Journal of Official Statistics, Vol. 25, No. 1, 2009, pp. 125–138

Privacy-Preserving Analysis of Vertically Partitioned Data Using Secure Matrix Products

Alan F. Karr[1], *Xiaodong Lin*[1*], *Ashish P. Sanil*[1†] *and Jerome P. Reiter*[2]

Reluctance of statistical agencies and other data owners to share possibly confidential or proprietary data with others who own related databases is a serious impediment to conducting mutually beneficial analyses. In this article, we propose a protocol for conducting secure regressions and similar analyses on vertically partitioned data – databases with identical records but disjoint sets of attributes. This protocol allows data owners to estimate coefficients and standard errors of linear regressions, and to examine regression model diagnostics, without disclosing the values of their attributes to each other. No third parties are involved. The protocol can be used to perform other procedures for which sample means and covariances are sufficient statistics. The basis is an algorithm for secure matrix multiplication, which is used by pairs of owners to compute off-diagonal blocks of the full data covariance matrix.

Key words: Distributed databases; secure matrix product; vertically partitioned data; regression; data confidentiality.

1. Introduction

In numerous contexts, immense utility can arise from statistical analyses that integrate multiple, distributed databases. For example, statistical models can be fit using more records or more attributes when databases are integrated than when databases are analyzed separately. Data integration is complicated by concerns about data confidentiality, including legal, regulatory and even physical (scale of data) barriers. These concerns can be present even when the database owners cooperate to perform integrated analyses, and none seeks to break the confidentiality of others' data.

Within the statistics literature, most attention has been directed to the case of *horizontally partitioned* databases comprising the same numerical attributes for disjoint sets of data subjects. For example, several state or local educational agencies might want to combine their students' data to improve the precision of analyses of the general student

[1] National Institute of Statistical Sciences, Research Triangle Park, NC, U.S.A. Email: karr@niss.org, linxd@math.uc.edu and ashish@niss.org
[2] Duke University, Durham, NC, U.S.A. Email: jerry@stat.duke.edu
[*]Currently at University of Cincinnati, Cincinnati, OH, U.S.A.
[†]Currently at Berry Consultants, Foster City, CA, U.S.A.

Acknowledgments: This research was supported by NSF grant EIA-0131884 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. The authors thank Lawrence Cox of the National Center for Health Statistics (NCHS) for insightful comments and suggestions.

population. Analyses based on sufficient statistics that are additive across the databases can be performed using secure summation (Benaloh 1987) to calculate these statistics. Examples include linear regression (Karr et al. 2004, 2005a, b), secure construction of contingency tables (Karr et al. 2007), data integration (Karr et al. 2007), maximum likelihood estimation for exponential families (Karr et al. 2007), secure EM algorithms (Reiter et al. 2004) and adaptive regression splines (Ghosh et al. 2007).

Less attention has been directed to *vertically partitioned* databases comprising the same data subjects but containing different sets of attributes. For example, one government agency might have employment information, another health data, and a third information about education, all for the same individuals or establishments. Linear regression for vertically partitioned data is treated in Sanil et al. (2004), but under the highly restrictive assumption that the response attribute is shared among all the owners.

In the computer science literature, privacy-preserving data mining (PPDM) has emerged as a promising approach in a variety of contexts (Agrawal and Srikant 2000; Clifton et al. 2003b; Lindell and Pinkas 2000; Vaidya and Clifton 2004). One root of PPDM is secure multi-party computation (SMPC) (Yao 1982), of which secure summation is a special case. Algorithms for horizontally partitioned data have been developed for data mining with association rules (Evfimievski et al. 2004; Kantarcioglu and Clifton 2002) and model-based clustering (Lin et al. 2004). For vertically partitioned data, secure analysis methods exist for association rule mining (Vaidya and Clifton 2002), *K*-means clustering (Vaidya and Clifton 2003), and linear discriminant analysis (Du et al. 2004). Some of these techniques are incomplete from a statistical perspective: for example, estimators are calculated but standard errors and other quantities that statisticians would regard as integral parts of analyses are not.

Other approaches, some of which have been studied by both statisticians and computer scientists, include data distortion and randomization. The field of statistical disclosure limitation is concerned with balancing protection of confidential data values with dissemination of useful information derived from the data. Examples include protecting categorical data underlying large contingency tables (Dobra et al. 2002, 2003), servers that disseminate the results of analyses rather than data (Gomatam et al. 2005a) and data swapping (Gomatam et al. 2005b). Underlying these methods are quantified measures of data utility and disclosure risk (for individual records) (Karr et al. 2006a).

The techniques presented in this article are designed to enable the database owners to perform analyses that none can perform individually because none has access to all the attributes. They protect the database owners from one another in the sense that only aggregated information is exchanged. Specifically, we show how to perform regression and related analyses on vertically partitioned data using an alternative approach to those of Du et al. (2004) and Sanil et al. (2004).

We assume the database owners will not share data values but are willing to share sample means and covariances of their individual databases. Our main focus is on computation of the full data covariance matrix (Section 2), whose computation requires that the owners surrender some dimensions of their data to each other.

It is important to note that the loss of protection discussed in this article applies only to the database owners vis-à-vis one another, and only in an aggregated sense of the span of their databases. Indeed, this is true in general for PPDM. The measure LP defined by (6) below does not address threats to the confidentiality of data records or the privacy of individual data subjects, the traditional focus of statistical disclosure limitation (Doyle et al. 2001; Willenborg and de Waal 2001), arising from either computation of the full data covariance matrix or use of it to perform analyses such as regressions. One exception to this, whereby an agency could learn exact data values held by another agency for one subject, is discussed in Section 2.4.

From shared means and the securely computed full data covariance matrix, the owners can perform richer sets of analyses than estimating regression coefficients. These analyses include inference for the coefficients, model diagnostics and model selection. We note that the approach of Du et al. (2004) can be modified to share sample covariance matrices, although the protocol presented in Section 2.1 holds advantages over it. The approach of Sanil et al. (2004) cannot be so modified.

The article is organized as follows. In Section 2, we provide a description of our method for computing the full data covariance matrix, under which lies a linear-algebra-based protocol for computing secure matrix products. Once this matrix has been calculated, it is possible, as shown in Section 3, to conduct secure linear regressions on arbitrary subsets of attributes, with proper attention to items such as model diagnostics. This section also describes extensions to other analyses. Conclusions appear in Section 4.

2. Computation of the Full Data Covariance Matrix

We label the data owners as Agency A, Agency B, . . . , Agency Ω, even though they might be private companies or other data holders. The "global" database **X**, illustrated for three agencies in Figure 1, is partitioned vertically among the agencies:

$$\mathbf{X} = [\mathbf{X}^A\mathbf{X}^B. . .\mathbf{X}^\Omega] \tag{1}$$

Let n be the number of records in the global database, suppose that agency α has p_α attributes, and let $p = p_\text{A} + \cdots + p_\Omega$.

A number of issues, some subtle and possibly difficult, underlie the process. First, the agencies must have a privacy-protecting method of determining which data subjects are common to all of their databases. The most straightforward way to do this is by means of a common primary key, such as social security numbers. In some instances, however, it might be necessary to use record linkage methods (Fellegi and Sunter 1969), or privacy-preserving versions (Clifton et al. 2003a; Schadow et al. 2002). Second, we assume that **X** comprises only complete records. (Methods for addressing systematically missing values are discussed in Karr et al. (2007) and Reiter et al. (2004).) Third, we assume that the agencies have aligned their common data subjects in the same order. Finally, we assume that the sets of attributes in the $\mathbf{X}^\alpha$ are disjoint; if not, the agencies coordinate so that each common attribute is included in only one agency's data.

Data quality problems (Karr et al. 2006b) may be present but are ignored. For instance, two agencies may have the same attribute but different values for it. This problem is detectable (for instance, using a secure dot product), but fixable only using external domain knowledge. Similarly, the possibility that the agencies' databases pertain to the same subjects but not at the same time (e.g., health data are from one year and economic data from another) can only be dealt with "off-line."

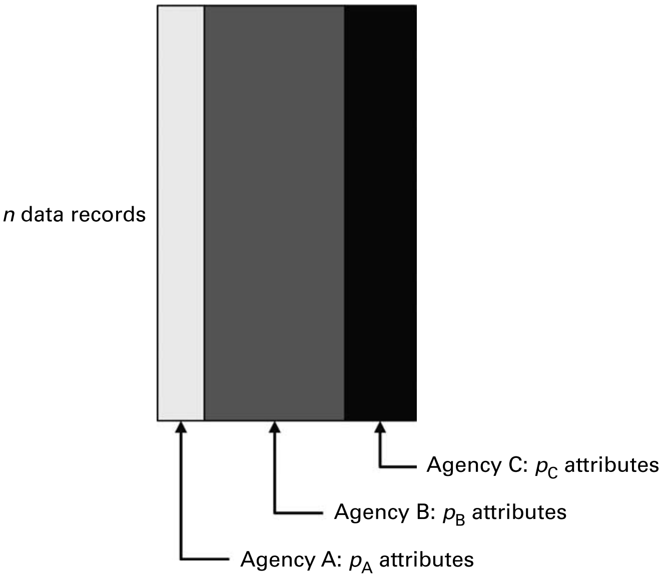
n data records
Agency C: p_C attributes
Agency B: p_B attributes
Agency A: p_A attributes

Fig. 1. Pictorial representation of vertically partitioned data. The blocks represent the data values held by three agencies

We further assume that the agencies are semi-honest: they adhere strictly to protocols designed to preserve privacy, and perform calculations using their real data. We believe the semi-honesty assumption is realistic for many data integration settings, including government agencies seeking to perform combined analyses using their data. Furthermore,

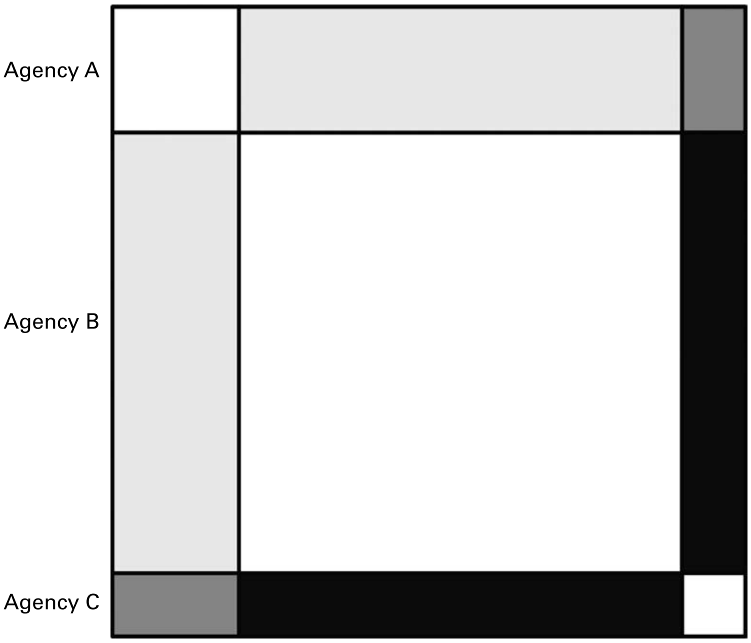

Fig. 2. Pictorial representation of the full data covariance matrix $\mathbf{X}^T\mathbf{X}$ *for three agencies*

we assume that the agencies do not collude with each other; some implications of this are discussed in Section 2.3.

The statistical analyses discussed in Section 3 are all based on the $(p \times p)$-dimensional full data covariance matrix $\mathbf{X}^T\mathbf{X}$, which is shown pictorially, also for three agencies, in Figure 2. The goal of the agencies is to compute and share $\mathbf{X}^T\mathbf{X}$ in a way that minimizes the information each reveals to the others about its data values. As that figure shows, $\mathbf{X}^T\mathbf{X}$ consists of:

On-diagonal blocks of the form $(\mathbf{X}^\alpha)^T\mathbf{X}^\alpha$. Each of these must be computed by one agency and shared with the others.

Off-diagonal blocks of the form $(\mathbf{X}^\alpha)^T\mathbf{X}^\beta$, where $\alpha \neq \beta$. Each of these must be computed by two agencies, and the result shared with the others.

Section 2.1 describes a protocol for computation of the $(\mathbf{X}^\alpha)^T\mathbf{X}^\beta$ using secure matrix multiplication.

2.1. A Secure Protocol for Computing Matrix Products

Here we use a form of secure matrix multiplication to compute off-diagonal blocks $(\mathbf{X}^\alpha)^T\mathbf{X}^\beta$ of the full data covariance matrix $\mathbf{X}^T\mathbf{X}$. For simplicity, consider Agencies A and B. We write the data of Agency A as

$$\mathbf{X}^{\mathrm{A}} = \left[X_1^{\mathrm{A}} X_2^{\mathrm{A}} \ldots X_{p_{\mathrm{A}}}^{\mathrm{A}}\right]$$

This is an unconventional representation in the sense that the X_i^{A} are *columns* in Agency A's data matrix – each belongs to $\mathbb{R}^n$. Similarly, we write Agency B's data as

$$X^{\mathrm{B}} = \left[X_1^{\mathrm{B}} X_2^{\mathrm{B}} \ldots X_{p_{\mathrm{B}}}^{\mathrm{B}}\right]$$

We assume that $\mathbf{X}^{\mathrm{A}}$ and $\mathbf{X}^{\mathrm{B}}$ are of full rank; if not, each agency removes any linearly dependent columns.

Agency A and Agency B wish to compute securely the $(p_{\mathrm{A}} \times p_{\mathrm{B}})$-dimensional matrix $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$, and share it with the other agencies. We first describe a generic protocol for computing $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$, and then show how it can be applied in such a way that the information exchanged between the two agencies is symmetric. Our protocol is as follows:

Step 1: Agency A generates a set of g n-dimensional vectors $\{Z_1 Z_2, \ldots, Z_g\}$ such that

$$Z_i^T X_j^{\mathrm{A}} = 0 \text{ for all } i \text{ and } j \tag{2}$$

and sends to Agency B the $(n \times g)$-dimensional matrix

$$\mathbf{Z} = [Z_1 Z_2 \ldots Z_g]$$

A method for generating $\mathbf{Z}$ is presented in the Appendix, which yields Z_i that are orthonormal. The choice of g is discussed in Section 2.2.

Step 2: Agency B computes

$$\mathbf{W} = (\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{X}^{\mathrm{B}} \tag{3}$$

where $\mathbf{I}$ is an $(n \times n)$-dimensional identity matrix, and sends $\mathbf{W}$ to Agency A.

Step 3: Agency A calculates

$$(\mathbf{X}^{\mathrm{A}})^T\mathbf{W} = (\mathbf{X}^{\mathrm{A}})^T(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)\mathbf{X}^{\mathrm{B}} = (\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$$

where the second equality holds since $(\mathbf{X}^{\mathrm{A}})^T\mathbf{Z} = 0$, and shares the value of $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$ with the other agencies.

In view of **Step 2**, Agency A could instead send the $n \times n$-dimensional matrix $\mathbf{Z}\mathbf{Z}^T$ to Agency B. On the face of it, this is safer, but in fact the loss of protection discussed in Section 2.2 does not change.

2.2. *Loss of Protection*

The absolute and relative protection that the protocol in Section 2.1 provides to Agencies A and B depends on the parameter g in **Step 1**. First, consider two extreme cases:

- $g = 0$: In this case, in (3), $\mathbf{W} = \mathbf{X}^{\mathrm{B}}$, so agency A has learned agency B's data exactly.
- $g = n - p$: In this case, Agency B knows exactly the orthogonal complement of $\mathbf{X}^{\mathrm{A}}$ in $\mathbb{R}^n$. While this does not specify $\mathbf{X}^{\mathrm{A}}$ exactly, Agency B does know the span of $\mathbf{X}^{\mathrm{A}}$.

Neither of these makes sense in general. Note also that the extreme cases are not precisely symmetric.

A principled method for choosing g is to consider the *loss of protection* incurred by the agencies, which we denote by LP(α) for Agency α. We measure loss of protection to one agency by the number of (linearly independent) constraints the other agency has on its data. Thus, for Agency A,

$$\mathrm{LP(A)} = p_{\mathrm{A}}p_{\mathrm{B}} + p_{\mathrm{A}}g \tag{4}$$

The first term in (4) represents B's knowledge of the $p_{\mathrm{A}\cdot\mathrm{B}}$ entries of $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$ at the end of the process. The second term reflects that B knows both $\mathbf{Z}$ and that $(\mathbf{X}^{A})^T\mathbf{Z} = 0$ – that is, (2), which contains $p_{\mathrm{A}} \times g$ constraints. One can also view LP(A) relative to the total "degrees of freedom" in $\mathbf{X}^{\mathrm{A}}$, which is $n \times p_{\mathrm{A}}$. Similarly,

$$\mathrm{LP(B)} = p_{\mathrm{A}}p_{\mathrm{B}} + p_{\mathrm{B}}(n - g) \tag{5}$$

The first term is the same as in (4) and the second term reflects that A knows that

$$\mathrm{rank}(\mathbf{W}) = n - g$$

(Indeed, this is why Agency A cannot invert $\mathbf{W}$ to obtain $\mathbf{X}^{\mathrm{B}}$). The total loss of protection, as a function of g, is then

$$\mathrm{LP}(g) = \mathrm{LP(A)} + \mathrm{LP(B)} = 2p_{\mathrm{A}}p_{\mathrm{B}} + np_{\mathrm{B}} + (p_{\mathrm{A}} - p_{\mathrm{B}})g \tag{6}$$

Note that when $p_{\mathrm{A}} = p_{\mathrm{B}}$, $\mathrm{LP}(g) = 2p_{\mathrm{A}}p_{\mathrm{B}} + np_{\mathrm{B}}$, no matter what the value of g. Harking back to the two extreme cases, when $p_{\mathrm{A}} = p_{\mathrm{B}}$, the total loss of protection is constant, and g affects only how that loss is distributed between Agency A and Agency B.

In general, it seems desirable that the loss of protection should be shared equally by the two agencies. To measure this, we introduce the *inequity*

$$\mathrm{I}(g) = |\mathrm{LP(A)} - \mathrm{LP(B)}| = |(p_\mathrm{A} + p_\mathrm{B})g - np_\mathrm{A}| \tag{7}$$

Setting **I**(g) to its minimal value of zero yields the optimal choice of g:

$$g^* = \frac{p_\mathrm{A}}{p_\mathrm{A} + p_\mathrm{B}} n \tag{8}$$

The value of $g*$ in (8) has a natural interpretation: Agencies A and B together possess $p_\mathrm{A} + p_\mathrm{B}$ attributes, so $p_\mathrm{A}/(p_\mathrm{A} + p_\mathrm{B})$ is Agency A's share of those attributes. When A has a larger share of attributes, it must surrender more information to B than *vice versa.*

2.3. Other Equity Issues

The optimal value $g*$ in (7) applies only to computation of $(\mathbf{X}^\mathrm{A})^T\mathbf{X}^\mathrm{B}$. This is not the only equity issue associated with computation of $\mathbf{X}^T\mathbf{X}$.

When the agencies have different numbers of attributes, perhaps the most glaring inequity is associated with the on-diagonal blocks of $(\mathbf{X}^\alpha)^T\mathbf{X}^\alpha$ of $\mathbf{X}^T\mathbf{X}$. The dimensions of $(\mathbf{X}^\alpha)^T\mathbf{X}^\alpha$ are $p_\alpha \times p_\alpha$, and is as shown in Figure 2, these blocks may differ substantially in size. Without even considering off-diagonal blocks, each agency α is surrendering p_α^2 constraints on its data $\mathbf{X}^\alpha$.

Also, the loss of privacy LP(A) in (4) is what Agency A loses to Agency B in the course of computing $(\mathbf{X}^\mathrm{A})^T\mathbf{X}^\mathrm{B}$. None of this is fully given up to any other agencies when $(\mathbf{X}^\mathrm{A})^T\mathbf{X}^\mathrm{B}$ is shared, because no agency other than B knows $\mathbf{X}^\mathrm{B}$. Other agencies, however, know different constraints on $\mathbf{X}^\mathrm{B}$, and second-order computations may be possible.

But Agency A must engage in calculation of $(\mathbf{X}^\mathrm{A})^T\mathbf{X}^\beta$ with *every other* Agency β, which does increase loss of privacy. However, under the assumption that agencies do not collude, the loss does not increase. Even if agencies do collude, Agency A can mitigate the effects by making the **Z** matrices it sends to other agencies subsets of one matrix. This means that the agency B for which $g*$ in (8) is largest learns the most about $\mathbf{X}^\mathrm{A}$, and what any other agency learns is a subset of this.

2.4. Other Threats to Privacy

The protocol in Section 2.1 is not immune to breaches of privacy. For instance, if the matrix **Z** in **Step 1** of the protocol is such that $(\mathbf{I} - \mathbf{Z}\mathbf{Z}^T)$ contains a column with all zeros except for a nonzero constant in one row, then Agency A learns from $(\mathbf{X}^\mathrm{A})^T\,\mathbf{W}$ the value of Agency B's data for the data subject in that row. Of course, this problem is detectable by Agency B, which could then simply not respond.

Even when the agencies are semi-honest, disclosures might be generated because of the values of the attributes themselves. Note that these issues are the result of computation of $\mathbf{X}^T\mathbf{X}$ by any method, and are neither caused nor alleviated by use of the protocol in Section 2.1.

As a simple example, suppose $\mathbf{X}^{\mathrm{A}}$ includes an attribute that equals zero for all but one of the data subjects. Even with a legitimate $\mathbf{Z}$, then $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$ will reveal that subject's value of $\mathbf{X}^{\mathrm{B}}$.

Similar problems arise if $\mathbf{X}^{\mathrm{A}}$ is sparse and there is reliable prior information on the locations of nonzero entries. In this case, the effective number of degrees of freedom in $\mathbf{X}^{\mathrm{A}}$ is less – perhaps much less – than $n \times p_{\mathrm{A}}$.

Still other issues can occur. For instance, attributes might satisfy constraints of the form "Gross income $\geq$ net income plus federal tax plus state tax," which have effectively unpreventable potential to reveal information. Similarly, if one record in $\mathbf{X}^{\mathrm{A}}$ contains a dominant attribute value (Willenborg and de Waal 2001), for instance to the extent that it exceeds 90% of the sum of all values of that attribute, then that value is revealed approximately in all of the $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$.

The implications of our methods in settings where data change over time (for example, longitudinal studies by official statistics agencies) have not been worked out. Clearly repeated application of our protocol to evolving databases reveals more and more information.

Finally, there may be problems that are revealed only when analyses are conducted. For example, Agency A can perform a regression of every other attribute on its attributes. Should one of those regressions have a high coefficient of determination, then A knows that it has in its own data a good predictor of that attribute, even if it is owned by another agency.

One way in which agencies might attempt to deal with such issues would be to share only some of their attributes. How this would work and whether it would be effective in preserving privacy are subjects for future research.

2.5. Other Protocols for Secure Matrix Multiplication

The protocol in Section 2.1 is not the only protocol available for secure matrix multiplication. It differs from other protocols, however, in two important respects. The first of these is the use of the inequity in (7) as a means of determining the value of *g,* using (8). The second is that our protocol has extremely low communication overhead.

The technique introduced in Du and Atallah (2001) utilizes the "1 of out *N*" oblivious transfer protocol. While it is arguably more secure, there is an extremely high communication cost. In this approach, for computation of $(\mathbf{X}^{\mathrm{A}})^T\mathbf{X}^{\mathrm{B}}$, Agencies A and B agree on two numbers k and m such that k^m is very big. The protocol repeats m times, each involving k transfers of a $p_{\mathrm{A}} \times n$ matrix and "1 out of k" transfer of a matrix of size $p_{\mathrm{A}} \times p_{\mathrm{B}}$. Our approach involves only the transfer of the matrices $\mathbf{Z}$ and $\mathbf{W}$, and is substantially more efficient.

Du et al. (2004) proposed a secure matrix product protocol that is more similar to ours. In their approach, the two agencies generate an invertible matrix $\mathbf{M}$ jointly. Then A passes $(\mathbf{X}^{\mathrm{A}})^T\mathbf{M}_{\mathrm{left}}$ to B and B passes $(\mathbf{X}^{\mathrm{B}})^T\mathbf{M}_{\mathrm{top}}^{-1}$ to A. Here $\mathbf{M}_{\mathrm{left}}$ is the left half of the $\mathbf{M}$ matrix and $\mathbf{M}_{\mathrm{top}}^{-1}$ is the top half of $\mathbf{M}^{-1}$. By doing so Agencies A and B achieve roughly the same loss of protection (see Section 2.2) value as we obtain. There are, though, several advantages to our approach. In particular, the invertible matrix $\mathbf{M}$ needs to be agreed on by both parties, which entails substantial communication cost when n is large. By contrast, in our protocol, $\mathbf{Z}$ is simply generated by Agency A. Second and more important, inequity as operationalized by (7) provides a principled way of choosing g.

3. Performing Statistical Analyses

Once the full data covariance matrix has been calculated as described in Section 2, it is possible securely to conduct linear regression analyses, as well as stepwise regression, ridge regression and model diagnostics. We show how to do so in this section.

Note that any *one specific* analysis (for example, a linear regression involving one response and only a subset of the predictors) may require only a (square) submatrix of the full covariance matrix. The same is true of a *prespecified* small set of regressions. If this were the case, the agencies would both increase protection and save computation as well as communication overhead by computing only an appropriate submatrix of $\mathbf{X}^T\mathbf{X}$.

3.1. Secure Linear Regression

For clarity, we now write the $(n \times p)$-dimensional data matrix $\mathbf{X}$ of (1) as

$$\mathbf{X} = [X_1 . . . X_p] \tag{9}$$

where each X_i has dimension $n \times 1$, and belongs to exactly one of the $\mathbf{X}^\alpha$. To account for intercepts in regressions, we assume the first column of $\mathbf{X}$ in (9) consists entirely of ones (and is owned by Agency A, although that is not material). For simplicity assume that all other attribute means are zero; this is not restrictive, since if the agencies are willing to share the on-diagonal blocks of the covariance matrix, they would certainly be willing to share means.

Assume that $\mathbf{X}^T\mathbf{X}$ is calculated using the method described in Section 2. Following that computation, some checking may be necessary. For instance, it is possible that two agencies hold perfectly correlated attributes without knowing it. Simple issues of this sort are detectable from $\mathbf{X}^T\mathbf{X}$, and readily addressed. More complex issues (for instance, complex multi-agency multi-collinearity) are more problematic.

A regression model of a response attribute $X_{\text{resp}} \in \{X_1, . . . , X_p\}$ on a set of predictors $\mathbf{X}_{\text{pred}} \subseteq \{X_0, . . . , X_p\}/\{X_{\text{resp}}\}$ is of the form

$$X_{\text{resp}} = \boldsymbol{X}_{\text{pred}}\beta + \varepsilon \tag{10}$$

where $\varepsilon \sim N(0, \sigma^2)$. The maximum likelihood estimates of β and σ^2, as well as the standard errors of the estimated coefficients, can be easily obtained from $\mathbf{X}^T\mathbf{X}$, for example using the sweep algorithm (Beaton 1964; Schafer 1997). Indeed, only the restriction of $\mathbf{X}^T\mathbf{X}$ to $\mathbf{X}_{\text{pred}} \cup \{X_{\text{resp}}\}$ is needed.

3.2. Model Diagnostics

Estimated regression coefficients are of limited value when a regression model such as (10) does not describe the data adequately. Hence, model diagnostics are essential. The types of diagnostic measures available in vertically partitioned data settings depend on what additional information the agencies are willing to share. Diagnostics based on residuals require the predicted values

$$\widehat{X_{resp}} = \mathbf{X}_{\text{pred}}\hat{\beta} = \mathbf{X}_{\text{pred}}\left[\mathbf{X}_{\text{pred}}^{T}\mathbf{X}_{\text{pred}}\right]^{-1}\mathbf{X}_{\text{pred}}^{T}X_{\text{resp}} \tag{11}$$

These can be calculated using the secure matrix multiplication protocol of Section 2.1. Alternatively, since each agency can calculate $\hat{\beta}$ from $\mathbf{X}^{T}\mathbf{X}$, each can compute that portion of $\mathbf{X}_{\text{pred}}\hat{\beta}$ associated with its attributes, and these vectors can be summed across agencies using secure summation (Benaloh 1987).

Once the predicted values are known, the agency owning the response attribute X_{resp} can calculate the residuals $X_{\text{resp}} - X_{\text{pred}}\hat{\beta}$. If that agency is willing to share these with the other agencies, each agency can perform plots of residuals and report the nature of any lack of fit to the other agencies. Sharing residuals $X_{\text{resp}} - X_{\text{pred}}\hat{\beta}$ also enables all agencies to obtain Cook's distance measures (Cook and Weisberg 1982), since these are solely a function of the residuals and the diagonal elements of the hat matrix $\mathbf{H} = \mathbf{X}_{\text{pred}}\left[\mathbf{X}_{\text{pred}}^{T}\mathbf{X}_{\text{pred}}\right]^{-1}\mathbf{X}_{\text{pred}}^{T}$. We note that the diagonal elements of $\mathbf{H}$ can be used as well to generate standardized and Studentized residuals. Additionally, the agency with X_{resp} can make a plot of the residuals versus predicted values, and a normal quantile plot of the residuals, and report any evidence of model violations to the other agencies. The number of residuals exceeding certain thresholds, i.e., outliers, can also be reported.

3.3. *Other Analyses*

The approach outlined in Sections 3.1 and 3.2 extends readily to other classes of statistical analyses, although the issues raised in Sections 2.3 and 2.4 need to be considered in detail in each instance.

A simple example is weighted least squares regression. If $\mathbf{T}$ is the $n \times n$ (diagonal) matrix of weights, then each agency premultiplies its attributes by $\mathbf{T}^{1/2}$, and the analysis proceeds as described in Sections 3.1 and 3.2.

To run semi-automatic model selection procedures such as stepwise regression, the agencies can calculate covariance matrices securely, then select models based on criteria that are functions of the full covariance matrix $\mathbf{X}^{T}\mathbf{X}$, such as the F-statistic or the Akaike Information Criterion.

It is also possible to perform ridge regression (Hoerl and Kennard 1970) securely. Ridge regression shrinks estimated regression coefficients away from the maximum likelihood estimates by imposing a penalty on their magnitude. Written in matrix form, ridge regression seeks $\hat{\beta}$ that minimizes

$$\text{Ridge}(\beta; \lambda) = \left(X_{\text{resp}} - \mathbf{X}_{\text{pred}}\beta\right)^{T}\left(X_{\text{resp}} - \mathbf{X}_{\text{pred}}\beta\right) + \lambda\beta^{T}\beta \tag{12}$$

where λ is a specified constant. The ridge regression estimate of the coefficients is

$$\hat{\beta}_{\text{Ridge}} = \left(\mathbf{X}_{\text{pred}}^{T}\mathbf{X}_{\text{pred}} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}_{\text{pred}}^{T}\mathbf{X}_{\text{resp}} \tag{13}$$

Once $\mathbf{X}_{\text{pred}}^{T}\mathbf{X}_{\text{pred}}$ and $X_{\text{pred}}^{T}X_{\text{resp}}$ have been calculated securely, each agency can perform the calculation in (13).

4. Conclusion

Using our linear algebra-based approach, it is possible for statistical agencies and other data holders to obtain matrix products in vertically partitioned data settings. This enables agencies with vertically partitioned data to perform linear regressions without sharing their data values. We anticipate that the secure matrix protocol will be useful for other techniques that depend on sample covariance matrices, such as some forms of cluster and discriminant analysis. Future research areas include protocols for sharing nonlinear analyses securely, the potential of data encryption in vertically partitioned data, methods for matching records securely, and assessing disclosure risk to data subjects.

Appendix

Generating Z *from* $\boldsymbol{X}^{\mathrm{A}}$

Step 1 of the secure matrix product protocol of Section 2.1 requires vectors $\{Z_1, Z_2,\dots Z_g : Z_i \in \mathbb{R}^n\}$ such that $Z_i^T X_j^{\mathrm{A}} = 0$ for all i and j. These can be generated using the QR-decomposition of $\mathbf{X}^A$, which—recall that $\mathbf{X}^A$ is $(n \times p_A)$-dimensional—is given by

$$\mathbf{X}^{\mathrm{A}} = \mathbf{QR}$$

where $\mathbf{Q}$ is an $(n \times n)$ orthonormal matrix and $\mathbf{R}$ is an $(n \times p)$ upper-triangular matrix. See Golub and Van Loan (1996) and Press et al. (1992) for details on properties of and algorithms for the QR-decomposition. The calculation is both fast and numerically accurate. To construct $\mathbf{Z}$, partition $\mathbf{Q}$ columnwise as

$$\mathbf{Q} = [\mathbf{Q}_1\mathbf{Q}_2]$$

where $\mathbf{Q}_1$ consists of the leftmost p_{A} columns of $\mathbf{Q}$. Then, with ran($\mathbf{M}$) denoting the range of a matrix $\mathbf{M}$,

$$\mathrm{ran}(\mathbf{X}^{\mathrm{A}}) = \mathrm{ran}(\mathbf{Q}_1)$$

and

$$\mathrm{ran}(\mathbf{X}^{\mathrm{A}})^{\perp} = \mathrm{ran}(\mathbf{Q}_2)$$

Hence $\mathbf{Z}$ can be easily obtained by selecting (randomly or informatively) g columns of $\mathbf{Q}_2$.

While it may seem that orthonormality of the $\mathrm{Z_i}$ poses a risk to Agency A, this is not so, since Agency B can always calculate a set of orthonormal vectors with the same span, using the Graham-Schmidt procedure.

If Agency A fears that Agency B's knowing that a QR-decomposition was used reveals extra information, it can permute the columns of $\mathbf{X}^{\mathrm{A}}$ before doing the decomposition, and permute the columns of $\mathbf{Z}$ correspondingly before sending $\mathbf{ZZ}^T$ to Agency B.

6. References

Agrawal, R. and Srikant, R. (2000). Privacy-preserving Data Mining. Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 439–450.

Beaton, A.E. (1964). The Use of Special Matrix Operations in Statistical Calculus. Research Bulletin RB-64-51. Princeton, NJ: Educational Testing Service.

Benaloh, J. (1987). Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret. CRYPTO86, A.M. Odlyzko (ed.).: Springer-Verlag, Lecture Notes in Computer Science No. 263, 251–260.

Clifton, C., Doan, A., Elmagarmid, A., Kantarcioglu, M., Schadow, G., Suciu, D., and Vaidya, J. (2003a). Privacy-preserving Data Integration and Sharing. Presented at 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, Paris, June 13, 2004.

Clifton, C., Kantarcioglu, M., Lin, X., Vaidya, J., and Zhu, M. (2003b). Tools for Privacy-Preserving Distributed Data Mining. KDD Explorations, 4, 28–34.

Cook, R.D. and Weisberg, S. (1982). Residuals and Influence in Regression. London: Chapman & Hall.

Dobra, A., Fienberg, S.E., Karr, A.F., and Sanil, A.P. (2002). Software Systems for Tabular Data Releases. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, 10, 529–544.

Dobra, A., Karr, A.F., and Sanil, A.P. (2003). Preserving Confidentiality of High-Dimensional Tabular Data: Statistical and Computational Issues. Statistics and Computing, 13, 363–370.

Doyle, P., Lane, J., Theeuwes, J.J.M., and Zayatz, L.V. (2001). Confidentiality, Disclosure and Data Access: Theory and Practical Application for Statistical Agencies. Amsterdam: Elsevier.

Du, W. and Atallah, M.J. (2001). Privacy-Preserving Cooperative Scientific Computations. 14th Computer Security Foundations Workshop. New York: IEEE Press, 273–282.

Du, W., Han, Y., and Chen, S. (2004). Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. Proceedings of the 4th SIAM International Conference on Data Mining, 222–233.

Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J. (2004). Privacy Preserving Mining of Association Rules. Information Systems, 29, 343–364.

Fellegi, I.P. and Sunter, A.B. (1969). A Theory for Record Linkage. Journal of the American Statistical Association, 64, 1183–1210.

Ghosh, J., Reiter, J.P., and Karr, A.F. (2007). Secure Computation with Horizontally Partitioned Data Using Adaptive Regression Splines. Computational Statistics and Data Analysis, 51, 5813–5820.

Golub, G. and Van Loan, C. (1996). Matrix Computations, (3rd ed.). Baltimore: Johns Hopkins University Press.

Gomatam, S., Karr, A.F., Reiter, J.P., and Sanil, A.P. (2005a). Data Dissemination and Disclosure Limitation in a World Without Microdata: A Risk-Utility Framework for Remote Access Analysis Servers. Statistical Science, 20, 163–177.

Gomatam, S., Karr, A.F., and Sanil, A.P. (2005b). Data Swapping as a Decision Problem. Journal of Official Statistics, 21, 635–656.

Hoerl, A.E. and Kennard, R. (1970b). Ridge Regression: Biased Estimation for Nonorthogonal Problems. Technometrics, 12, 55–67.

Kantarcioglu, M. and Clifton, C. (2002). Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. Proceedings of the 2002 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 24–31.

Karr, A.F., Feng, J., Lin, X., Reiter, J.P., Sanil, A.P., and Young, S.S. (2005a). Secure Analysis of Distributed Chemical Databases Without Data Integration. Journal of Computer-Aided Molecular Design, 2005, 1–9, November.

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, 335–345.

Karr, A.F., Kohnen, C.N., Oganian, A., Reiter, J.P., and Sanil, A.P. (2006a). A Framework for Evaluating the Utility of Data Altered to Protect Confidentiality. The American Statistician, 60, 224–232.

Karr, A.F., Lin, X., Reiter, J.P., and Sanil, A.P. (2004). Analysis of Integrated Data Without Data Integration. Chance, 17, 26–29.

Karr, A.F., Lin, X., Reiter, J.P., and Sanil, A.P. (2005b). Secure Regression on Distributed Databases. Journal of Computational and Graphical Statistics, 14, 263–279.

Karr, A.F., Sanil, A.P., and Banks, D.L. (2006b). Data Quality: A Statistical Perspective. Statistical Methodology, 3, 137–173.

Lin, X., Clifton, C., and Zhu. Y. (2004). Privacy Preserving Clustering with Distributed EM Mixture Modeling. Knowledge and Information Systems. Published on-line on December, 23.

Lindell, Y. and Pinkas, B. (2000). Privacy Preserving Data Mining. Advances in Cryptopoly – Crypto2000, Lecture Notes in Computer Science, Vol. 1880. New York: Springer-Verlag, 20–24.

Press, W.H., Teulosky, S.A., Vetterling, W.T., and Flannery, B.P. (1992). Numerical Recipes in C: The Art of Scientific Computing, (2nd ed.). Cambridge, UK: Cambridge University Press.

Reiter, J.P., Karr, A.F., Kohnen, C.N., Lin, X., and Sanil, A.P. (2004). Secure Regression for Vertically Partitioned, Partially Overlapping Data. Proceedings of the American Statistical Association.

Sanil, A.P., Karr, A.F., Lin, X., and Reiter, J.P. (2004). Privacy Preserving Regression Modelling via Distributed Computation. Proceedings of the Tenth ACM SIGKDD International Conference Knowledge Discovery and Data Mining, 677–682.

Schadow, C., Grannis, S., and McDonald, C. (2002). Privacy-Preserving Distributed Queries for a Clinical Case Research Network. Privacy, Security and Data Mining. Volume 14 of Conferences in Research and Practice in Information Technology, C. Clifton and V. Estivill-Castro (eds). Sydney: Australian Computer Society, 55–65.

Schafer, J.L. (1997). Analysis of Incomplete Multivariate Data. London: Chapman & Hall.

Vaidya, J. and Clifton, C. (2002). Privacy Preserving Association Rule Mining in Vertically Partitioned Data. Proceedings of the Eighth ACM SIGKDD International

Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 639–644.

Vaidya, J. and Clifton, C. (2003). Privacy Preserving *k*-Means Clustering over Vertically Partitioned Data. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 206–215.

Vaidya, J. and Clifton, C. (2004). Privacy-Preserving Data Mining: Why, How and What for? Security and Privacy Magazine, 2, 18–27.

Willenborg, L.C.R.J. and de Waal, T. (2001). Elements of Statistical Disclosure Control. New York: Springer-Verlag.

Yao, A.C. (1982). Protocols for Secure Computations. Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science. New York: ACM Press, 160–164.

Received October 2004
Revised September 2008