Secure Logistic Regression with Multi-Party Distributed Databases

Stephen E. Fienberg[1], Alan F. Karr[2], Yuval Nardi[1], Aleksandra B. Slavković[3]
Carnegie Mellon University[1]
National Institute of Statistical Sciences[2]
Pennsylvania State University[3]

Abstract

We consider the problem of performing a logistic regression in a secure way across partially overlapping data bases, owned by multiple parties. The idea is to have a shared computation for the purposes of model fitting and assessment using the pooled data, in such a fashion that no party's data is directly disclosed to any other party. We describe the relationship of this problem to other well-formulated statistical ones such as analysis of missing data. We then discuss the computational details, the meaning of secure computation in this context, and the implications for the protection of privacy, both for data base owners and for the individuals whose data are incorporated into the calculations.

KEY WORDS: Logistic Regression, Privacy-Preserving Data Mining, Secure Multi-Party Computation, EM.

1. Introduction and Background

We address the problem of performing a logistic regression in a secure way on pooled data collected separately by several parties (agencies) without actually directly combining their databases. Specifically, the parties want to fit a model and make inferences using the pooled data in a way that no party's data is disclosed to any other party. The partitioning of data may occur in several ways. 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.* Examples of parties are government agencies or competing business establishments.

In this paper, we consider a structured mixed case. Attributes are partitioned among parties, but not every data record is common to all parties. There is a horizontal partitioning component as well: one party may be the only one holding any attributes for some records. The assumption that attributes are partitioned is central: it avoids reconciling possibly different attribute values in multiple databases. But, it eliminates potentially important cases such as "party A holds attributes 1, 2, and 3 for some of its subjects and 4, 5, and 6 for others, while the opposite is true for party B." The analysis we consider is logistic regression. In the case of categorical predictor variables we can carry out the analysis directly on the logistical scale or indirectly via the examination of corresponding log-linear models. Each strategy offers distinct advantages in specific instances, cf. Bishop et al. (1975. Reprinted, Springer-Verlag, New York, 2007.), Fienberg (1980. Reprinted, Springer-Verlag, New York, 2007.), and the application for horizontal partitioning in Fienberg et al. (2006).

In the computer science literature, the problems we address are termed *privacy-preserving data mining* (PPDM). Often the emphasis is on algorithms rather than full statistical analyses. Examples include the results of the application of association rules and K-means clustering. For details, see Clifton et al. (2006) and for a partial explanation why these are not the same as shared secure computation, see Fienberg et al. (2006).

The PPDM literature tends to focus on either the horizontal or the vertical partitioned cases. For results concerning horizontally partitioned data, see Fienberg et al. (2006), Ghosh et al. (2006) (adaptive regression splines), Karr et al. (2005) (regression) and Karr et al. (2007) (regression, data integration, contingency tables, maximum likelihood, Bayesian posterior distributions; regression for vertically partitioned data).

Sanil et al. (2004) and Sanil et al. (2007) treat *secure linear regression for vertically partitioned data* from two very different perspectives. Under the often unrealistic assumption that the agency holding the response attribute is willing to share it with the other agencies, Sanil et al. (2004) apply the algorithm for derivative-free quadratic optimization due to (Powell, 1964) to solve the least squares minimization problem directly, yielding the estimated coefficients $\hat{\beta}$. Only limited diagnostic information is available, however. Sanil et al. (2007) use a form of secure matrix multiplication to calculate off-diagonal blocks of the full-data covariance matrix. These calculations occur pairwise between agencies, and *do* entail loss of information. In Sanil et al. (2007), it is shown how to minimize this loss. 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 Zhan (2002) and Du et al. (2004) describe similar, but less complete, approaches.

We do not emphasize data pre-processing in this paper, but the issues are complex. *Measurement error* creates problems of record linkage and resolution of the quantities to be used in a calculation. To do secure computation in the sense we describe we require that all data to be in the same units, and records must be linked unambiguously. Determining which records are common to all parties without unnecessary revelation of information is itself a daunting challenge.

The full implementation of our approach will be described in a longer paper. Here we provide background and methodological details, especially those linking our work to the more traditional statistical and machine learning literature, and we briefly describe a practical application.

2. Logistic Regression for Vertically Partitioned Data Bases

Let $Y_1, \ldots, Y_n$ 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

$$\operatorname{logit}(\pi_i) = \sum_{j=0}^{p} x_{ij}\beta_j = (X\beta)_i \ , \tag{1}$$

where $\operatorname{logit}(\pi) = \log[\pi/(1-\pi)]$, X is the associated $n \times (p+1)$ design matrix whose first column is unity, and $(a)_i$ stands for the i-th element of the vector a.

For vertically partitioned data held by K parties, we have $X = [X_1, X_2, \ldots, X_K]$, 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 (1) as

$$\operatorname{logit}(\pi_i) = \sum_{k=1}^{K} (X_k\beta_k)_i \ . \tag{2}$$

This additivity across parties is crucial. Indeed, virtually all of the work noted in §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

$$l(\beta) = y^t \left(\sum_{k=1}^{K} X_k\beta_k \right) - \sum_{i=1}^{n} \log \left[1 + \exp \left\{ \sum_{k=1}^{K} (X_k\beta_k)_i \right\} \right] \ . \tag{3}$$

Here, the superscript t denotes matrix transpose.

We must obtain the maximum likelihood estimator $\hat{\beta}$ of β through an iterative procedure. We show below how to implement a secure Newton-Raphson algorithm to find roots of the likelihood equations. Karr et al. (2007) describe a similar approach to numerical maximization of likelihood functions for horizontally partitioned data. For simplicity of presentation, we focus on $K = 2$, and remark, at the end, on how to generalize to a multi-party scenario.

Let $X = [U, V]$, and $\beta = [\alpha, \gamma]$. Here, U is an $n \times (1 + p_1)$ matrix, and V is an $n \times p_2$ matrix, with $p_1 + p_2 = p$. Also, letting x_i^t denote the rows of X, we write $x_i^t = (u_i^t, v_i^t)$, for $i = 1, \ldots, n$. Let π denote the n-vector whose elements are π_i, $i = 1, \ldots, n$. Differentiating the log-likelihood with respect to α and γ, we obtain the gradient $\nabla_l(\beta) = (l_\alpha(\beta), l_\gamma(\beta))$, where

$$l_\alpha(\beta) \ = \ U^t y - \sum_{i=1}^{n} u_i\pi_i = U^t(y - \pi) \ , \tag{4}$$

$$l_\gamma(\beta) \ = \ V^t y - \sum_{i=1}^{n} v_i\pi_i = V^t(y - \pi) \ . \tag{5}$$

The derivation of the gradient follows from the fact that $\partial(U\alpha)_i/\partial\alpha = u_i$, $\partial(V\gamma)_i/\partial\gamma = v_i$ and

$$\pi_i \equiv \pi_i(U, V) = \frac{\exp\{(U\alpha)_i + (V\gamma)_i\}}{1 + \exp\{(U\alpha)_i + (V\gamma)_i\}} \ . \tag{6}$$

Note that π depends on the full data $X = [U, V]$, and thus cannot be calculated locally by any party involved.

The Hessian $H_l(\beta)$ is the matrix with sub-block matrices $l_{\alpha\alpha}(\beta), l_{\alpha\gamma}(\beta), l_{\gamma\alpha}(\beta), l_{\gamma\gamma}(\beta)$, given by

$$\begin{array}{lcllcl} l_{\alpha\alpha}(\beta) & = & -U^t D_\pi U & \quad l_{\alpha\gamma}(\beta) & = & -V^t D_\pi U \\ l_{\gamma\alpha}(\beta) & = & -U^t D_\pi V & \quad l_{\gamma\gamma}(\beta) & = & -V^t D_\pi V \end{array} \tag{7}$$

for a diagonal matrix $D_\pi = \operatorname{diag}\{\pi_i(1-\pi_i)\}$. The Hessian results from a direct differentiation of the gradient and uses the following relationships: $\partial\pi_i(X)/\partial\alpha = u_i^t\pi_i(1-\pi_i)$, and $\partial\pi_i(X)/\partial\gamma = v_i^t\pi_i(1-\pi_i)$. The Newton-Raphson algorithm updates an old (current) value of the parameter, $\hat{\beta}_{\mathrm{OLD}}$, via

$$\hat{\beta}_{\mathrm{NEW}} = \hat{\beta}_{\mathrm{OLD}} - H_l^{-1}(\hat{\beta}_{\mathrm{OLD}})\nabla_l(\hat{\beta}_{\mathrm{OLD}}) \ . \tag{8}$$

Over the past twenty years, computer scientists developed a number of efficient algorithms to securely evaluate a function whose inputs are distributed among several parties, known as secure multi-party computation (SMC) protocols (Goldwasser, 1997; Yao, 1982). Specifically, we will be using the *secure summation protocol*—a secure algorithm to compute a sum without sharing distributed inputs (Benaloh, 1987), and a *secure matrix multiplication*—a secure way to multiply two private matrices. 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.

The first party, holding design matrix U, picks an initial choice $\alpha^{(0)}$. Likewise, the second party, holding design matrix V, picks an initial choice $\gamma^{(0)}$. Together they form $\beta^{(0)} = (\alpha^{(0)}, \gamma^{(0)})$. Note, however, that 'in principle' they don't need to share their values of the parameter. But, as alluded below, this may cause some computational problems. Therefore, in order to facilitate these, one might consider also the case where the parties do share their values. Using the two-party secure summation protocol, they jointly obtain $\pi^{(0)}$ by (2) or (6). (Strictly speaking, secure summation is not possible for two parties, but this is not an issue in the general case.) Plugging this into expressions (4), (5) and (7), the parties can utilize a secure matrix multiplication (e.g., as in

Sanil et al. (2007)) to have also the gradient $\nabla_l^{(0)}$, and the Hessian $H_l^{(0)}$. To see this, assume that the party holding data U holds in addition (and without loss of generality) the response variable y. This party can clearly compute $l_\alpha(\beta)$ locally. The other party needs either to obtain the response variable (assuming the first party is willing to share), or to apply a secure matrix product to have its part $l_\gamma(\beta)$. Off-diagonal sub-block matrices of the Hessian may be computed by applying a secure matrix product. The inverse may be evaluated by the following general formula,

$$A^{-1} = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{21} & \mathcal{A}_{22} \end{bmatrix},$$

where $\mathcal{A}_{11} = (A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$, $\mathcal{A}_{12} = -A_{11}^{-1}A_{12}(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}$, $\mathcal{A}_{21} = -A_{22}^{-1}A_{21}(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1}$, and $\mathcal{A}_{22} = (A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}$. Here, A is an $N \times N$ matrix partitioned as:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

The product $H_l^{-1}\nabla_l$ is conducted according to the sub-block matrices of H_l^{-1} and with the aid of a secure matrix product. After completion, each party may update its own share of the parameter β, and thus obtain the next point of the Newton-Raphson sequence $\beta^{(1)}$. Because of numerical complexities, however, there are subtleties (Karr et al., 2007), e.g., the parties must agree on the value of the Newton-Raphson iteration. Also the parties must be willing to share their estimated values of their components of β. This is a non-trivial assumption for pure vertically partitioned data and may reveal some confidential information.

There are possible leakage issues that one needs to address. Risk of disclosure may result from the process used to obtain π_i. Although we apply a secure summation protocol, party holding data V knows the parameter α corresponding to party holding data U (by assumption). Therefore, party holding V may gain valuable information in the course of the evaluation, especially for sparse rows u_i^t of U. In fact, the algorithm we present in this section and in the sequel are as much secure as the secure multi-party protocols are. In that respect, the secure matrix product used to evaluate the gradient may suffer from loss of privacy since one of the matrices has dimension one, e.g., V^ty. See Sanil et al. (2007) for a proposed secure matrix product protocol that achieves the goal of equating the loss of privacy incurred by both parties.

The generalization to multi-party problems ($K > 2$) is quite straightforward. One only has to use an appropriate multi-party secure sum protocol, and to apply the matrix multiplication protocol to every pair of parties.

3. Logistic Regression in the General Case

We now consider the case described in §1: attributes are partitioned among parties, but not every data record is common to all parties. It is natural to put this case into a missing data framework.

As before, let $X = [X_1, \ldots, X_K]$ for parties $A_1, \ldots, A_K$, but each sub-block matrix X_k may now contain missing values (for those records that party A_k does not have in its possession). Therefore, we expand the covariates $x_i \in \mathbb{R}^{p+1}$ as $x_i = (x_i^{o_i}, x_i^{m_i})$, where o_i, m_i stand for the observed and missing attributes, respectively, and they are both further expanded as $x_i^{o_i} = (x_i^{o_i}(A_1), \ldots, x_i^{o_i}(A_K))$ and $x_i^{m_i} = (x_i^{m_i}(A_1), \ldots, x_i^{m_i}(A_K))$.

We begin with a simplified version of the general case. We assume that there is no overlapping of variables between different parties, and that an exact match can be made using unique identifiers that the parties share. We have a missing data element and thus choose to view the problem in a missing data framework, where we can make distributional assumptions regarding the variables and then use the EM algorithm to impute missing values. There are two cleanly identifiable cases; one involves only continuous covariates and the other only categorical ones. We commence with the continuous case.

3.1 Continuous Predictors

Following Williams et al. (2005) we presume that the covariates x_i follow a Gaussian mixture model (GMM) with two components to the incompleteness, associated with the mixture parameter and with the Gaussian distribution parameters which correspond to genuinely missing data elements, $x_i^{m_i}$. We apply a version of the EM algorithm simultaneously to both components.

Assume, for a moment, that only one party is involved (or that the parties *are* willing to share their data). The formulation of the GMM and the EM algorithm is as follows. Let $\phi(\cdot\,; \mu, \Sigma)$ denote the multivariate normal density with mean vector μ, and variance matrix Σ. Let the distribution of $x_i = (x_i^{o_i}, x_i^{m_i})$ be given by:

$$f(x_i) = \sum_{j=1}^{J} \pi_j \phi(x_i\,;\, \mu_j, \Sigma_j), \tag{9}$$

where $\mu_j = (\mu_j^{o_i}, \mu_j^{m_i})$, and Σ_j is a partitioned matrix with sub-matrices $\Sigma_j^{o_io_i}, \Sigma_j^{o_im_i}, \Sigma_j^{m_io_i}$, and $\Sigma_j^{m_im_i}$. Denote by Z the missing component, where $Z_{ij} = 1$ if observation x_i originated from the j'th mixture component. For $\theta = \{(\pi_j, \mu_j, \Sigma_j)\,,\, j = 1, \ldots, J\}$, the complete log-likelihood is then

$$l_c(\theta|X, Z) = \sum_{i=1}^{n}\sum_{j=1}^{J} Z_{ij}\left[\log\phi(x_i\,;\, \mu_j, \Sigma_j) + \log\pi_j\right]. \tag{10}$$

The E-step evaluates the conditional expectation of $l_c(\theta|X, Z)$ given the observed data $X^o = \{x_i^{o_i}\,,\, i = 1, \ldots, n\}$. Here we simply need the standard calculation regarding the expectation of normal random variables. Maximization with respect to the mixture parameter is easy, and yields

$$\pi_j = \frac{1}{n}\sum_{i=1}^{n}\hat{\alpha}_{ij}, \text{ where } \hat{\alpha}_{ij} = \frac{\phi(x_i^{o_i}\,;\,\hat{\mu}_j^{o_i}, \hat{\Sigma}_j^{o_i o_i})\hat{\pi}_j}{\sum_{k=1}^{J}\phi(x_i^{o_i}\,;\,\hat{\mu}_k^{o_i}, \hat{\Sigma}_k^{o_i o_i})\hat{\pi}_k}. \tag{11}$$

The hats in (11) emphasize the fact that computations use values of θ from a preceding iteration. The updating equations for the normal parameters using an EM algorithm given by Williams et al. (2005) are:

$$\mu_j = \frac{1}{\sum_{i=1}^{n}\hat{\alpha}_{ij}}\sum_{i=1}^{n}\hat{\alpha}_{ij}\begin{bmatrix} x_i^{o_i} \\ \hat{E}_{ij}\end{bmatrix}, \tag{12}$$

$$\Sigma_j = \frac{1}{\sum_{i=1}^{n}\hat{\alpha}_{ij}}\sum_{i=1}^{n}\hat{\alpha}_{ij}\Omega_{ij}\ , \tag{13}$$

where

$$\Omega_{ij} = \left\{\left(\begin{bmatrix} x_i^{o_i} \\ \hat{E}_{ij}\end{bmatrix} - \mu_j\right)\left(\begin{bmatrix} x_i^{o_i} \\ \hat{E}_{ij}\end{bmatrix} - \mu_j\right)^t + \begin{bmatrix} 0 & 0 \\ 0 & \hat{V}_{ij}\end{bmatrix}\right\},$$

and $\hat{E}_{ij}$, and $\hat{V}_{ij}$ denote, respectively, conditional expectation (the imputed values) and conditional variance of $x_i^{m_i}$ given $x_i^{o_i}$, with respect to the parameters from the previous iteration (and normal component j).

To incorporate the fact that the parties are unwilling to (or cannot) share their values we perform a secure logistic regression in two steps. The first step involves the generalization of the updating equations to the case under which $K > 1$ parties hold private data. In the case of only two parties A and B, the parameters take the form:

$$\mu_j = (\mu_j^{o_i}(A), \mu_j^{o_i}(B), \mu_j^{m_i}(A), \mu_j^{m_i}(B))\ , \tag{14}$$

and Σ_j is, again, a partitioned matrix with sub-matrices in an appropriate form, e.g.,

$$\begin{gathered}\Sigma_j^{o_i o_i}(A,A), \Sigma_j^{o_i o_i}(A,B), \Sigma_j^{o_i m_i}(A,A), \ldots \\ \ldots, \Sigma_j^{m_i m_i}(A,B), \Sigma_j^{m_i m_i}(B,B)\ .\end{gathered} \tag{15}$$

We must then show how to use secure protocols to evaluate these equations. The second step may use the MLE's of $\theta = \{(\pi_j, \mu_j, \Sigma_j)\ ,\ j = 1, \ldots, J\}$, or the imputed values together with a standard Newton-Raphson algorithm to estimate the logistic parameters, through the approach presented for vertically partitioned data.

Now suppose that K parties, $A_1, A_2, \ldots, A_K$, are involved. By repeating the arguments from the E-step and M-step, we end up with similar expressions for the parameters. Expression (11) for π_j remains identical. We only need to take care when analyzing it since $x_i^{o_i}$ is decomposed now into the different parties' values. The expressions for μ_j, and Σ_j have the same structure, as one expects. The term $\hat{E}_{ij}$ may be written as: $\hat{E}_{ij} = (\hat{E}_{ij}(A_1), \ldots, \hat{E}_{ij}(A_K))^t$, with each component given by:

$$\begin{aligned}\hat{E}_{ij}(A_k) &= \mathbb{E}\left(x_i^{m_i}(A_k)|x_i^{o_i}, Z_{ij} = 1, \hat{\theta}\right) \\ &= \hat{\mu}_j^{m_i}(A_k) + \hat{\Sigma}_j^{m_i,o_i}(A_k,\cdot)\hat{\Sigma}_j^{-1,o_i o_i}(\cdot,\cdot)(x_i^{o_i} - \hat{\mu}_j^{o_i})\ ,\end{aligned} \tag{16}$$

for $k = 1, \ldots, K$. The notation $\hat{\Sigma}_j^{m_i o_i}(A_k,\cdot)$ is used to denote the covariance (under mixture component j) between $x_i^{m_i}(A_k)$ and $x_i^{o_i} = (x_i^{o_i}(A_1), \ldots, x_i^{o_i}(A_K))$. Other uses of this notation are to be understood similarly.

The term $\hat{V}_{ij}$ is the (conditional) variance-covariance matrix of $x_i^{m_i}$ given $(x_i^{o_i}, Z_{ij} = 1, \hat{\theta})$. This matrix is a partitioned matrix whose blocks may be written as

$$\begin{aligned}&\hat{V}_{ij}(A_k, A_l) = \\ &\quad\hat{\Sigma}_j^{m_i m_i}(A_k, A_l) - \hat{\Sigma}_j^{m_i o_i}(A_k,\cdot)\hat{\Sigma}_j^{-1,o_i o_i}(\cdot,\cdot)\hat{\Sigma}_j^{o_i m_i}(\cdot, A_l)\ ,\end{aligned} \tag{17}$$

where $k, l = 1, \ldots, K$.

We now show how to iterate in a secure fashion using the updating equations. For π_j, note that the numerator involves essentially the sum:

$$\sum_{k,l=1}^{K} x_i^{o_i}(A_k)^t\hat{\Sigma}^{-1,o_i o_i}(A_k, A_l)x_i^{o_i}(A_l)\ . \tag{18}$$

The (sub-)sum over all $k = l$, whose addends are local to each party, may be computed by a secure sum protocol. The sum over $k \neq l$ involves different parties and may be computed by a secure dot product protocol (Sanil et al., 2007). Together, we securely find π_j for every j.

Consider next the terms $\hat{E}_{ij}(A_k)$, and $\hat{V}_{ij}(A_k, A_l)$. We may proceed as suggested in Reiter et al. (2004). We group together records in the database according to missing data patterns. The parties will only have to share summary statistics (see (21) and (23) below). Let us assume that the parties are willing to share the values of μ_j, and Σ_j. Assume further that there are no missing values in the private raw data held separately by the various parties, that is, $x_i(A_k)$ is either $x_i^{o_i}(A_k)$ or $x_i^{m_i}(A_k)$, and also $\hat{\mu}_j^{m_i}(A_k) = \hat{\mu}_j(A_k)$.

Under the previous assumptions, the $\hat{V}_{ij}(A_k, A_l)$, given by (17), can be computed by each of the participating parties. The only troublesome term is the second term on the second line of (16), namely,

$$\hat{\Sigma}_j^{m_i,o_i}(A_k,\cdot)\hat{\Sigma}_j^{-1,o_i o_i}(\cdot,\cdot)(x_i^{o_i} - \hat{\mu}_j^{o_i}). \tag{19}$$

While the matrix $\hat{\Sigma}_j^{m_i,o_i}(A_k,\cdot)\hat{\Sigma}_j^{-1,o_i o_i}(\cdot,\cdot)$ is shared by party A_k and the rest, the vector of observed $x_i^{o_i}$ is composed of private block vectors and is not available to party A_k.

Let M be the number of missing data patterns. Let $\{I_1, \ldots, I_M\}$ be a partition of $\{1, \ldots, n\}$, i.e., $I_1, \ldots, I_M$ are mutually disjoint and $\cup_{m=1}^{M} I_m = \{1, \ldots, n\}$. Referring to (12), we only need to care about the missing components $\sum_{i=1}^{n}\hat{\alpha}_{ij}\hat{E}_{ij}$. Decomposing the sum over all observations according to the missing data patterns, for party k we write

$$\begin{aligned}&\sum_{i=1}^{n}\hat{\alpha}_{ij}\hat{E}_{ij}(A_k) = \\ &\quad\sum_{m=1}^{M} a_{j,k,m}\sum_{i\in I_m}\hat{\alpha}_{ij} + \sum_{m=1}^{M} b_{j,k,m}\left[\sum_{i\in I_m}\hat{\alpha}_{ij}x_i^{o_i}\right]\ ,\end{aligned} \tag{20}$$

where $a_{j,k,m}$, and $b_{j,k,m}$ are functions of $\hat{\mu}_j, \hat{\Sigma}_j$, and are fixed within any missing pattern I_m:

$$\begin{aligned} a_{j,k,m} &= \hat{\mu}_j^{m_i}(A_k) - \hat{\Sigma}_j^{m_i,o_i}(A_k,\cdot)\hat{\Sigma}_j^{o_i,o_i}(\cdot,\cdot)^{-1}\hat{\mu}_j^{o_i} \ , \\ b_{j,k,m} &= \hat{\Sigma}_j^{m_i,o_i}(A_k,\cdot)\hat{\Sigma}_j^{o_i,o_i}(\cdot,\cdot)^{-1} \ . \end{aligned}$$

All parties know $\hat{\alpha}_{ij}$, and can calculate it securely. Therefore, the first addend in (20) poses no disclosure limitation problem. For the second addend, we assume that the parties are willing to share a linear combination of their values:

$$\sum_{i \in I_m} \hat{\alpha}_{ij} x_i^{o_i}, \quad \text{for every } , m = 1, \ldots, M. \tag{21}$$

This is similar to the table of summary statistics that Reiter et al. (2004) consider, where the sum of the observed is shared. Every party A_l computes $\sum_{i \in I_m} \hat{\alpha}_{ij} x_i^{o_i}(A_l)$, for every j, m and shares it among the other parties. This reveals nothing to party A_k, with $k \neq l$, who only gets to learn a linear combination, say b, of the observed variables: $b = \sum_{i \in I_m} \hat{\alpha}_{ij} x_i^{o_i}(A_l)$. Note that party A_k knows $\hat{\alpha}_{ij}$, but still cannot have $x_i^{o_i}(A_l)$. However, there may be concerns for privacy, e.g., when I_m is very small for some $m = 1, \ldots, M$, and $x_i^{o_i}(A_l)$ is of a particular sparse form. One other drawback is that the parties need to share the linear combination every iteration, since $\hat{\alpha}_{ij}$ changes in every iteration. This may cause serious computational issues as well as possible risk for privacy; more iterations leads to an increase in the number of sharing, which, undoubtedly, means more information that might be leaked out.

Consider now the updating of $\hat{\Sigma}_j$ through (13). The rightmost term that involves $\hat{V}_{ij}$ causes no problem, and is assumed to be known for every party. The other term in the curely brackets of (13) may be written as:

$$\begin{pmatrix} x_i^{o_i}(x_i^{o_i})^t & x_i^{o_i}\hat{E}_{ij}^t \\ \hat{E}_{ij}(x_i^{o_i})^t & \hat{E}_{ij}\hat{E}_{ij}^t \end{pmatrix} - \begin{pmatrix} x_i^{o_i} \\ \hat{E}_{ij} \end{pmatrix} \mu_j^t - \mu_j((x_i^{o_i})^t, \hat{E}_{ij}^t) + \mu_j\mu_j^t \ .$$

Summing up over all $i = 1, \ldots, n$ leads to:

$$\frac{1}{\sum_{l=1}^n \hat{\alpha}_{lj}} \sum_{i=1}^n \hat{\alpha}_{ij} \begin{pmatrix} x_i^{o_i}(x_i^{o_i})^t & x_i^{o_i}\hat{E}_{ij}^t \\ \hat{E}_{ij}(x_i^{o_i})^t & \hat{E}_{ij}\hat{E}_{ij}^t \end{pmatrix} - \mu_j\mu_j^t \ , \tag{22}$$

the last term, $\mu_j\mu_j^t$, being readily calculated.

The first term in (22) consists essentially of three sums of matrices, $\sum_{i=1}^n \hat{\alpha}_{ij} x_i^{o_i}(x_i^{o_i})^t$, $\sum_{i=1}^n \hat{\alpha}_{ij} x_i^{o_i} \hat{E}_{ij}^t$, and $\sum_{i=1}^n \hat{\alpha}_{ij} \hat{E}_{ij} \hat{E}_{ij}^t$. The first sum decomposes into sums of the form $\sum_{i \in I_m} \hat{\alpha}_{ij} x_i^{o_i}(A_k) x_i^{o_i}(A_l)^t$. This is given by dot products of the covariates of parties A_k and A_l, and may be securely calculated using the secure dot product protocol, when $k \neq l$. When $k = l$, this quantity is known only to party A_k. In parallel to the sharing of the linear combination (21), we assume that the parties are willing also to share

$$\sum_{i \in I_m} \hat{\alpha}_{ij} x_i^{o_i}(x_i^{o_i})^t \quad , \quad \text{for every } m = 1, \ldots, M \ , \tag{23}$$

The on-diagonal blocks of this matrix are the matrices that the parties share. The off-diagonal blocks are securely computed as explained above. Consider now the other two sums mentioned earlier. Note that $\hat{E}_{ij}(A_k) = a_{j,k,m} + b_{j,k,m} x_i^{o_i}$. Therefore, the sharing of (21), and (23) (and the use of a secure dot product) is sufficient in order to compute $\sum_{i=1}^n \hat{\alpha}_{ij} x_i^{o_i} \hat{E}_{ij}^t$, and $\sum_{i=1}^n \hat{\alpha}_{ij} \hat{E}_{ij} \hat{E}_{ij}^t$ in a secure fashion. Note, however, that this repeated sharing may lead to threats for privacy, as already pointed out. Party A_k may try to use repeated values of $\hat{\alpha}_{ij}$ in order to guess $x_i^{o_i}(A_l)$.

3.2 Categorical Predictors

Suppose now that the covariates $x_i, i = 1, \ldots, p$ are categorical. We pursue the analysis on the logistical scale and defer a discussion of the related log-linear model formulation to the full paper. The $n \times p$ design matrix (neglecting the first column of ones) can be regarded as a p-dimensional contingency table. Let C be the total number of cells. We index the cells using a single symbol, $c = 1, \ldots, C$. Corresponding to each x_i, we may associate a vector w_i of size $1 \times C$, which maps subject i into cell c. The elements of w_i are all zeros except for the c'th element which is equal to one.

When missing values appear in the raw data, we no longer can associate x_i with a *unique* vector w_i, and are only able to form a subset S_i of cells where subject i could lie. This leads to sub-tables of lower-dimension corresponding to the observed variables $x_i^{o_i}$ (See Little and Rubin (2002, Ch. 14)).

A natural modification of the Gaussian mixture model in the continuous case is the following. We presume that the variables w_i follow a Multinomial mixture model (MMM). The distribution of w_i is given by:

$$f(w_i) = \sum_{j=1}^J \pi_j \text{Mult}(w_i\,; (1, C, p_j)) \ , \tag{24}$$

where $\text{Mult}(w_i\,; (1, C, p_j))$ is the multinomial probability function corresponding to a single trial with C possible outcomes of probabilities $p_j = (p_{j1}, \ldots, p_{jC})$. Let $\theta = \{(\pi_j, p_j)\,; j = 1, \ldots, J\}$. The complete log-likelihood (see (10)) is:

$$\sum_{i=1}^n \sum_{j=1}^J \sum_{c=1}^C Z_{ij} w_{ic} \log\{p_{jc}\} + \sum_{i=1}^n \sum_{j=1}^J Z_{ij} \log\{\pi_j\} \ , \tag{25}$$

where we used the fact that $w_{ic}! = 1$ for each $i = 1, \ldots, n$, $c = 1, \ldots, C$. Details of the E-step and the M-step are given below.

3.2.1 E-step

The E-step evaluates the conditional expectation (under previous values of the parameter θ) of the complete log-likelihood given $\{S_i\,; i = 1, \ldots n\}$. Let $\hat{\alpha}_{ij} = \mathbb{E}_{\hat{\theta}}[Z_{ij} \mid S_i]$.

Then,

$$\hat{\alpha}_{ij} = \frac{\hat{\pi}_j \sum_{c \in S_i} \hat{p}_{jc}}{\sum_{k=1}^{J} \hat{\pi}_k \sum_{c \in S_i} \hat{p}_{kc}} \ , \tag{26}$$

and

$$\mathbb{E}_{\hat{\theta}}\big[Z_{ij} w_{ic} \,|\, S_i\big] = \left\{ \begin{array}{cc} \frac{\hat{p}_{jc}}{\sum_{\tilde{c} \in S_i} \hat{p}_{j\tilde{c}}} \times \hat{\alpha}_{ij} & c \in S_i \\ 0 & c \notin S_i \end{array} \right. .$$

3.2.2 M-step

Maximization with respect to π_j, subject to $\sum_{j=1}^{J} \pi_j = 1$, yields:

$$\pi_j = \frac{1}{n} \sum_{i=1}^{n} \hat{\alpha}_{ij} \ . \tag{27}$$

Let $\hat{\beta}_{ij} = \hat{\alpha}_{ij} / \sum_{c \in S_i} \hat{p}_{jc}$. Then,

$$\frac{\partial}{\partial p_{i\tilde{c}}} \Big(\sum_{i=1}^{n} \hat{\beta}_{ij} \sum_{c \in S_i} \log\{p_{jc}\} \, \hat{p}_{jc} \Big) = \sum_{i=1}^{n} \hat{\beta}_{ij} \frac{\hat{p}_{j\tilde{c}}}{p_{j\tilde{c}}} \mathbb{I}_{\{\tilde{c} \in S_i\}} \ ,$$

which leads to

$$p_{jc} = \frac{\hat{p}_{jc} \sum_{i=1}^{n} \hat{\beta}_{ij} \mathbb{I}_{\{c \in S_i\}}}{\sum_{\tilde{c}=1}^{C} \hat{p}_{j\tilde{c}} \sum_{i=1}^{n} \hat{\beta}_{ij} \mathbb{I}_{\{\tilde{c} \in S_i\}}} \ . \tag{28}$$

Note that the updating equations, (27) and (28), depend on the data only through $S_i, i = 1, \ldots, n$. We assume that every party knows the missing data pattern of every observation, i.e., the S_i's are known to every party. We also assume that each party knows the set of categories for every variable. Therefore, the analysis can be executed separately by each of the participating parties. Parties don't have to share summary statistics, only what is needed for every party to learn S_i.

Upon convergence, each party fill-in their missing values by drawing observations from the marginal multinomial distribution whose parameters are the $\hat{\pi}_j$, and a suitable subset of the $\hat{p}_j$.

4. An example

We illustrate the approach described above and our secure logistic regression protocol using a restructured version of data from the 1993 US Current Population Survey. Versions of these data have been used previously to illustrate several other approaches to confidentiality protection. There are 48,842 cases with 8 categorical variables, listed in the Table 1, which, for purely illustrative purposes, we partition in Table 2 among $K = 3$ agencies, designated as A, B, and C, whom we presume wish to jointly, but securely, analyze the data.

Consider $m = 3$, the third missing data pattern illustrated in Table 2. The entries of the table are aligned by missing data patterns. Party A wants to impute its missing values for variables x_1, x_2 and records $\{10201, ..., 22481\} = I_3$. Note that $\hat{\alpha}_{5j}$ and $\hat{\alpha}_{6j}$ are known to both party A and parties (B, C). Party B computes $\sum_{i \in I_3} \hat{\alpha}_{ij} (x_{i3}, x_{i4}, x_{i5})^t$ and sends it to party A. Likewise, party C computes $\sum_{i \in I_3} \hat{\alpha}_{ij} (x_{i6}, x_{i7})^t$ and sends it to party C. Party A then computes the elements of $\hat{\mu}_j$ corresponding to its variables, and it does not get to see the observed variables $(x_{i3}, x_{i4}, x_{i5}, x_{i6}, x_{i7})$ themselves.

5. Simulation

We have simulated a vertically partitioned, partially overlapping data of size $n = 100$ from the Gaussian mixture model:

$$f(x_i) = \pi_1 \, \phi(x_i; \mu_1, I_6) + \pi_2 \, \phi(x_i; \mu_2, I_6) \ ,$$

where $\pi = (\pi_1, \pi_2)^t = (0.25, 0.75)^t$, $\mu_2 = -\mu_1 = (2, 2, 2, 2, 2, 2)^t$, and I_6 is the identity matrix of dimensions 6×6. The six attributes were distributed among $K = 3$ parties (2 attributes each) according to the following missing pattern: Party A did not record values for observations $(41 - 60)$, Party B did not record values for observations $(21 - 40), (61 - 80)$, and Party C did not record values for observations $(1 - 20), (61 - 80)$. This database is similar to Table 2. Initializing the EM algorithm with parameters $\pi = (0.5, 0.5)^t$, and $\mu_2 = -\mu_1 = (2, 2, 2, 2, 2, 2)$, we obtained the following maximum likelihood estimates $\hat{\pi}_1 = (0.27, 0.73)$, $\hat{\mu}_1 = -(1.9208, 1.7280, 1.4835, 2.0116, 1.9428, 1.7600)^t$, and $\hat{\mu}_2 = (2.1434, 1.9317, 1.9981, 1.9186, 2.1776, 1.9769)^t$. These estimates show good agreement with the true values. We also initialized the EM algorithm with $\pi = (0.1, 0.9)^t$, and $\mu_2 = -\mu_1 = (1, 1, 1, 1, 1, 1)^t$, and obtained estimates $\hat{\pi}_1 = (0.24, 0.76)$, $\hat{\mu}_1 = -(1.9260, 1.9113, 1.8845, 2.0219, 1.8883, 1.8353)^t$, and $\hat{\mu}_2 = (1.8479, 1.7490, 1.8608, 2.0578, 2.1636, 1.8924)^t$, which are again in good agreement with the true values.

We conducted another simulation study to examine how intensive is the algorithm in terms of CPU time, and with increasing number of parties. We simulated a Gaussian mixture model of size $n = 1,000$, and distributed the database among $p = 45$ parties. Figure 1 shows the time in seconds it took the algorithm to converge as a function of the number of parties. The algorithm's time-to-converge appears to be roughly linear in the number of parties.

6. Discussion

We presented algorithms which can be used to perform secure logistic regression when the data base is distributed among K parties. We considered three types of data partitioning: horizontally partitioned data, vertically partitioned data and a general case which involves vertically partitioned, partially overlapping data. We also considered (separately) continuous predictors and categorical predictors.

The secure logistic regression protocol is computationally intensive since the secure matrix operations need to performed at each iteration of the Newton-Raphson algorithm. For example, a linear combination of weighted

Variable	Label	Categories
Age (in years)	X_1	$< 25, 25 - 55, > 55$
Employer Type (*Empolyment*)	X_2	Gov, Pvt, SE, Other
Education	X_3	<HS, HS, Bach, Bach+, Coll
Marital status (*Marital*)	X_4	Married, Other
Race	X_5	White, Non-White
Sex	X_6	Male, Female
Hours Worked (*HrsWorked*)	X_7	$< 40, 40, > 40$
Annual Salary (*Salary*)	Y	$< \$50K, \$50K+$

Table 1: Description of the response variable (Y) and explanatory variables ($X_1, ..., X_7$) used for illustration of the secure logistic regression protocol.

	Agency A		Agency B			Agency C	
n	X_1	X_2	X_3	X_4	X_5	X_6	X_7
1	√	√	√	√	√	●	●
...	√	√	√	√	√	●	●
9000	√	√	√	√	√	●	●
9001...10200	√	√	●	●	●	√	√
10201...22481	●	●	√	√	√	√	√
22482...48842	√	√	●	●	●	●	●

Table 2: An example of an (aligned) concatenated database. Party A records values of x_1, x_2 and observations $1 - 10200, 22482 - 48842$, Party B records values of x_3, x_4, x_5 of observations $1 - 9000, 10201 - 22481$, and Party C records values of x_6, x_7 of observations $9001 - 22481$.

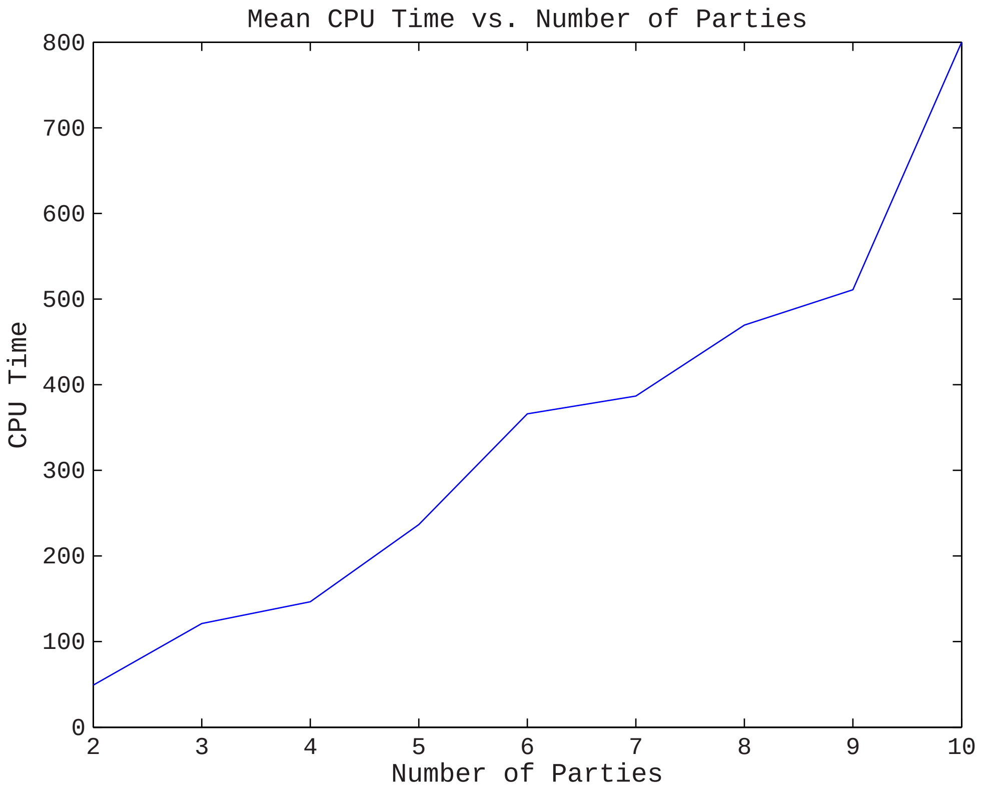

Figure 1: Mean CPU time vs. Number of Parties

sums must be shared at each iteration. Fienberg et al. (2006) point out that for horizontal case log-linear "secure" computation is more efficient. In the general case, an EM algorithm precedes the logistic parameter estimation through the Newton-Raphson algorithm. This may pose further computational issues.

The secure logistic regression protocol is not free from leakages. Generally speaking, it is as much secure as the secure multi-party computations are. But these computations may disclose information. Two parties participating in such a computation each relinquish information to the other, in the form, e.g., of vectors orthogonal to their respective databases (Sanil et al., 2007). Furthermore, risk for privacy may come from other reasons as well. For example, the assumption that parties are willing to share their parts of the estimated parameters may reveal information. There are other disclosure risks: if the analysis reveals that attributes held by agency A predict those held by agency 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 Sanil et al. (2004). The secure EM protocol is preformed with in each data missingness pattern identified in the "global" aligned database. Risks associated with this protocol, as discussed in Reiter et al. (2004), are applicable in our setting and need further careful consideration.

Acknowledgments

We owe special thanks to Matthew Tibbits for his programming efforts. This research was supported in part by NSF grant EIA-01-31884 to the National Institute of Statistical Sciences, by NSF grant DMS-0631589 and Army contract DAAD19-02-1-3-0389 to Carnegie Mellon University, and by NSF grant SES-0532407 to Pennsylvania State University.

References

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

Y.M.M. Bishop, S.E. Fienberg, and P.W. Holland. *Discrete Multivariate Analysis: Theory and Practice.* MIT Press, Cambridge MA, 1975. Reprinted, Springer-Verlag, New York, 2007.

C. Clifton, J. Vaidya, and M. Zhu. *Privacy Preserving Data Mining.* Springer-Verlag, New York, 2006.

W. Du, Y. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 4th SIAM International Conference on Data Mining*, pages 222–233, April 2004.

W. Du and Z. Zhan. A practical approach to solve secure multiparty computation problems. In *New Security Paradigms Workshop*, pages 127–135, New York, September 2002. ACM Press.

S.E. Fienberg. *Analysis of Cross-Classified Categorical Data.* MIT Press, Cambridge MA, 2nd edition, 1980. Reprinted, Springer-Verlag, New York, 2007.

S.E. Fienberg, W.J. Fulp, A.B. Slavkovic, and T.A. Wrobel. "Secure" log-linear and logistic regression analysis of distributed databases. In J. Domingo-Ferrer and L. Franconi, editors, *Privacy in Statistical Databases: CENEX-SDC Project International Conference, PSD 2006*, pages 277–290, Berlin, 2006. Springer-Verlag.

J. Ghosh, J.P. Reiter, and A.F. Karr. Secure computation with horizontally partitioned data using adaptive regression splines. *Computational Statistics and Data Analysis*, 2006. To appear.

S. Goldwasser. Multi-party computations: Past and present. In *Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing*, pages 1–6, New York, August 1997. ACM Press.

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

A.F. Karr, X. Lin, J.P. Reiter, and A.P. Sanil. Secure regression on distributed databases. *Journal of Computational and Graphical Statistics*, 14(2):263–279, 2005.

Roderick J. A. Little and Donald B. Rubin. *Statistical analysis with missing data.* Wiley Series in Probability and Statistics. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, second edition, 2002. ISBN 0-471-18386-5.

M.J.D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7:152–162, 1964.

J.P. Reiter, A.F. Karr, C.N. Kohnen, X. Lin, and A.P. Sanil. Secure regression for vertically partitioned, partially overlapping data. *Proceedings of the American Statistical Association*, 2004.

A.P. Sanil, A.F. Karr, X. Lin, and J.P. Reiter. Privacy preserving regression modelling via distributed computation. In *Proc. Tenth ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining*, pages 677–682, 2004.

A.P. Sanil, A.F. Karr, X. Lin, and J.P. Reiter. Privacy preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 2007. Revised manuscript under review.

D. Williams, X. Liao, Y. Xue, and L. Carin. Incomplete-data classification using logistic regression. In *ICML '05: Proceedings of the 22nd International Conference on Machine Learning*, pages 972–979, New York, 2005. ACM Press.

A.C. Yao. Protocols for secure computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 160–164, New York, 1982. ACM Press.