Hindawi
Journal of Mathematics
Volume 2021, Article ID 6651480, 4 pages
https://doi.org/10.1155/2021/6651480

Research Article

A New Algorithm for Privacy-Preserving Horizontally Partitioned Linear Programs

Chengxue Zhang, Debin Kong, Peng Pan, and Mingyuan Zhou

Yantai Nanshan University, Yantai 265713, China

Correspondence should be addressed to Debin Kong; kongdebin@nanshan.edu.cn

Received 13 December 2020; Revised 13 January 2021; Accepted 29 January 2021; Published 11 February 2021

Academic Editor: Sun Young Cho

In a linear programming for horizontally partitioned data, the equality constraint matrix is divided into groups of rows. Each group of the matrix rows and the corresponding right-hand side vector are owned by different entities, and these entities are reluctant to disclose their own groups of rows or right-hand side vectors. To calculate the optimal solution for the linear programming in this case, Mangasarian used a random matrix of full rank with probability 1, but an event with probability 1 is not a certain event, so a random matrix of full rank with probability 1 does not certainly happen. In this way, the solution of the original linear programming is not equal to the solution of the secure linear programming. We used an invertible random matrix for this shortcoming. The invertible random matrix converted the original linear programming problem to a secure linear program problem. This secure linear programming will not reveal any of the privately held data.

1. Introduction

Recently, people have become interested in privacy-preserving classification and data mining [1–10] and have been involved in the field of optimization, especially in linear programming [11–15], where the data to be classified or mined belongs to different entities that are not willing to disclose the data. Mangasarian [13] proposed a random matrix which make the original linear programming problem into a secure linear programming problem. When the random matrix is not full rank [16], especially when the entities collide with each other, the original linear programming problem is not equivalent to the secure linear programming problem. We address this problem by using an invertible matrix multiplied by the two sides of the equality constraints of the linear program. This procedure converts the original linear program to an equivalent secure linear program, and this security linearity does not reveal any private data. This solution vector can be made public and applied by all entities. On the contrary, this algorithm prevents entities from colliding with each other.

Here, we define some symbols. If a vector is not transposed to the row vector by the superscript T, the vector will be a column vector. For a vector $x \in R^n$, the symbol x_j will represent the j^{th} component or j^{th} block of the component. We will define the scalar (inner) product of two vectors x and y in the n-dimensional real space R^n as $x^T y$. The symbol $A \in R^{m\times n}$ will represent a real $m \times n$ matrix. Similarly, A^T will represent the transpose of A and $A_{i.}$ will represent the i row or i block of rows of A and $A_{.j}$ the j^{th} column or the j^{th} block of columns of A. A zero vector in a real space of any dimension will be denoted by 0.

2. Privacy-Preserving Linear Programming for Horizontally Partitioned Data

Consider the following linear programming:

$$\begin{aligned} &\min z = c^T x, \\ &\text{s.t.} \quad Ax = b \\ &\qquad x \geq 0. \end{aligned} \tag{1}$$

Here, $(A\ \ b)$ consists of the matrix $A \in R^{m\times n}$ and the right-hand vector $b \in R^m$ and is divided into p horizontal blocks. The number of rows of the p horizontal block is recorded as $m_1, m_2, \ldots, m_p$, where $m_1 + m_2 + \cdots + m_p = m$. An m order identity matrix E is divided into $p$vertical blocks. The number of columns of the p vertical block is recorded as $m_1, m_2, \ldots, m_p$, where $m_1 + m_2 + \cdots + m_p = m$. Each block of rows of $[A\ \ b]$ corresponding to the index sets $I_1, I_2, \ldots, I_p$, $\cup_{i=1}^{p} I_i = \{1, 2, \ldots, m\}$, is owned by a distinct entity that is unwilling to make its block of data public or share it with the other entities. We will accomplish this goal by the following transformation.

Each entity $i, i = 1, 2, \ldots, p$, chooses its own private random matrix $B_{.I_i} \in R^{m\times m_i}$, whose corresponding index set is I_i. The value of each element in $B_{.I_i}$ is in the interval $(0, 1)$. The following decompositions can be obtained:

$$A = \begin{pmatrix} A_{I_1.} \\ A_{I_2.} \\ \vdots \\ A_{I_p.} \end{pmatrix} \text{ and } b = \begin{pmatrix} b_{I_1} \\ b_{I_2} \\ \vdots \\ b_{I_p} \end{pmatrix}.$$

Define

$$B = \begin{pmatrix} B_{.I_1} + \lambda E_{.I_1} & B_{.I_2} + \lambda E_{.I_2} & \cdots & B_{.I_p} + \lambda E_{.I_p} \end{pmatrix}, \quad \lambda \in R,\ \lambda \geq n. \tag{2}$$

Because the matrix B is an m order strictly diagonally dominant matrix, we can easily conclude that the matrix B is an invertible matrix [17]. Based on this fact, we define the following operation:

$$\begin{aligned} BA &= \begin{pmatrix} B_{.I_1} + \lambda E_{.I_1} & B_{.I_2} + \lambda E_{.I_2} & \cdots & B_{.I_p} + \lambda E_{.I_p} \end{pmatrix} \begin{pmatrix} A_{I_1.} \\ A_{I_2.} \\ \vdots \\ A_{I_p.} \end{pmatrix} \\ &= \left(B_{.I_1} + \lambda E_{.I_1}\right)A_{I_1.} + \left(B_{.I_2} + \lambda E_{.I_2}\right)A_{I_2.} + \cdots + \left(B_{.I_p} + \lambda E_{.I_p}\right)A_{I_p.}, \\ Bb &= \begin{pmatrix} B_{.I_1} + \lambda E_{.I_1} & B_{.I_2} + \lambda E_{.I_2} & \cdots & B_{.I_p} + \lambda E_{.I_p} \end{pmatrix} \begin{pmatrix} b_{I_1} \\ b_{I_2} \\ \vdots \\ b_{I_p} \end{pmatrix} \\ &= \left(B_{.I_1} + \lambda E_{.I_1}\right)b_{I_1} + \left(B_{.I_2} + \lambda E_{.I_2}\right)b_{I_2} + \cdots + \left(B_{.I_p} + \lambda E_{.I_p}\right)b_{I_p}. \end{aligned} \tag{3}$$

According to the above discussion, the original linear programming (1) was converted into the following secure linear programming:

$$\begin{aligned} \min\ & z = c^T x, \\ \text{s.t.},\ & BAx = Bb \\ & x \geq 0. \end{aligned} \tag{4}$$

The linear programming (1) and the linear programming (4) have the same solution set since the matrix B is invertible. The linear programming (4) is quite safe since only the entity i knows $B_{.I_i}$, $i = 1, 2, \ldots, p$. Other entities cannot compute $A_{I_i.}$ and b_{I_i} from $(B_{.I_i} + \lambda E_{.I_i})A_{I_i.}$ and $(B_{.I_i} + \lambda E_{.I_i})b_{I_i}$ without knowing the random matrix $B_{.I_i}$. We regard the linear programming (4) as a secure linear programming. Whether the linear programming (1) is equivalent to the linear programming (4) or not? Let us discuss next.

Proposition 1. *If the matrix B is an m order invertible matrix; then, the secure linear program (4) is solvable if and only if the linear program (1) is solvable in case the solution sets of the two linear programs are identical.*

Proof. As the matrix B is an m-order invertible matrix, the following relation holds:

$$Ax = b \Leftrightarrow BAx = Bb. \tag{5}$$

Therefore, the feasible regions of the two linear programs are the same. Again according to the objective functions of the linear programming (1) and the linear programming (4), we can conclude that the two linear programs have the same solution set.

The following algorithm can get the best solution of the linear programming (1) without revealing any private data. □

3. Formulation of the Privacy-Preserving Algorithm

As shown in Section 2, the linear program (1) is divided among p entities. We put forward the following algorithm:

Step 1. All entities choose a suitable real number $\lambda, \lambda \geq n$ together.

Step 2. Suppose the matrix $\begin{pmatrix} A_{I_i.} & b_{I_i} \end{pmatrix}$ has m_i rows, where $i = 1, 2, \ldots, p$. A random matrix $B_{.I_i}$ is generated by the entity possessing the matrix $\begin{pmatrix} A_{I_i.} & b_{I_i} \end{pmatrix}$, where $B_{.I_i} \in R^{m\times m_i}$. The value of each element in $B_{.I_i}$ is in the interval (0,1), and $B_{.I_i}$ is not public.

Step 3. The entity that owns the matrix $\begin{pmatrix} A_{I_1.} & b_{I_1} \end{pmatrix}$ is responsible to compute $(B_{.I_1} + \lambda E_{.I_1})A_{I_1.}$ and $(B_{.I_1} + \lambda E_{.I_1})b_{I_1}$, and the result is passed to the entity that owns the matrix $\begin{pmatrix} A_{I_2.} & b_{I_2} \end{pmatrix}$. Then, the entity that owns the matrix $\begin{pmatrix} A_{I_2.} & b_{I_2} \end{pmatrix}$ is responsible to compute $(B_{.I_1} + \lambda E_{.I_1})A_{I_1.} + (B_{.I_2} + \lambda E_{.I_2})A_{I_2.}$ and $(B_{.I_1} + \lambda E_{.I_1})b_{I_1} + (B_{.I_2} + \lambda E_{.I_2})b_{I_2}$, and the result is passed to the entity that owns the matrix $\begin{pmatrix} A_{I_3.} & b_{I_3} \end{pmatrix}$. And, finally, the entity that owns the matrix $\begin{pmatrix} A_{I_p.} & b_{I_p} \end{pmatrix}$ is responsible to compute the following:

$$\begin{aligned} BA = &\left(B_{.I_1} + \lambda E_{.I_1}\right)A_{I_1.} + \left(B_{.I_2} + \lambda E_{.I_2}\right)A_{I_2.} \\ &+ \cdots + \left(B_{.I_p} + \lambda E_{.I_p}\right)A_{I_p.}, \end{aligned} \tag{6}$$

$$\begin{aligned} Bb = &\left(B_{.I_1} + \lambda E_{.I_1}\right)b_{I_1} + \left(B_{.I_2} + \lambda E_{.I_2}\right)b_{I_2} \\ &+ \cdots + \left(B_{.I_p} + \lambda E_{.I_p}\right)b_{I_p}. \end{aligned} \tag{7}$$

Step 4. Utilizing the linear programming (4) to calculate the minimum value and the optimal solution of the

objective function, which is the minimum value and the optimal solution of the objective function of the linear programming (1).

Remark 1. Through this algorithm, the solution vector x can be used publicly. However, it does not reveal any entity's data.

4. Numerical Experiments

A linear programming:

$$\begin{aligned} \min\ & z = -3x_1 - 5x_2, \\ & x_1 + x_3 = 8, \\ & 2x_2 + x_4 = 12, \\ \text{s.t.}\ & 3x_1 + 4x_2 + x_5 = 36, \\ & x_i \geq 0, \quad i = 1, \ldots, 5. \end{aligned} \tag{8}$$

We can find that the optimal solution of (8) is $x^* = (4, 6, 4, 0, 0)^T$.Let $I_1 = \{1, 2\}, I_2 = \{3\}, A_{I_1.} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \end{pmatrix}$, $A_{I_2.} = \begin{pmatrix} 3 & 4 & 0 & 0 & 1 \end{pmatrix}$, $b_{I_1} = \begin{pmatrix} 8 \\ 12 \end{pmatrix}, b_{I_2} = (36)$, and $\lambda = 5$.

Entity 1 generates a random matrix $B_{.I_1}$ which is not published. Note that

$$B_{.I_1} = \begin{pmatrix} 0.9501 & 0.7621 \\ 0.2311 & 0.7621 \\ 0.6068 & 0.7621 \end{pmatrix}. \tag{9}$$

Entity 1 makes public its matrix product $(B_{.I_1} + \lambda E_{.I_1})A_{I_1.}$ and $(B_{.I_1} + \lambda E_{.I_1})b_{I_1}$.

Entity 2 generates a random matrix $B_{.I_2}$ which is not published. Note that

$$B_{.I_2} = \begin{pmatrix} 0.7621 \\ 0.7621 \\ 0.7621 \end{pmatrix}. \tag{10}$$

Entity 2 makes public its matrix product $(B_{.I_2} + \lambda E_{.I_2})A_{I_2.}$ and $(B_{.I_2} + \lambda E_{.I_2})b_{I_2}$.

These products do not reveal any private data, but it can be used to calculate the constraint matrix BA and the right-hand side Bb of the secure linear programming. Next, we derive a linear programming (11) from the linear programming (4), which is equivalent to linear programming (8):

$$\begin{aligned} \min\ & z = -3x_1 - 5x_2, \\ & 8.2364x_1 + 4.5726x_2 + 5.9501x_3 + 0.7621x_4 + 0.7621x_5 = 84.1816, \\ \text{s.t.}\ & 2.5174x_1 + 14.5726x_2 + 0.2311x_3 + 5.7621x_4 + 0.7621x_5 = 98.4296, \\ & 17.8931x_1 + 24.5726x_2 + 0.6068x_3 + 0.7621x_4 + 5.7621x_5 = 221.4352, \\ & x_i \geq 0 \quad i = 1, \ldots, 5. \end{aligned} \tag{11}$$

The solution of this secure linear programming (11) is the same as that of the linear programming (8). This solution can be made public without revealing any private data.

If we use Mangasarian's study [13] which proposed the algorithm of privacy-preserving horizontally partitioned linear programs, the linear programming (8) needs to be converted into the following linear programming:

$$\begin{aligned} \min\ & z = -3x_1 - 5x_2, \\ & 3.2364x_1 + 4.5726x_2 + 0.9501x_3 + 0.7621x_4 + 0.7621x_5 = 44.1816, \\ \text{s.t.}\ & 2.5174x_1 + 4.5726x_2 + 0.2311x_3 + 0.7621x_4 + 0.7621x_5 = 38.4296, \\ & 2.8931x_1 + 4.5726x_2 + 0.6068x_3 + 0.7621x_4 + 0.7621x_5 = 41.4352, \\ & x_i \geq 0 \quad i = 1, \ldots, 5. \end{aligned} \tag{12}$$

The optimal solution to secure linear program (12) is $x^{*\prime} = (8, 4, 0, 0, 0)^T$. This is not consistent with the optimal solution for the original linear programming (8). The reason for this error is that the random matrix $\begin{pmatrix} B_{.I_1} & B_{.I_2} \end{pmatrix} = \begin{pmatrix} 0.9501 & 0.7621 & 0.7621 \\ 0.2311 & 0.7621 & 0.7621 \\ 0.6068 & 0.7621 & 0.7621 \end{pmatrix}$ is not a full rank matrix. In this way, the original linear programming (8) is not equivalent to the secure linear program (12).

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Aluminum-Copper Strip Material Intelligent Process Control Technology Project Based on Industrial Big Data (Grant no. 2017YFB0306404).

References

[1] L. Sun, W.-S. Mu, B. Qi, and Z.-J. Zhou, "A new privacy-preserving proximal support vector machine for classification of vertically partitioned data," *International Journal of Machine Learning and Cybernetics*, vol. 6, no. 1, p. 109, 2014.

[2] M. A. Xindi, L. I. Hui, M. A. Jianfeng et al., "APPLET: a privacy-preserving framework for location-aware recommender system," *Science China*, vol. 60, no. 9, pp. 5–20, 2017.

[3] D. R. Sahu, J.-C. Yao, M. Verma, and K. K. Shukla, "Convergence rate analysis of proximal gradient methods with applications to composite minimization problems," *Optimization*, vol. 70, no. 1, pp. 1–26, 2020.

[4] E. Luo, Q. Liu, J. H. Abawajy, and G. Wang, "Privacy-preserving multi-hop profile-matching protocol for proximity mobile social networks," *Future Generation Computer Systems*, vol. 68, p. 222, 2017.

[5] S. Y. Cho, "Implicit extragradient-like method for fixed point problems and variational inclusion problems in a Banach space," *Symmetry*, vol. 12, no. 6, p. 998, 2020.

[6] M.-J. Xiao, L.-S. Huang, Y.-L. Luo, and H. Shen, "Privacy preserving id3 algorithm over horizontally partitioned data," in *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05)*, pp. 239–243, Dalian, China, December 2005.

[7] N. T. An, N. M. Nam, and X. Qin, "Solving k-center problems involving sets based on optimization techniques," *Journal of Global Optimization*, vol. 76, no. 1, pp. 189–209, 2020.

[8] S. Y. Cho, "A monotone Bregan projection algorithm for fixed point and equilibrium problems in a reflexive Banach space," *Filomat*, vol. 34, no. 5, pp. 1487–1497, 2020.

[9] K. Chen and L. Liu, "Privacy preserving data classification with rotation perturbation," in *Proceedings of the Fifth International Conference of Data Mining (ICDM'05)*, pp. 589–592, Houston, TX, USA, November 2005.

[10] O. L. Mangasarian, E. W. Wild, and G. M. Fung, "Privacy-preserving classification of vertically partitioned data via random kernels," *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 3, pp. 1–16, 2008.

[11] L. V. Nguyen and X. Qin, "The minimal time function associated with a collection of sets," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 26, p. 93, 2020.

[12] O. L. Mangasarian, "Privacy-preserving linear programming," *Optimization Letters*, vol. 5, no. 1, pp. 165–172, 2011.

[13] O. L. Mangasarian, "Privacy-preserving horizontally partitioned linear programs," *Optimization Letters*, vol. 6, no. 3, pp. 431–436, 2012.

[14] S. Y. Cho, "A convergence theorem for generalized mixed equilibrium problems and multivalued asymptotically nonexpansive mappings," *Journal of Nonlinear and Convex Analysis*, vol. 21, pp. 1017–1026, 2020.

[15] W. Li, H. Li, and C. Deng, "Privacy-preserving horizontally partitioned linear programs with inequality constraints," *Optimization Letters*, vol. 7, no. 1, pp. 137–144, 2013.

[16] X. Feng and Z. Zhang, "The rank of a random matrix," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 689–694, 2007.

[17] Q. Y. N. Chao-Wang and D. Yi-Yi, *Numericalanalysis*, pp. 210-211, Huazhong University of Science and Technology Press, Wuhan, China, 2006.