Privacy-Preserving Horizontally Partitioned Linear Programs

Olvi L. Mangasarian*

Abstract

We propose a simple privacy-preserving reformulation of a linear program whose equality constraint matrix is partitioned into groups of rows. Each group of matrix rows and its corresponding right hand side vector are owned by a distinct private entity that is unwilling to share or make public its row group or right hand side vector. By multiplying each privately held constraint group by an appropriately generated and privately held random matrix, the original linear program is transformed into an equivalent one that does not reveal any of the privately held data or make it public. The solution vector of the transformed secure linear program is publicly generated and is available to all entities.

Keywords: security, privacy-preserving, linear programming, horizontally partitioned data

1 INTRODUCTION

Recent interest in privacy-preserving classification and data mining [15, 13, 14, 6, 2, 10, 8, 9, 12], wherein the data to be classified or mined is owned by different entities that are unwilling to reveal the data they hold or make it public, has spread to the field of optimization and in particular linear programming [3, 12, 1, 7]. In [1] a number of shortcomings in the privacy-preserving linear programming literature are pointed out. In [7] a method for handling privately held vertical partitions of a linear programming constraint matrix and cost vector is proposed that is based on private random transformations of the corresponding problem variables. In this work we deal with a *horizontal* partition of the equality constraint matrix of a linear program and the corresponding right hand side vector into groups. Each group is owned by a distinct entity wishing to keep its data private, but at the same time wanting a solution to the overall problem based on all the constraints. We address this problem by multiplying each privately held equality constraint by an appropriate and privately generated and held random matrix. This process transforms the original linear program into an equivalent secure linear program that does not reveal any of the privately held data or make it public. The solution vector is then generated publicly and is made available to all entities.

We briefly describe the contents of the paper. In Section 2 we give the theory and in Section 3 an implementation of our method for a privacy-preserving linear programming formulation using a random transformation of the problem constraints. In Section 4 we give numerical examples demonstrating our approach. Section 5 concludes the paper with an open problem.

We describe our notation now. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector $x \in R^n$ the notation x_j will signify either the j-th component or j-th block of components. The scalar (inner) product of two vectors x and y in the n-dimensional real space R^n will be denoted by $x'y$. The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix, A' will denote the transpose of A, A_i will denote the i-th row or i-th block of rows of A and $A_{\cdot j}$ the j-th column or the j-th block of columns of A. A vector of zeros in a real space of arbitrary dimension will be denoted by 0.

*Computer Sciences Department, University of Wisconsin, Madison, WI 53706 and Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.*olvi@cs.wisc.edu*.

2 Privacy-Preserving Linear Programming for Horizontally Partitioned Data

We consider the linear program:

$$\min_{x\in X} c'x \text{ where } X=\{x \mid Ax=b,\ x\geq 0\}. \tag{2.1}$$

Here, the matrix $A\in R^{m\times n}$ together with the right hand side vector $b\in R^m$, that is $[A \quad b]$, are divided into p horizontal blocks of $m_1, m_2, \ldots\ldots$ and m_p, $(n+1)$-dimensional rows with $m_1+m_2+\ldots+m_p=m$. Each block of rows of A and corresponding block of the right hand side vector b are "owned" by a distinct entity that is unwilling to make this block of data public or share it with the other entities. We wish to solve this linear program without revealing any privately held data. We shall achieve this by proceeding as follows.

Each of the p entities chooses its own privately held random matrix $B_{\cdot i}\in R^{k\times m_i}$, $i=1,\ldots,p$, where $k\geq m$. Define:

$$B=[B_{\cdot 1}\ \ B_{\cdot 2}\ldots\ldots B_{\cdot p}]\in R^{k\times m}. \tag{2.2}$$

We note immediately that the rank of the randomly generated matrix $B\in R^{k\times m}$ with $k\geq m$ is m [4], which is the reason for choosing $k\geq m$. Utilizing this fact we define the following transformations:

$$BA=[B_{\cdot 1}\ B_{\cdot 2}\ \ldots B_{\cdot p}]\begin{bmatrix}A_1\\A_2\\ \vdots\\ A_p\end{bmatrix}=B_{\cdot 1}A_1+B_{\cdot 2}A_2+\ldots+B_{\cdot p}A_p\in R^{k\times n}, \tag{2.3}$$

and

$$Bb=[B_{\cdot 1}\ B_{\cdot 2}\ \ldots B_{\cdot p}]\begin{bmatrix}b_1\\b_2\\ \vdots\\ b_p\end{bmatrix}=B_{\cdot 1}b_1+B_{\cdot 2}b_2+\ldots+B_{\cdot p}b_p\in R^{k}. \tag{2.4}$$

With these transformations our original linear program (2.1) turns into the following "secure" linear program:

$$\min_{y\in Y} c'y \text{ where } Y=\{y \mid BAy=Bb,\ y\geq 0\}. \tag{2.5}$$

We use the term "secure" to describe the transformed linear program (2.5) because it does not reveal any of the privately held data $\begin{bmatrix}A_1 & b_1\\A_2 & b_2\\ & \vdots & \\ A_p & b_p\end{bmatrix}$, $i=1,\ldots,p$. This is so because for each entity different from entity i, it is impossible to compute either A_i from the revealed product $B_{\cdot i}A_i$, or b_i from the revealed product $B_{\cdot i}b_i$ *without* knowing the random matrix $B_{\cdot i}$ chosen by entity i and known to itself only. See also Remark 3.2 below.

We now relate our original linear program (2.1) to the transformed linear program (2.5) as follows.

PROPOSITION 2.1. *Let $k\geq m$ for the random matrix $B\in R^{k\times m}$ of (2.2). The secure linear program (2.5) is solvable if and only if the linear program (2.1) is solvable in which case the solution sets of the two linear programs are identical.*

Proof Beause the rank of the random matrix B of (2.2) is m the following equivalence is obvious:

$$Ax=b \iff BAx=Bb. \tag{2.6}$$

Consequently the feasible region X of the original linear program (2.1) is equivalent to the feasible region Y of the secure linear program (2.5). Since both objective functions are the same, it follows immediately that both problems have the same solution set. $\square$

We turn now to an explicit implementation of the secure linear programming formulation (2.5).

3 Privacy-Preserving for Horizontally Partitioned Linear Program (PPHPLP)

Starting with the linear program (2.1) that is partitioned among p entities as described in Section 2, we propose the following algorithm for generating a solution to the linear program without disclosing any of the privately held data.

ALGORITHM 3.1. **PPHPLP Algorithm**

(I) All p entities agree on a value for $k \geq m$, where k is the number of rows of the random matrix $B \in R^{k\times m}$ as defined in (2.2).

(II) Each entity generates its own privately held random matrix $B_{\cdot i} \in R^{k\times m_i}$, $i = 1,\ldots\ldots,p$, where m_i is the number of rows held by entity i which results in:

$$B = [B_{\cdot 1}\ \ B_{\cdot 2}\ldots\ldots B_{\cdot p}] \in R^{k\times m}. \tag{3.7}$$

(III) Each entity i makes public only its matrix product $B_{\cdot i}A_i$ as well as its right hand side product $B_{\cdot i}b_i$. These products do not reveal either A_i or b_i but allow the public computation of the full constraint matrix needed for the secure linear program (2.5):

$$BA = [B_{\cdot 1}A_1 + B_{\cdot 2}A_2 + \ldots + B_{\cdot p}A_p] \in R^{k\times n}, \tag{3.8}$$

as well as the right hand side for (2.5):

$$Bb = [B_{\cdot 1}b_1 + B_{\cdot 2}b_2 + \ldots + B_{\cdot p}b_p] \in R^{k}. \tag{3.9}$$

(IV) A public optimal solution vector y to the secure linear program (2.5) is obtained which, by Proposition 2.1, also solves the original linear program (2.1).

REMARK 3.2. *Note that in the above algorithm no entity i reveals its data matrix A_i or its right hand side vector b_i. However, the solution vector y obtained is publicly available. Note further that it is impossible to compute the m_in numbers constituting $A_i \in R^{m_i\times n}$, given only the kn numbers constituting the revealed matrix product $B_{\cdot i}A_i \in R^{k\times n}$, and not knowing $B_{\cdot i} \in R^{k\times m_i}$. Similarly it is impossible to compute $b_i \in R^{m_i}$ from $B_{\cdot i}b_i \in R^k$. Hence, all entities share the publicly computed optimal solution, but without revealing their privately held data.*

We turn now to some computational results.

4 Computational Results

We demonstrate our results by solving two examples as follows on a 4 Gigabyte machine running $i386\_rhe15$ Linux. We utilize the CPLEX linear programming code [5] within MATLAB [11] to solve our linear programs. The first example has 600 constraints and 1000 variables, while the second example has 1000 variables and 1000 constraints. For both examples, the components of the matrix A were uniformly distributed in the interval $[-50,50]$, and approximately half of the components of the primal solution x of (2.1) were uniformly distributed in the interval $[0,10]$ while the other half was zero. Similarly, approximately half of the constraints of the dual problem were active, that is satisfied as equalities, while the other half were inactive. We used $k = 1000$ in both examples.

EXAMPLE 4.1. *For our first example we generated a random solvable linear program (2.1) with $m = 600$ and $n = 1000$. We partitioned the rows of A as well as the right hand side vector b into three groups with $m_1 = 100$, $m_2 = 200$ and $m_3 = 300$. We generated three random matrices, with coefficients uniformly distributed in the interval [0,1] with $B_1 \in R^{k\times m_1}$, $B_2 \in R^{k\times m_2}$ and $B_3 \in R^{k\times m_3}$. We solved the secure linear program (2.5) and compared its optimal solution with that of (2.1). The two optimal solutions were identical. Computation time was 11.817 seconds for the secure linear program (2.5).*

EXAMPLE 4.2. *For our second example we generated a random solvable linear program (2.1) with* $m = 1000$ *and* $n = 1000$ *with approximately half of the primal constraints being redundant. We partitioned the rows of A as well as the right hand side vector b into three groups with* $m_1 = 200$*,* $m_2 = 300$ *and* $m_3 = 500$*. We generated three random matrices, with coefficients uniformly distributed in the interval [0,1] with* $B_1 \in R^{k\times m_1}$*,* $B_2 \in R^{k\times m_2}$ *and* $B_3 \in R^{k\times m_3}$*. We solved the secure linear program (2.5) and compared its optimal solution with that of (2.1). The two optimal solutions were identical. Computation time was 14.184 seconds (2.5).*

5 Conclusion and Outlook

We have shown how to securely solve a linear program when its equality constraint matrix and its right hand side data are partitioned among entities unwilling to share their data or make it public. Another interesting problem in this realm occurs when the equality constraints of the linear program (2.1) are inequality constraints instead. The approach proposed here does not work because we cannot multiply these inequality constraints by a random matrix $B \in R^{k\times m}$, even if $B \geq 0$, and preserve the original feasible region of the problem. We leave this as an open problem for future research.

Acknowledgments The research described in this Data Mining Institute Report 10-02, April 2010, was supported by the Microsoft Corporation and ExxonMobil.

References

[1] A. Bednarz, N. Bean, and M. Roughan. Hiccups on the road to privacy-preserving linear programming. In *Proceedings of the 8th ACM Workshopon on Privacy in the Electronic Society*, pages 117–120, 2009.

[2] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Proceedings of the Fifth International Conference of Data Mining (ICDM'05)*, pages 589–592. IEEE, 2005.

[3] W. Du. A study of several specific secure two-party computationproblems. Technical report, Purdue University, West Lafayette, IN, 2001. PhD Dissertation.

[4] X. Feng and Z. Zhang. The rank of a random matrix. *Applied Mathematics and Computation*, 185:689–694, 2007.

[5] ILOG, Incline Village, Nevada. *ILOG CPLEX 9.0 User's Manual*, 2003. http://www.ilog.com/products/cplex/.

[6] Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. Cryptographically private support vector machines. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–624, New York, NY, USA, 2006. ACM.

[7] O. L. Mangasarian. Privacy-preserving linear programming. Technical Report 10-01, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, March 2010. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/10-01.pdf.

[8] O. L. Mangasarian and E. W. Wild. Privacy-preserving classification of horizontally partitioned data via random kernels. Technical Report 07-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, November 2007. Proceedings of the 2008 International Conference on Data Mining, DMIN08, Las Vegas July 2008, Volume II, 473-479, R. Stahlbock, S.V. Crone and S. Lessman, Editors.

[9] O. L. Mangasarian and E. W. Wild. Privacy-preserving random kernel classification of checkerboard partitioned data. Technical Report 08-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 2008. Annals of Information Systems XIII, 2010, 375-387.

[10] O. L. Mangasarian, E. W. Wild, and G. M. Fung. Privacy-preserving classification of vertically partitioned data via random kernels. Technical Report 07-02, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, September 2007. ACM Transactions on Knowledge Discovery from Data (TKDD) Volume 2, Issue 3, October 2008.

[11] MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2006. http://www.mathworks.com.

[12] J. Vaidya, C. Clifton, and M. Zu. *Privacy Preserving Data Mining*. Springer Science+Business Media Inc., New York, 2009.

[13] M.-J. Xiao, L.-S. Huang, H. Shen, and Y.-L. Luo. Privacy preserving id3 algorithm over horizontally partitioned data. In *Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, pages 239–243. IEEE Computer Society, 2005.

[14] H. Yu, X. Jiang, and J. Vaidya. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 603–610, New York, NY, USA, 2006. ACM Press.

[15] H. Yu, J. Vaidya, and X. Jiang. Privacy-preserving svm classification on vertically partitioned data. In *Proceedings of PAKDD '06*, volume 3918 of *LNCS: Lecture Notes in Computer Science*, pages 647 – 656. Springer-Verlag, January 2006.