Privacy Preserving Regression Modelling Via Distributed Computation

Ashish P. Sanil, Alan F. Karr and Xiaodong Lin
National Institute of Statistical Sciences
PO Box 14006
Research Triangle Park, NC 27709-4006
ashish@niss.org, karr@niss.org, linxd@samsi.info

Jerome P. Reiter
Duke University
Durham, NC 27708 USA
jerry@stat.duke.edu

ABSTRACT

Reluctance of data owners to share their possibly confidential or proprietary data with others who own related databases is a serious impediment to conducting a mutually beneficial data mining analysis. We address the case of vertically partitioned data – multiple data owners/agencies each possess a few attributes of every data record. We focus on the case of the agencies wanting to conduct a linear regression analysis with complete records without disclosing values of their own attributes. This paper describes an algorithm that enables such agencies to compute the *exact* regression coefficients of the global regression equation and also perform some basic goodness-of-fit diagnostics while protecting the confidentiality of their data. In more general settings beyond the privacy scenario, this algorithm can also be viewed as method for the distributed computation for regression analyses.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning.

General Terms

Algorithms, Security.

Keywords

Data confidentiality, data integration, secure multi-party computation, regression.

1. INTRODUCTION

In numerous contexts immense utility can arise from statistical analyses that "integrate" multiple, distributed databases. These analyses would be more informative than individual analyses, *i.e.*, it would enable us to fit models involving more attributes and/or estimate models more accurately (with lower standard errors of estimates). At the same time, concerns about data confidentiality pose strong legal, regulatory or even physical barriers to literally integrating the databases. These concerns are present even if the database "owners" are cooperating: they wish to perform the analysis, and none of them is specifically interested in breaking the confidentiality of any of the others' data. This need to balance the utility of better combined analyses with the risk of privacy violation has received considerable interest lately [1, 5]. Specifically, consider two cases

- *Vertically partioned data* When multiple parties have data on certain data subjects but each party only posseses data on different sets of attributes of those entities. *E.g.*, a government agency might have employment information, another health data, and a third information about education. A regression analysis on an integrated database would be more informative and powerful than, or at least complementary to, individual analyses.

- *Horizontally partioned data* When the participating agencies have databases that contain the same numerical attributes for disjoint sets of data subjects. *E.g.*, several State Departments of Education might want to combine their student data in order to conduct a more accurate analysis of student performance for the general student population.

The results of such analyses may be either used by the database owners themselves or disseminated more widely.

In this paper, we show how to perform secure linear regression for "vertically partitioned data". (The problem of horizontally partitioned data is addressed in a companion paper [9]). Our work is similar in spirit to [12, 13] who describe methods for doing cluster analysis and association rule discovery in vertically partitioned data. The recent work by Du, *et al* [6] for the 2-agency setting also presents a complementary approach for conducting secure linear regressions. The practical implentation of the algorithm we present is simpler since it can be implemented solely as secure data transfer among the agencies (it does not rely on circuit representations which implicitly involve a trusted – possibly software – third party), and it also permits some basic model

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'04 August 22–25, 2004, Seattle, Washington, USA.
Copyright 2004 ACM 1-58113-888-1/04/0008 ...$5.00.

validation. However, Du, *et al*'s method does permit a much better characterization of the degree of private information revealed. Their method is also applicable in the two-agency setting, whereas ours requires at least three participating agencies. A related stream of research concerns the analysis of data in distributed databases (unencumbered by privacy concerns). There is a fairly extensive literature on this topic [8]. Some work has also been done on linear regression in distributed data settings [7, 15], and it might be possible to extend some of these methods so that privacy is preserved.

We view a group of government agencies seeking to perform a combined analysis on their data as a setting that reflects well the semi-honest data sharing scenario we deal with. Hence, we term the participants "agencies" even though in some settings they might be corporations or other data holders. In Section 2 we outline the privacy preserving regression problem. This is followed, in Section 3, by a brief description of Powell's method for numerical minimization and a secure summation protocol that together form building blocks of our procedure. Section 4 contains a description of the main algorithm and Section 5 discusses what has been revealed by using the procedure as well as what the agencies collectively learn (including possible paths for conducting regression analyses). We end with concluding remarks in Section 6.

2. THE REGRESSION PROBLEM

Consider the case when we need to fit the standard linear regression model [14]

$$\mathbf{y} = X\beta + \epsilon, \tag{1}$$

where

$$X = [\mathbf{x}_1 \dots \mathbf{x}_p], \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \tag{2}$$

with

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{in} \end{bmatrix}, \tag{3}$$

and

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}, \qquad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}. \tag{4}$$

Under the condition that

$$\varepsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{5}$$

The least-squares estimate (which is also the maximum likelihood estimate) for β is obtained as the minimizer of the the "sum of squared errors" function

$$E(\beta) = (\mathbf{y} - X\beta)^T(\mathbf{y} - X\beta). \tag{6}$$

The function in equation (6) is a quadratic in β and the minimizer is well-known and readily calculated as

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{7}$$

We will assume the following scenario: There are several agencies interested in computing this regression equation, but each agency only possesses part of the data. $K > 2$ agencies, $A_1, A_2, \dots, A_K$, are involved. Agency A_j possesses d_j columns of the predictor attributes ($\mathbf{x}$'s), and I_j denotes the index set of A_j's predictors. In addition, we assume that all agencies know the "response" attribute, $\mathbf{y}$. (We believe this is not strictly necessary; but if this is not so, we need to add another layer of security. The details of this are quite complex and they also raise some additional subtle issues, such as information asymmetry. We will deal with these issues in a separate paper.) Also, if $\mathbf{u}$ has components $(u_1, \dots, u_m)$, we will use u_{I_j} as shorthand for $\{u_i\}_{i \in I_j}$. The following example clarifies this notation.

EXAMPLE 1. *If $K = 3$ agencies are involved, and if agency A_1 knows $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, A_2 knows $\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$, and A_3 knows $\mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9$. Then $d_1 = d_2 = d_3 = 3$, $I_1 = \{1, 2, 3\}$ $I_2 = \{4, 5, 6\}$ and $I_3 = \{7, 8, 9\}$. Also,*

$$X = [\overbrace{\mathbf{x}_1\, \mathbf{x}_2\, \mathbf{x}_3}^{X_{I_1}}\ \overbrace{\mathbf{x}_4\, \mathbf{x}_5\, \mathbf{x}_6}^{X_{I_2}}\ \overbrace{\mathbf{x}_7\, \mathbf{x}_8\, \mathbf{x}_9}^{X_{I_3}}]$$

and

$$\beta^T = (\overbrace{\beta_1, \beta_2, \beta_3}^{\beta_{I_1}}\ \overbrace{\beta_4, \beta_5, \beta_6}^{\beta_{I_2}}\ \overbrace{\beta_7, \beta_8, \beta_9}^{\beta_{I_3}}).$$

We consider the case where agencies $A_1, A_2, \dots, A_K$ collectively wish to compute β without sharing their possibly confidential data (they are also assumed to be unwilling to share summary statistics that relate their data to the other agencies' data, such as correlations between attributes). Calculation of β using equation (6) requires the sharing of at least some summary statistics. We develop a strategy for a distributed computation of β using direct numerical minimization of $E(\beta)$. In our scheme, each agency A_j will be able to obtain its component $\hat{\beta}_{I_j}$ of the global estimate $\hat{\beta}$ without revealing its data to any of the other agencies. It is assumed that all the agencies will share their $\hat{\beta}_{I_j}$ with the other agencies so that everyone benefits from the analysis (and also so that everyone has some incentive to participate in this exercise). In addition to $\hat{\beta}_{I_j}$, all agencies also learn the vector of residuals, $\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\beta}$, as a by-product of our procedure. They could use $\hat{\mathbf{e}}$ to conduct basic diagnostic tests about the regression model.

We now note some finer points related to our setup before proceeding with the details.

Remark 1: As in other data sharing protocols, we require one agency to assume a lead role in initiating and coordinating the process. This is a purely administrative role and doesn't imply any information advantage or disadvantage. We will assume that Agency 1 is the designated leader.

Remark 2: The databases need to have a common primary key that enables the agencies to align the records correctly in the same order (possibly under direction from Agency 1).

Remark 3: We assume that the attribute sets do not overlap ($I_j \cap I_k = \emptyset$). If any attributes overlap, *i.e.*, if more than one agency posseses the same attribute, we can designate one of the owning agencies as the designated "owner". This is not a problem since the agencies share all the β's at the end of the estimation process. On a related note: regression models such

as (1) will typically include a constant or "intercept" term. This is equivalent to one of the $\mathbf{x}$'s being a column of 1's. Without loss of generality, we will assume that one of the attributes is $\mathbf{x}^T = (1, 1, \ldots, 1)$ and that it is "owned" by Agency 1.

3. PRELIMNARIES

We now provide some background on a numerical minimization technique that forms the crux of our proposed method, and a protocol to compute secure sums that is an essential component of our method.

3.1 Powell's Method for a Quadratic function of *p* variables

Powell's algorithm [10] for finding the minimizer of a function of many variables forms the basis of our proposed procedure. We will use it to directly find $\hat{\beta}$ as a numerical solution to the problem $\arg\min_{\beta \in \Re^p} E(\beta)$ (from equation (6)) in a manner such that the agencies are not required to share their data. Powell's method is a derivative-free numerical minimization method that solves the multidimensional minimization problem by solving a series of 1-dimesional line minimization problems. A very high-level description of the algorithm is as follows. Start with a set of suitably chosen set of p vectors in $\Re^p$ which will serve as "search directions". Start at an arbitrary starting point in $\Re^p$ and determine the step size δ along the first search direction $\mathbf{s}^{(1)}$ that will minimize the objective function (this is a 1-dimensional minimization problem). We then move distance δ along $\mathbf{s}^{(1)}$. Then move an optimal step in the second search direction $\mathbf{s}^{(2)}$, and so on until all the search directions are exhausted. After that, appropriate updates are made to the set of search directions, and the iterations continue until the minimum is obtained. Specifically, the procedure for finding the minimizer of a function $E(\beta)$ consists of an initialization step and an iteration block as described below.

Initialization : Select an arbitrary orthogonal basis[1] for $\Re^p$: $\mathbf{s}^{(1)}, \mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(1)} \in \Re^p$. Also pick an arbitrary starting point $\tilde{\beta} \in \Re^p$.

Iteration : Repeat the following block of steps p times.

- Set $\beta \leftarrow \tilde{\beta}$.
- For $i = 1, 2, \ldots, p$:
  - Find δ that minimizes $f(\beta + \delta \mathbf{s}^{(i)})$.
  - Set $\beta \leftarrow \beta + \delta \mathbf{s}^{(i)}$.
- For $i = 1, 2, \ldots, (p-1)$: Set $\mathbf{s}^{(i)} \leftarrow \mathbf{s}^{(i+1)}$.
- Set $\mathbf{s}^{(p)} \leftarrow \beta - \tilde{\beta}$.
-
  - Find δ that minimizes $f(\beta + \delta \mathbf{s}^{(p)})$.
  - Set $\tilde{\beta} \leftarrow \beta + \delta \mathbf{s}^{(p)}$.

Note that each iteration of the iteration block involves solving $(p+1)$ 1-dimensional line minimization problems (to determine the δ's). Powell established the remarkable result that if $f(\beta)$ is a quadratic function, then exactly p iterations of the iteration block would yield the *exact minimizer* of $f(\beta)$! (This involves solving $p(p+1)$ line minimization problems to obtain the minimizer of a quadratic function.) We refer the reader to [3, 10, 11] for proofs and elaborations.

[1]Powell's original algorithm used the coordinate axis vectors $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ as the basis. Brent [3] shows that an arbitrary orthogonal basis also suffices.

Note

In our regession case (equation (6)), the δ that minimizes $E(\beta + \delta \mathbf{s}^{(i)}) = (\mathbf{y} - X(\beta + \delta \mathbf{s}^{(i)}))^T (\mathbf{y} - X(\beta + \delta \mathbf{s}^{(i)}))$, is readily obtained as

$$\delta = \frac{(\mathbf{y} - X\beta)^T X \mathbf{s}^{(i)}}{(X\mathbf{s}^{(i)})^T X \mathbf{s}^{(i)}}. \tag{8}$$

3.2 Secure Summation

Consider $K > 2$ cooperating, such that Agency j has a value v_j, and suppose that the agencies wish to calculate $v = \sum_{j=1}^{K} v_j$ in such a manner that each Agency j can learn only the minimum possible about the other agencies' values, namely, the value of $v_{(-j)} = \sum_{\ell \neq j} v_\ell$. The secure summation protocol [2, 4] can be used to effect this computation.

Choose m to be a very large number which is known to all the agencies such that v is known to lie in the range $[0, m)$. Agency 1 is assumed to be the leader. The remaining agencies are numbered $2, \ldots, K$. Agency 1 generates a random number R, chosen uniformly from $[0, m)$. Agency 1 adds R to its local value v_1, and sends the sum $s_1 = (R + v_1) \bmod m$ to Agency 2. Since the value R is chosen uniformly from $[0, m)$, Agency 2 learns nothing about the actual value of v_1.

For the remaining agencies $j = 2, \ldots, k-1$, the algorithm is as follows. Agency j receives

$$s_{j-1} = (R + \sum_{s=1}^{j-1} v_s) \bmod m,$$

from which it can learn nothing about the actual values of $v_1, \ldots, v_{j-1}$. Agency j then computes and passes on to Agency $j+1$

$$s_j = (s_{j-1} + v_j) \bmod m = (R + \sum_{s=1}^{j} v_s) \bmod m.$$

Finally, agency K adds v_K to $s_{K-1} (\bmod m)$, and sends the result s_K to agency 1. Agency 1, which knows R then calculates v by subtraction:

$$v = (s_K - R) \bmod m$$

and shares this value with the other agencies. This method for secure summation faces an obvious problem if, contrary to our assumption, some agencies collude. However, there exist collusion-resistant versions of secure summation that involve each agency partitioning their constribution to the sum and all agencies conducting multiple rounds of the summation process to obtain the required sum [4, 9] (the order of the agencies could also be permuted to add an extra layer of security)[2].

4. ALGORITHM FOR THE DISTRIBUTED COMPUTATION OF THE REGRESSION COEFFICIENTS

Our algorithm is essentially Powell's algorithm implemented in such a manner that each agency A_j updates its own components of the β's and its own components of the search directions based on the data attributes it owns and one n-dimensional vector common to all agencies that is computed using secure summation. The details are as follows.

[2]We thank two referees for pointing this out.

1. Let $\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(p)} \in \Re^p$ be p-dimensional vectors that will serve as a set of search directions in $\Re^p$ which we will use for finding the optimal estimate $\hat{\beta}$. The $\mathbf{s}^{(r)}$ will be initially chosen and later updated in such a manner that A_j knows only the $\mathbf{s}^{(r)}_{I_j}$ components of each $\{\mathbf{s}^{(r)}\}_{r=1}^{p}$.

2. Initially, $\mathbf{s}^{(r)}$ are chosen as follows. Each A_j picks an orthogonal basis for $\Re^{d_j}$: $\{\mathbf{v}^{(r)}\}_{r \in I_j}$. Then for $r \in I_j$ let $\mathbf{s}^{(r)}_{I_j} = \mathbf{v}^{(r)}$, and $s^{(r)}_l = 0$ for $l \notin I_j$. Each agency should pick its basis at random so that the other agencies cannot obviously guess it.

   EXAMPLE 2. *In the setting of Example 1, if the initial search directions were written as columns of a matrix $S = [\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(p)}]$, then S would have the form*

$$\begin{pmatrix} s_1^{(1)} & s_1^{(2)} & s_1^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 \\ s_2^{(1)} & s_2^{(2)} & s_2^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 \\ s_3^{(1)} & s_3^{(2)} & s_3^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_4^{(4)} & s_4^{(5)} & s_4^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_5^{(4)} & s_5^{(5)} & s_5^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_6^{(4)} & s_6^{(5)} & s_6^{(6)} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_7^{(7)} & s_7^{(8)} & s_7^{(9)} \\ 0 & 0 & 0 & 0 & 0 & 0 & s_8^{(7)} & s_8^{(8)} & s_8^{(9)} \\ 0 & 0 & 0 & 0 & 0 & 0 & s_9^{(7)} & s_9^{(8)} & s_9^{(9)} \end{pmatrix}$$

   Where the non-zero diagonal blocks are each orthogonal bases for $\Re^3$ picked by A_1, A_2, A_3.
   Thus, $\{\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, \ldots, \mathbf{s}^{(p)}\}$ constitutes an orthogonal basis for $\Re^p$.

3. Let $\tilde{\beta} = \left(\tilde{\beta}_{I_1}, \tilde{\beta}_{I_2}, \ldots, \tilde{\beta}_{I_k}\right) \in \Re^p$ be the initial starting value of β obtained by each A_j picking $\tilde{\beta_{I_j}}$ arbirarily.

4. Perform the **Basic Iteration Block** below p times. The final value of $\tilde{\beta}$ will be exactly the optimal estimate $\hat{\beta}$.

The Basic Iteration

1. Each A_j sets $\beta_{I_j} \leftarrow \tilde{\beta}_{I_j}$.

2. For $r = 1, 2, \ldots, p$:

   (a) Each A_j computes $X_{I_j}\beta_{I_j}$ and $X_{I_j}\mathbf{s}^{(r)}_{I_j}$.

   (b) $\mathbf{z} = \mathbf{y} - X\beta = \mathbf{y} - \sum_{j=1}^{k} X_{I_j}\beta_{I_j}$ and $\mathbf{w} = X\mathbf{s}^{(r)} = \sum_{j=1}^{k} X_{I_j}\mathbf{s}^{(r)}_{I_j}$ are computed collectively by $A_1, A_2, \ldots, A_k$ using the secure summation protocol to compute the sums $\sum_{j=1}^{k} X_{I_j}\beta_{I_j}$ and $\sum_{j=1}^{k} X_{I_j}\mathbf{s}^{(r)}_{I_j}$. (Except in the first iteration of this block when, for a given r, $X_{I_j}\mathbf{s}^{(r)}_{I_j}$ is non-zero only for the agency who owns $\mathbf{x}_r$. Revealing this to all agencies is too risky, so only that particular agency, say A_r, will compute $\mathbf{w}$, but not reveal it to the others.)

   (c) All parties compute $\delta = \mathbf{z}^T\mathbf{w}/\mathbf{w}^T\mathbf{w}$. (Except in the first iteration, when A_r computes this and announces it to the rest.)

   (d) Each A_j updates $\beta_{I_j} \leftarrow \beta_{I_j} + \delta \cdot \mathbf{s}^{(r)}_{I_j}$.

3. For $r = 1, 2, \ldots, (p-1)$: Each A_j updates $\mathbf{s}^{(r)}_{I_j} \leftarrow \mathbf{s}^{(r+1)}_{I_j}$.

4. Each A_j updates $\mathbf{s}^{(p)}_{I_j} \leftarrow \beta_{I_j} - \tilde{\beta}_{I_j}$.

5. $\mathbf{z}, \mathbf{w}$ and δ are computed as before, and each A_j's updates $\beta_{I_j} \leftarrow \beta_{I_j} + \delta \cdot \mathbf{s}^{(p)}_{I_j}$.

5. DISCUSSION

5.1 What is Revealed

Consider any one step of the iteration, the only common information exchanged by the agencies are the $\mathbf{z}$ and $\mathbf{w}$ vectors. *Since each component of the vectors is computed using secure summation, the sources of disclosure threat for each component is the same as in using the secure summation protocol.* However, the actual risk to the data $\mathbf{x}$ is less since there is some masking with components of the $\mathbf{s}$ vectors. Specifically, the vulnerability is highest in the first step of the iteration since (due to the way we have chosen the initial $\mathbf{s}$) only one agency contributes to the sum $\mathbf{w}$ at each round of the basic iteration block. We avoid risk of disclosure by having on the contributing agency compute δ privately and announcing it to the others. If we assume that the agencies select their initial bases completely at random so that it is impossible for the others to guess it, and if the summation has been properly conducted in a secure manner according to the secure-summation protocol, then no private information is revealed if only $\mathbf{z}$ and $\mathbf{w}$ are common knowledge. If each iteration were independent, then the procedure is clearly secure. However, the values that each agency contributes to the sum are functionally related from one iteration to the next. This relation is complex and difficult to easily express. We expect that this complexity combined with the form of the secure sum protocol will make it impossible for malicious agencies to exploit the iteration-to-iteration dependency of the values to compromise data privacy.

In general we also observe that the data for agencies who have a larger number of variables is more secure since the components of $X_{I_j}\mathbf{s}^{(r)}_{I_j}$ involve summing over a larger number of $\mathbf{s}^{(r)}_{I_j}$.

5.2 What is Learned

After the regression coefficients are shared the agencies learn at least three useful quantities:

1. The **global coefficients**, $\hat{\beta}$. This enables the individual agencies to assess the effect of their variables on the response variable after accounting for the effects of the other agencies' variables. They can also assess the size of effects of the other agencies' variables. If an agency obtains a complete record for some individual, the global regression equation can also be used for prediction of the response value. A comparison of the globally obtained coefficients with the coefficients of the local regression (*i.e.*, the regression of $\mathbf{y}$ on X_{I_j}) could also be informative.

2. The vector of **residuals**, $\mathbf{e} = \mathbf{y} - X\hat{\beta}$ is also known (this is equal to final $\mathbf{z}$ in our iterative procedure). The residuals permit us to perform diagnostic tests

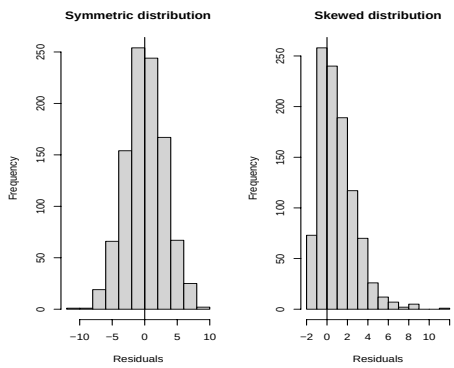

Figure 1: Distributions of the residuals

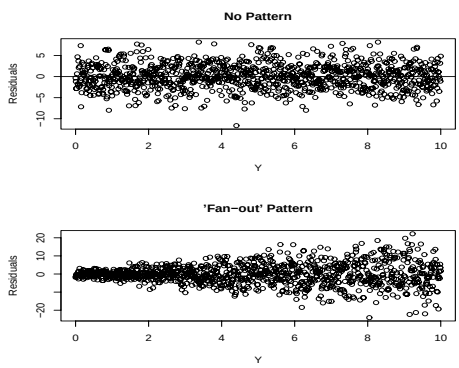

Figure 2: Scatterplots of the residuals

to determine if the linear regression model is appropriate (*i.e.*, if the model assumptions are satisfied). One could perform formal statistical tests. Two simple visual diagnostics are shown in Figures 1-2. If assumptions are satified, the distribution of the residuals ought to be symmetric. Figure 1 shows histograms of two sets of hypothetical residuals. The skewed distribution indicates a violation of the assumptions. The residuals can also be examined for systematic patterns. Residuals from a valid model should exhibit no pattern when plotted against, for example, $\mathbf{y}$. In Figure 2, the top plot shows residuals from a valid regression and the bottom one shows a distinct 'fan-out' pattern indicating a violation of the "equal-variance" assumption (other patterns could indicate other violations such as violation of a linear relationship assumption).

3. The **coefficient of determination**, R^2. Agencies can compute

$$R^2 = \frac{\mathbf{y}^T\mathbf{y} - \mathbf{e}^T\mathbf{e}}{\mathbf{y}^T\mathbf{y}}. \tag{9}$$

This $R^2 \in [0, 1]$ is a useful measure of the strength of the linear relationship assumed. Low values of R^2 indicate a weak linear relationship and high values indicate a good linear fit.

6. CONCLUDING REMARKS

We have presented a privacy preserving linear regession analysis algorithm that permits agencies to obtain the global regression equation as well as perform rudimentary goodness-of-fit diagnostics without revealing their data.

The local numerical computations to be conducted by each party are simple and efficient. The obvious dominant restricting factor in the computational effort required is conducting the $O(np^2)$ secure computations. We need to conduct experiments to evaluate the limits of this method both inters of the scale of the data as well as the number of agencies

There are some situations that the agencies need to be aware of.

- Our method critically relies on semi-honestness. If an agency is malicious and participates only to sabotage the collective efforts of the others, it can be quite successful by secretly not following protocol.
- The method is susceptible to "unfortunate" data. For instance, it might turn out that $R^2 \approx 1$ and all $\beta_j \approx 0$ for $j \neq 3$; then $\mathbf{x}_3$ is at risk.
- The ownership of certain attributes itself might be a sensitive issue. For instance, a agency that provides investment advice might possess health-related data on their clients that they would like to include in the regression, but would not like to reveal that to other agencies.

Outside the privacy scenario, our procedure also appears to be useful as a method for computing a common regression equation when the data reside in distributed databases. This application for distributed computation or as a strategy for a scalable method for linear regression is worth exploring further.

Our procedure also illustrates a more generally useful strategy for devising privacy-preserving data mining methods: Fitting of most data mining models involves the estimation of the parameters of the model by solving an optimization problem. Well-established models have well-established standard procedures for solving the optimization problem. Our approach indicates that exploring other non-standard methods for carrying out the optimization might lead to a method that is more suitable in the privacy preserving setting.

7. ACKNOWLEDGMENTS

This research was supported by NSF grant EIA–0131884 to the National Institute of Statistical Sciences. We wish to thank Max Buot, Christine Kohonen and four referees for useful discussion and comments.

8. REFERENCES

[1] ACM. *SIGKDD Explorations*, volume 4, December 2002.

[2] J. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret sharing. In A. M. Odlyzko, editor, *CRYPTO86*. Springer Verlag, 1987. Lecture Notes in Computer Science No. 263.

[3] R. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[4] C. Clifton, K. M., J. Vaidya, X. Lin, and M. Zhu. Tools for privacy preserving data mining. *ACM-SIGKDD Explorations*, 4(2):28–34, December 2002.

[5] W. Du and M. J. Atallah. Secure Multi-Party Computation Problems and Their Applications: A Review and Open Problems. In *New Security Paradigms Workshop*, pages 11–20, Cloudcroft, NM, September 2001.

[6] 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*, April 2004.

[7] D. E. Hershberger and H. Kargupta. Distributed Multivariate Regression Using Wavelet-Based Collective Data Mining. *Journal of Parallel and Distributed Computing*, 61(3):372–400, 2001.

[8] H. Kargupta and K. Liu. Distributed data mining bibliography.

[9] A. F. Karr, X. Lin, J. P. Reiter, and A. P. Sanil. Secure regression on distributed databases. *J. Computational and Graphical Statistics*, 2004. Submitted for publication. Available on-line at www.niss.org/dgii/technicalreports.html.

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

[11] W. H. Press, S. A. Teulosky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, second edition, 1992.

[12] J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *The Eighth ACM SIGKDD Iternational Conference on Knowledge Discovery and Data Mining*, pages 639–644, July 2002.

[13] J. Vaidya and C. Clifton. Privacy preserving k-means clustering over vertically partitioned data. In *The Ninth ACM SIGKDD Iternational Conference on Knowledge Discovery and Data Mining*, August 2003.

[14] S. Weisberg. *Applied Linear Regression*. Wiley, 1985.

[15] Y. Xing, M. G. Madden, J. Duggan, and G. J. Lyons. Distributed Regression for Heterogeneous Data Sets. In M. R. Berthold, H.-J. Lenz, E. Bradley, R. Kruse, and C. Borgelt, editors, *Proceedings of 5th International Symposium on Intelligent Data Analysis (IDA2003)*, LNCS 2810, pages 544–553, Berlin, German, August 2003. Springer.