$$

where $x_{i,k}$ are the measurements of record i restricted to the set of variables held by party A_k , and $\beta = (\beta_1, \dots, \beta_K)$. This additivity across parties is crucial. Indeed, virtually all of the work noted in Section 3.1 for horizontally partitioned data depends on “anonymous” sharing of analysis-specific sufficient statistics that add over the parties.

We can now write the log-likelihood function, up to an additive constant, as

$$l(\beta) = y' \left(\sum_{k=1}^K X_k \beta_k \right) - \sum_{i=1}^N \log \left[1 + \exp \left\{ \sum_{k=1}^K x'_{i,k} \beta_k \right\} \right]. \quad (13)$$

We must obtain the maximum likelihood estimator $\hat{\beta}$ of β through an iterative procedure like before. We show below how to implement a secure Newton-Raphson algorithm to find roots of the likelihood equations for the vertically

partitioned case. Karr et al. [9] describe a similar approach to numerical maximization of likelihood functions for horizontally partitioned data.

To estimate $\beta = (\beta_1, \dots, \beta_K)$, at each iteration of the Newton-Raphson algorithm, we calculate the new estimate of $\hat{\beta}$ by

$$\hat{\beta}^{(s+1)} = \hat{\beta}^{(s)} + (X'W^{(s)}X)^{-1}X'y^{(s)}, \quad (14)$$

where $W^{(s)} = \text{diag}(\pi_i^{(s)}(1 - \pi_i^{(s)}))$, $\pi_i^{(s)} = (1 + \exp\{-x'_i\beta^{(s)}\})^{-1}$, and $y^{(s)} = y - \pi^{(s)}$ (see (5)).

In both horizontal and vertical settings thus far we have assumed that the parties are semi-honest, i.e., they follow the protocol but may retain intermediate values, and use them to gain more information. We now outline an additional set of assumptions required for the vertical partitioning scheme. We assume that (a) some ordering of subjects has been performed, (b) party A_1 holds the response variable y , and is not willing to share it (as usually happens in practice), (c) parties are not willing to share intermediate values of their components of $\hat{\beta}$, except for the convergent estimated parameter (see below for a possible security breach if they *are* willing), and (d) parties *are* willing to share some ‘summary’ statistics (see below). Note that a protocol is not private according to *privacy by simulation* (see, e.g., [23], [24]) if a participating party may learn more information using any intermediate values, than it could have learned based on its input and output only! A way around this problem is usually achieved by decrypting the intermediate values in such a way that parties learn only *random shares* of the values; a random share of an output O is a set of random outputs O_1, \dots, O_K , such that O_j is kept hidden from the other parties, and such that $O = O_1 + \dots + O_K$. This idea relies on a result, coming from the secure multiparty computation community, that there exists a secure protocol for any probabilistic polynomial-time functional (see [25], [26]). These generic protocols are computationally inefficient unless the size of the problem is relatively small. We are currently working on designing specific protocols for our problem.

We now describe a protocol that parties can follow to update $\hat{\beta}$, using equation (14), in a secure way. We define $I = X'y^{(s)}$ and $II = (X'W^{(s)}X)$. Then

- Party A_k picks an initial value $\beta_k^{(0)}$, $k = 1, \dots, K$.
- Parties obtain $\pi_i^{(s)}$ by applying a K -party secure summation to $x'_i\beta^{(s)} = \sum_{k=1}^K x'_{i,k}\beta_k^{(s)}$.
- Write $I = (I_1, \dots, I_K)'$. Party A_k , for every $k = 2, \dots, K$, obtains $I_k = X'_k y^{(s)}$ by applying secure inner product to $X'_k y$ (note that this is done only once). The calculation of I_1 requires no secure protocol since the response y is assumed to be held by party A_1 . At the end of this step each party A_k holds privately I_k . The interactions are pairwise between party A_1 , and party A_k , for $k = 2, \dots, K$.
- Parties apply secure matrix product to off-diagonal sub-matrices of II . Diagonal sub-matrices, $X'_k W^{(s)} X_k$, are computed locally by each party, and they then share the results with the other parties (this sharing has to be done in every iteration). At the end of this step each party gets to learn II .

- Each party can now invert II . Suppose

$$II^{-1} = \begin{pmatrix} A_{11}^{(s)} & A_{12}^{(s)} & \cdots & A_{1K}^{(s)} \\ A_{21}^{(s)} & A_{22}^{(s)} & \cdots & A_{2K}^{(s)} \\ \vdots & \vdots & \cdots & \vdots \\ A_{K1}^{(s)} & A_{K2}^{(s)} & \cdots & A_{KK}^{(s)} \end{pmatrix},$$

for suitable matrices $A_{jk}^{(s)}$. Then, to work with equation (14), party A_j obtains

$$\sum_{k=1}^K A_{jk}^{(s)} I_k.$$

using secure summation by initiating the protocol, and *never sharing the result*.

- Each party updates its own share of the estimated parameter, $\hat{\beta}$.

A Possible Privacy Breach. Trying to relax the assumption that the agencies are unwilling to share intermediate values of their components of $\hat{\beta}$ may lead to serious privacy breaches. To see this, suppose now that the agencies *are* willing to share the intermediate values of $\hat{\beta}$. In order to update $\hat{\beta}$ we compute $x'_i \hat{\beta}^{(s)}$, for each $i = 1, \dots, n$. Suppressing dependence on i , we write

$$x' = x'_i = (x'_1, x'_2, \dots, x'_K) \in \mathbb{R}^p.$$

The computation of $x' \hat{\beta}^{(s)}$ involves the secure sum and given any fixed s reveals nothing but the sum. However, *enough* iterations may lead to full privacy leakage. To see this, suppose that we iterate p times. Every agency gets to learn $\hat{\beta}^{(s)}$ for $s = 1, \dots, p$, and

$$\hat{a}^{(s)} := x' \hat{\beta}^{(s)}, \quad s = 1, \dots, p.$$

Therefore, they can form the following system of linear equations:

$$x' \hat{\beta} = \hat{a}', \tag{15}$$

where $\hat{\beta} = [\hat{\beta}^{(1)}, \dots, \hat{\beta}^{(p)}]$ is a $p \times p$ matrix, and $\hat{a}' = (\hat{a}^{(1)}, \dots, \hat{a}^{(p)})$. If, and this is the key, $\{\hat{\beta}^{(1)}, \dots, \hat{\beta}^{(p)}\}$ are linearly independent, one may solve (15)

$$x = (\hat{\beta}^{-1})' \hat{a}.$$

Therefore, one iteration may not reveal private information, but enough iterations may. Similar concerns may appear in the calculation of $X' y^{(s)}$.

4 Discussion and Future Directions

What are the disclosure risks with respect to distributed databases? In this setting, one goal is to perform the analysis on the unaltered data, by anonymously

sharing sufficient statistics rather than the actual data. To perform secure logistic regression with continuous predictors in the vertical case, however, one requires unique record identifiers common to all the databases. Such identifiers alone do not constitute identity disclosures, because if one shares them they do not necessarily link to associated attribute values. Nonetheless, the parties must be willing to share some intermediate estimates of the components of regression coefficients which may unintentionally reveal some identifying information (see Section 3.2). Secure logistic regression in the vertical case also poses attribute disclosure risks: if the analysis reveals that attributes held by party A predict those held by party B, then A gains knowledge of attributes held by B. This is equally true even for linear regression on pure vertically partitioned data, e.g., see [14].

For the horizontal case, there is no simple way of checking for individual disclosure risk (see Section 3.1). In the full categorical horizontal case, with the “secure” log-linear approach to logistic regression [6], the parties must only perform one round of secure summation to compute the relevant sufficient statistics. The “secure” logistic regression protocol is thus computationally more intensive than the log-linear method since the parties must perform a secure summation for each Newton-Raphson iteration. In the full quantitative horizontal case we cannot apply the log-linear approach.

A preliminary analysis indicates that “secure” Newton-Raphson protocol for logistic regression will have a different computational performance given the two partition types. The total computation time of the vertical case is strongly dependent on the number of parties. In contrast to the horizontal case, the vertical case must use secure matrix products to compute the off-diagonal block elements of the covariance matrix. The secure matrix product protocol requires a QR decomposition to mitigate leakage. This is a fairly expensive calculation, and we expect the total computation time for the vertically partitioned data set to increase roughly as $O(N^2)$. We are currently exploring the efficiency of our protocol for both the horizontal and vertical case.

The “secure” Newton-Raphson implementation needs further investigation. Each new iteration of the algorithm may present a leakage situation since secure matrix operations are not disclosure-free; e.g., two parties may each relinquish information to the other such as vectors that are orthogonal to their respective databases [15].

Beyond the Two Pure Cases. There may be computational reasons to consider horizontal and vertical cases separately, as well as separating cases with only categorical and only continuous predictors. Real world data are likely to be much more complex. Reiter et al. [27], and Fienberg et al. [28] describe a more general case, lying in between the two pure data partitioning schemes. It is referred to as *vertically partitioned, partially overlapping database*, where attributes are partitioned (vertically) among parties, but not every data record is common to all parties. Table 1 shows an example of a vertically partitioned, partially overlapping database where the database X is decomposed similarly as in the vertical partitioning scheme, but each sub-block matrix X_k contains missing values (for those records that party A_k does not have in its possession).

Table 1. Schematic representation of a vertically partitioned, partially overlapping data. Party *A* records values of x_1, x_2 and observations 1 – 2000, 3001 – 4000, Party *B* records values of x_3, x_4, x_5 of observations 1 – 1000, 2001 – 3000, and Party *C* records values of x_6, x_7 of observations 1001 – 3000.

n	Party A		Party B			Party C	
	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	✓	✓	✓	✓	✓	•	•
...	✓	✓	✓	✓	✓	•	•
1000	✓	✓	✓	✓	✓	•	•
1001...2000	✓	✓	•	•	•	✓	✓
2001...3000	•	•	✓	✓	✓	✓	✓
3001...4000	✓	✓	•	•	•	•	•

Looking at Table 1, one is naturally led to think that the vertical partitioning, partially overlapping case may be cast as a missing data problem. We are currently working on developing methods for logistic regression on vertically partitioned, partially overlapping data. In particular, we consider two cleanly identifiable cases, involving solely continuous predictors or solely categorical ones. For the continuous case we assume that X follows a Gaussian mixture model (GMM), whereas for the categorical case, we assume a Multinomial mixture model (MMM). We can apply well known approaches for dealing with missing values such as the EM algorithm. In particular, we develop a “secure” (double) EM algorithm where the “double” here stands for the two types of incompleteness; the first has to do with the mixture parameters and the GMM (or MMM) parameters, while the second captures the actual missing covariates. We can obtain parameter estimates by applying the “secure” (double) EM algorithm in conjunction with secure multi-party protocols.

5 Conclusion

There are many scientific and business settings which require statistical analyses that “integrate” data stored in multiple distributed databases. Unfortunately, barriers exist that prevent simple integration of the databases. In many cases, the owners of the distributed databases are bound by confidentiality to their data subjects, and cannot allow database access to outsiders.

We have outlined an approach to carry out “valid” statistical analysis for logistic regression with quantitative covariates on both horizontally and vertically partitioned databases that does not require actually integrating the data. This allows parties to perform analyses on the global database while preventing exposure of details that are beyond those used in the joint computation.

We are currently developing a log-linear model approach for strictly vertically partitioned databases and a more general secure logistic regression for problems involving partially overlapping databases with measurement error.

Acknowledgments. The research reported here was supported in part by NSF grants EIA9876619 and IIS0131884 to the National Institute of Statistical Sciences, by NSF Grant SES-0532407 to the Department of Statistics, Penn State University, and by Army contract DAAD19-02-1-3-0389 to CyLab at Carnegie Mellon University.

References

1. Fienberg, S.: Privacy and confidentiality in an e-commerce world: Data mining, data warehousing, matching and disclosure limitation. *Statistical Science* 21, 143–154 (2006)
2. Fienberg, S.: Data mining, privacy, disclosure limitation, and the hunt for terrorists. In: Chen, H., Reid, E., Sinai, J., Silke, A., Ganor, B. (eds.) *Terrorism Informatics: Knowledge Management and Data Mining for Homeland Security*. Springer, New York (2008)
3. Committee on Technical and Privacy Dimensions of Information for Terrorism Prevention and Other National Goals: Protecting Individual Privacy in the Struggle Against Terrorists: A Framework for Assessment. National Academy Press, Washington (2008)
4. Agrawal, R., Srikant, R.: Privacy preserving data mining. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas (2000)
5. Clifton, C., Vaidya, J., Zhu, M.: *Privacy Preserving Data Mining*. Springer, New York (2006)
6. Fienberg, S., Fulp, W., Slavkovic, A., Wrobel, T.: “Secure” log-linear and logistic regression analysis of distributed databases. In: Domingo-Ferrer, J., Franconi, L. (eds.) *PSD 2006*. LNCS, vol. 4302, pp. 277–290. Springer, Heidelberg (2006)
7. Ghosh, J., Reiter, J., Karr, A.: Secure computation with horizontally partitioned data using adaptive regression splines. *Computational Statistics and Data Analysis* (2006) (to appear)
8. Karr, A., Lin, X., Reiter, J., Sanil, A.: Secure regression on distributed databases. *Journal of Computational and Graphical Statistics* 14(2), 263–279 (2005)
9. Karr, A., Fulp, W., Lin, X., Reiter, J., Vera, F., Young, S.: Secure, privacy-preserving analysis of distributed databases. *Technometrics* (2007) (to appear)
10. Kantarcioglu, M., Clifton, C.: Privacy preserving data mining of association rules on horizontally partitioned data. *Transaction of Knowledge and Data Engineering* 16, 1026–1037 (2004)
11. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada (2002)
12. Yu, H., Jiang, X., Vaidya, J.: Privacy preserving svm using nonlinear kernels in horizontally partitioned data. In: *Proc. of ACM SAC Conference Data Mining Track* (2006)
13. Yu, H., Vaidya, J., Jiang, X.: Privacy-preserving svm classification on vertically partitioned data. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) *PAKDD 2006*. LNCS (LNAI), vol. 3918, pp. 647–656. Springer, Heidelberg (2006)
14. Sanil, A., Karr, A., Lin, X., Reiter, J.: Privacy preserving regression modelling via distributed computation. In: *Proc. Tenth ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining*, pp. 677–682 (2004)

15. Sanil, A., Karr, A., Lin, X., Reiter, J.: Privacy preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics* (2007); Revised manuscript under review (2007)
16. Du, W., Zhan, Z.: A practical approach to solve secure multi-party computation problems. In: *New Security Paradigms Workshop*, pp. 127–135. ACM Press, New York (2002)
17. Du, W., Han, Y., Chen, S.: Privacy-preserving multivariate statistical analysis: Linear regression and classification. In: *Proceedings of the 4th SIAM International Conference on Data Mining*, pp. 222–233 (2004)
18. Goldwasser, S.: Multi-party computations: Past and present. In: *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pp. 1–6. ACM Press, New York (1997)
19. Yao, A.: Protocols for secure computations. In: *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 160–164. ACM Press, New York (1982)
20. Benaloh, J.: Secret sharing homomorphisms: Keeping shares of a secret secret. In: *Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263*, pp. 251–260. Springer, Heidelberg (1987)
21. Agresti, A.: *Categorical Data Analysis*, 2nd edn. Wiley, New York (2002)
22. Bishop, Y., Fienberg, S., Holland, P.: *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, Cambridge (1975); Reprinted by Springer (2007)
23. Lindell, Y., Pinkas, B.: Privacy preserving data mining. *J. Cryptology* 15(3), 177–206 (2002)
24. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality* (2008) (to appear)
25. Yao, A.C.: How to generate and exchange secrets. In: *Proceedings of the 27th Symposium on Foundations of Computer Science (FOCS)*, pp. 162–167. IEEE, Los Alamitos (1986)
26. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game - a completeness theorem for protocols with honest majority. In: *Proceedings of the 19th annual Symposium on the Theory of Computing (STOC)*, pp. 218–229. ACM, New York (1987)
27. Reiter, J., Karr, A., Kohlen, C., Lin, X., Sanil, A.: Secure regression for vertically partitioned, partially overlapping data. In: *Proceedings of the American Statistical Association* (2004)
28. Fienberg, S., Karr, A., Nardi, Y., Slavkovic, A.: Secure logistic regression with distributed databases. In: *Proceedings of the 56th Session of the ISI, The Bulletin of the International Statistical Institute* (2007)