Optim Lett (2013) 7:137–144
DOI 10.1007/s11590-011-0403-2

ORIGINAL PAPER

Privacy-preserving horizontally partitioned linear programs with inequality constraints

Wei Li · Haohao Li · Chongyang Deng

Received: 15 August 2011 / Accepted: 20 September 2011 / Published online: 4 October 2011

Abstract In this paper we solve the open problem, finding the solutions for privacy-preserving horizontally partitioned linear programs with inequality constraints, proposed recently by Mangasarian (Optim Lett 2011, doi:10.1007/s11590-010-0268-9).

Keywords Linear programming · Privacy-preserving · Horizontally partitioned data

1 Introduction

Consider jointly-defined linear programs where two or more parties each contribute various components to form a whole optimization problem. The critical information to be kept hidden is dependent on the situation but could include the constraint set (which may contain private information about budgets financial health, production capacity, or network layout etc.) and the objective function (which may express costs or company aims). The goal is to solve the joint-program and yet keep the private information hidden as much as possible. This type of problem is referred to as "privacy-preserving linear program". Recently there has been substantial interest in privacy-preserving linear programming problems [1–6].

An approach to privacy-preserving linear programming has been developed is transformation-based method [1,4]. Bednarz et al. [6] pointed out some flaws of the these transformations.

W. Li (✉) · C. Deng
School of Science, Hangzhou Dianzi University, Hangzhou 310018, People's Republic of China
e-mail: weili@hdu.edu.cn

H. Li
Department of Mathematics, State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, People's Republic of China

Recently, Mangasarian [7,8] proposed some effective solution methods for privacy-preserving linear programming, based on random matrix transformation. Consider the linear program

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

Here, the matrix $A \in R^{m \times n}$ together with the right hand side vector $b \in R^m$, that is $[A\ b]$, are divided into p horizontal blocks of $m_1, m_2, \ldots$ and m_p, $(n+1)$-dimensional rows with $m_1 + m_2 + \cdots + m_p = m$. Each block of rows of $[A\ b]$ corresponding to the index sets $I_1, I_2, \ldots, I_p$, $\bigcup_{i=1}^{p} I_i = \{1, 2, \ldots, m\}$, is owned by a distinct entity that is unwilling to make its block of data public or share it with the other entities. In [8], a method for solving this linear program without revealing any privately held data is proposed. Also, in [8] the following open problem is posed:

"Another interesting problem in this realm occurs when the equality constraints of the linear program (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. Furthermore, if we convert the inequality constraints to equality constraints by adding slack variables, multiplying the ith identity matrix coefficient matrix of the slack variables of the ith entity by its privately held random matrix $B_{\cdot i}$ would reveal $B_{\cdot i}$. Hence, treating inequality constraints remains an open problem for future research."

This paper proposes a solution to this open problem. In the next section we give the theory behind our approach. Then in Sect. 3, we describe the algorithm based on the theory proposed in Sect. 2. An illustrative example of the proposed algorithm is given in Sect. 4 to demonstrate our approach.

We describe our notation now. All vectors will be column vectors unless transposed to a row vector by a superscript T. For a vector $x \in R^n$ the notation x_j will signify either the jth component or jth 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 $x^T y$. The notation $A \in R^{m \times n}$ will signify a real $m \times n$ matrix. For such a matrix, A^T will denote the transpose of A, A_i will denote the ith row or ith block of rows of A and $A_{\cdot,j}$ the jth column or the jth block of columns of A. A vector of zeros in a real space of arbitrary dimension will be denoted by 0.

2 Privacy-preserving linear programming for horizontally partitioned data with inequality constraints

Consider the linear program

$$\begin{aligned} &min\ \ c^T x \\ &s.t.\ Ax \leq b, \quad x \geq 0. \end{aligned} \tag{2}$$

Here, the matrix $A \in R^{m\times n}$ together with the right hand side vector $b \in R^m$, that is $[A\ b]$, are divided into p horizontal blocks of $m_1, m_2, \ldots$ and m_p, $(n+1)$-dimensional rows with $m_1 + m_2 + \cdots + m_p = m$. Each block of rows of $[A\ b]$ corresponding to the index sets $I_1, I_2, \ldots, I_p$, $\bigcup_{i=1}^{p} I_i = \{1, 2, \ldots, m\}$, 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 solve this linear program without revealing any privately held data. We shall achieve this by a transformation approach.

Each of entity $i, i = 1, \ldots, p$, chooses its own privately held random matrix $B_{\cdot I_i} \in R^{k\times m_i}$ with $k \geq m$, corresponding to the index set I_i. We thus have the following decompositions

$$A = \begin{bmatrix} A_{I_1\cdot} \\ A_{I_2\cdot} \\ \vdots \\ A_{I_p\cdot} \end{bmatrix}, b = \begin{bmatrix} b_{I_1} \\ b_{I_2} \\ \vdots \\ b_{I_p} \end{bmatrix}$$

and

$$B = [B_{\cdot I_1}, B_{\cdot I_2}, \ldots, B_{\cdot I_p}] \in R^{k\times m}. \tag{3}$$

The rank of the randomly generated matrix $B \in R^{k\times m}$ with $k \geq m$ is m [9]. Thus we have

$$\begin{aligned} BA &= [B_{\cdot I_1}, B_{\cdot I_2}, \ldots, B_{\cdot I_p}] \begin{bmatrix} A_{I_1\cdot} \\ A_{I_2\cdot} \\ \vdots \\ A_{I_p\cdot} \end{bmatrix} \\ &= B_{\cdot I_1}A_{I_1\cdot} + B_{\cdot I_2}A_{I_2\cdot} + \cdots + B_{\cdot I_p}A_{I_p\cdot} \in R^{k\times n} \end{aligned} \tag{4}$$

and

$$\begin{aligned} Bb &= [B_{\cdot I_1}, B_{\cdot I_2}, \ldots, B_{\cdot I_p}] \begin{bmatrix} b_{I_1} \\ b_{I_2} \\ \vdots \\ b_{I_p} \end{bmatrix} \\ &= B_{\cdot I_1}b_{I_1} + B_{\cdot I_2}b_{I_2} + \cdots + B_{\cdot I_p}b_{I_p} \in R^{k}. \end{aligned} \tag{5}$$

Now if we convert the inequality constraints in (2) to equality constraints by adding slack variables, multiplying the ith identity matrix coefficient matrix of the slack variables of the ith entity by its privately held random matrix $B_{\cdot I_i}$, as pointed out in [8], would reveal $B_{\cdot i}$. However, a simple trick can be exploited here to overcome this difficulty. Note the fact that, in simplex algorithm for linear program, we normally convert the inequality constraints, say, for example,

$$\begin{cases} 2x_1 - 3x_2 \le 3 \\ 4x_1 + 5x_2 \le 7 \end{cases} \tag{6}$$

to equality constraints as

$$\begin{cases} 2x_1 - 3x_2 + x_3 = 3 \\ 4x_1 + 5x_2 + x_4 = 7 \\ x_3 \ge 0, \quad x_4 \ge 0. \end{cases}$$

However, it is not necessary to do exactly in this way. The inequality constraints (6) can also be converted to equality constraints as, e.g.,

$$\begin{cases} 2x_1 - 3x_2 + 1.73468x_3 = 3, \\ 4x_1 + 5x_2 + 100.7683x_4 = 7 \\ x_3 \ge 0, \quad x_4 \ge 0. \end{cases}$$

In general, the inequality constraints (6) can be converted to equality constraints as

$$\begin{cases} 2x_1 - 3x_2 + d_1x_3 = 3 \\ 4x_1 + 5x_2 + d_2x_4 = 7 \\ x_3 \ge 0, \quad x_4 \ge 0, \end{cases} \tag{7}$$

where d_1, d_2 are arbitrarily chosen positive numbers.

Based on the discussion above, we continue as follows. Each entity i, $i = 1, \ldots, p$, chooses its own privately held random diagonal matrix $D_{I_i} \in R^{m_i \times m_i}$, corresponding to the index set I_i. The elements of D_{I_i} are randomly chosen positive real numbers. Denoted by D, the diagonal matrix $D = diag(D_{I_1}, D_{I_2}, \ldots, D_{I_p}) \in R^{m\times m}$ and introduce slack variables $x_s = (x_{m+1}, x_{m+2}, \ldots, x_{m+n})^T \in R^m$, the associated linear program in stand form is

$$\begin{aligned} &min \;\; c^T x \\ &s.t. \; Ax + Dx_s = b, \quad x, x_s \ge 0. \end{aligned} \tag{8}$$

Proposition 1 *If* (x^*, x_s^*) *is an optimal solution to linear program* (8)*, then* x^* *is optimal for linear program* (2).

Proof Note that (x^*, x_s^*) is optimal (and hence feasible) for linear program (8), $Ax^* + Dx_s^* = b$, $x^* \ge 0$, $x_s^* \ge 0$. Thus $x^* \ge 0$ and $Ax^* = b - Dx_s^* \le b$ (since $x_s^* \ge 0$ and D is a diagonal matrix with positive diagonal elements), and so x^* is feasible for linear program (2). To show that x^* is actually optimal for linear program (2), let x be any feasible solution for linear program (2), i.e., $Ax \le b$ and $x \ge 0$. Then x and $x_s = D^{-1}(b - Ax)$ are feasible for Linear program (8). Since (x^*, x_s^*) is optimal for linear program (8), it must be that $c^T x^* \le c^T x$, and so x^* is optimal for linear program (2). □

Conversely, if x^* is an optimal solution to linear program (2), then there is a vector x_s^* such that (x^*, x_s^*) is optimal for linear program (8). Thus, the linear program (8) is also a kind of standard form associated with linear program (2). Based on these discussions our original linear program (2) is transformed into the following secure linear program:

$$\begin{aligned} &min\ \ c^T x \\ &s.t.\ BAx + BDx_s = Bb, \quad x, x_s \geq 0. \end{aligned} \tag{9}$$

Note that the rank of the random matrix B of (3) is m, thus the following equivalence is obvious:

$$Ax + Dx_s = b \Leftrightarrow BAx + BDx_s = Bb. \tag{10}$$

Consequently the feasible region of the original linear program (8) is equivalent to the feasible region of the secure linear program (9). Since both objective functions are the same, it follows immediately that both problems have the same solution set. Thus, we obtain the following result.

Proposition 2 *Let $k \geq m$ for the random matrix $B \in R^{k\times m}$ of* (3)*. The secure linear program* (9) *is solvable if and only if the linear program* (8) *is solvable in which case the solution sets of the two linear programs are identical.*

In next section we describe an explicit implementation of the secure linear programming formulation (9).

3 Formulation of the privacy-preserving algorithm

Starting with the linear program (2) that is partitioned among p entities as described in Sect. 2, the following algorithm generates a solution to the linear program without disclosing any of the privately held data.

Algorithm 1

Step 1. 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 (3).

Step 2. Each entity generates its own privately held random matrix $B_{\cdot I_i} \in R^{k\times m_i}$, $i = 1, \ldots, p$, where m_i is the number of rows held by entity i which results in

$$B = [B_{\cdot I_1}, B_{\cdot I_2}, \ldots, B_{\cdot I_p}] \in R^{k\times m}. \tag{11}$$

Step 3. Each entity generates its own privately held random matrix $D_{I_i} \in R^{m_i\times m_i}$, $i = 1, \ldots, p$, where m_i is the number of rows held by entity i which results in

$$D = diag(D_{I_1}, D_{I_2}, \ldots, D_{I_p}) \in R^{m\times m} \tag{12}$$

Step 4. Each entity i makes public only its matrix product $B_{\cdot I_i} A_{I_i\cdot}$, $B_{\cdot I_i} D_{I_i}$ and $B_{\cdot I_i} b_{I_i}$. These products do not reveal either $A_{I_i\cdot}$, D_{I_i} and b_{I_i} but allow the public computation of the full constraint matrix needed for the secure linear program (9):

$$BA = B_{\cdot I_1} A_{I_1 \cdot} + B_{\cdot I_2} A_{I_2 \cdot} + \cdots + B_{\cdot I_p} A_{I_p \cdot} \in R^{k \times n}, \tag{13}$$

$$BD = B_{\cdot I_1} D_{I_1} + B_{\cdot I_2} D_{I_2} + \cdots + B_{\cdot I_p} D_{I_p} \in R^{k \times m} \tag{14}$$

and the right hand side for (9):

$$Bb = B_{\cdot I_1} b_{I_1} + B_{\cdot I_2} b_{I_2} + \cdots + B_{\cdot I_p} b_{I_p} \in R^{k} \tag{15}$$

Step 5. A public optimal solution vector (x, x_s) to the secure linear program (9) is obtained which, by Proposition 2, x solves the original linear program (2).

Remarks By algorithm, the solution vector x obtained is publicly available. However, it is impossible to compute $A_{I_i \cdot}, b_{I_i}$ without knowing $B_{\cdot I_i}$ and b_{I_i}. Hence, all entities share the publicly computed optimal solution, but without revealing their privately held data.

4 An illustrative example

Consider the linear program

$$\begin{aligned} min\ z &= -2x_1 - 3x_2 - 4x_3 \\ s.t.\quad x_1 + x_2 + 2x_3 &\le 2 \\ x_1 + 4x_2 - x_3 &\le 1 \\ x_1 + 2x_2 - 4x_3 &\le 1 \\ x_i \ge 0, \quad i &= 1, \ldots, 6. \end{aligned} \tag{16}$$

It can be shown that the optimal solution to (16) is $x^* = (0, 0.4444, 0.7778)^T$.

Let $I_1 = \{1, 2\}$, $I_2 = \{3\}$, i.e.,

$$A_{I_1 \cdot} = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 4 & -1 \end{bmatrix},$$
$$A_{I_2 \cdot} = [1 \quad 2 \quad -4],$$
$$b_{I_1 \cdot} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, b_{I_2} = [2].$$

Entity 1 generates its own privately held random matrixes

$$B_{\cdot I_1} = \begin{bmatrix} 0.4562 & 0.1367 \\ 0.5469 & 0.8739 \\ 0.5633 & 0.7693 \end{bmatrix} \in R^{3 \times 2},$$
$$D_{I_1} = \begin{bmatrix} 12.4965 & 0 \\ 0 & 25.6433 \end{bmatrix} \in R^{2 \times 2}$$

and makes public only its matrix product $B_{\cdot I_1} A_{I_1 \cdot}$, $B_{\cdot I_1} D_{I_1}$ and $B_{\cdot I_1} b_{I_1}$.

Entity 2 choose its own privately held random matrices

$$B_{\cdot I_2} = \begin{bmatrix} 0.3574 \\ 0.7763 \\ 0.6682 \end{bmatrix} \in R^{3\times 1},$$
$$D_{I_2} = [15.7628] \in R^{1\times 1}.$$

and makes public only its matrix product $B_{\cdot I_2} A_{I_2 \cdot}$, $B_{\cdot I_2} D_{I_2}$ and $B_{\cdot I_2} b_{I_2}$.

These products do not reveal either $A_{I_i \cdot}$, D_{I_i} and b_{I_i}, $i = 1, 2$ but allow the public computation of the full constraint matrix and the right hand side needed for the secure linear program. Thus, introducing the slack variable vector $x_s = (x_{m+1}, \dots, x_{m+n})^T$ we get the secure linear program associated with the program (16) below

$$\begin{aligned} &min\ z = -2x_1 - 3x_2 - 4x_3 \\ &s.t.\ \ 0.9503x_1 + 1.7178x_2 - 0.6539x_3 + 5.7009x_4 + 3.5054x_5 + 5.6336x_6 = 1.4065 \\ &\quad 2.1971x_1 + 5.5951x_2 - 2.8853x_3 + 6.8343x_4 + 22.4097x_5 + 12.2367x_6 = 2.7440 \\ &\quad 2.0008x_1 + 4.9769x_2 - 2.3155x_3 + 7.0393x_4 + 19.7274x_5 + x_6 10.5327 = 2.5641 \\ &\quad x_i \geq 0, \quad i = 1, \dots, 6. \end{aligned}$$

The optimal solution to secure linear program is

$$\bar{x} = (0.0000, 0.4444, 0.7778, 0.0000, 0.0000, 0.2044)^T.$$

The sub-vector of $\bar{x}$ of the first three components is the correct optimal solution $x^* = (0, 0.4444, 0.7778)^T$ to the original problem. The solution vector obtained is publicly available. On the other hand, no entity i, $i = 1, 2$, reveals its data matrix $A_{I_i \cdot}$ and its right hand side vector b_{I_i}.

Acknowledgments The authors are grateful to anonymous referees for their constructive comments and suggestions, which have improved the presentation of this paper. Project partially supported by the National Natural Science Foundation of China (Grant No. 61003194) and National Natural Science Foundation of China (Grant No. 11171316).

References

1. Du, W.: A study of several specific secure two-party computation problems. PhD thesis, Purdue University, Indiana (2001)
2. Chen, K., Liu, L.: Privacy preserving data classification with rotation perturbation. In: Proceedings of the fifth international conference of data mining (ICDM'05), pp. 589–592. IEEE (2005)
3. Li, J., Atallah, M.: Secure and private collaborative linear programming. In: Proceedings of international conference on collaborative computing, pp. 1–8 (2006)
4. Vaidya, J.: Privacy-preserving linear programming. In: Proceedings of the 2009 ACM symposium on Applied Computing, pp. 2002–2007. ACM, New York (2009)
5. Vaidya, J.: A secure revised simplex algorithm for privacy-preserving linear programming. In: International conference on advanced information networking and applications, pp. 347–354 (2009)

6. Bednarz, A., Bean, N., Roughan, M.: Hiccups on the road to privacy-preserving linear programming. In: Proceedings of the 8th ACM workshop on privacy in the electronic society, pp. 117–120 (2009)
7. Mangasarian, O.L.: Privacy-preserving linear programming. Optim. Lett. **5**, 165–172 (2011)
8. Mangasarian, O.L.: Privacy-preserving horizontally partitioned linear programs. Optim. Lett. (2011). doi:10.1007/s11590-010-0268-9
9. Feng, X., Zhang, Z.: The rank of a random matrix. Appl. Math. Comput. **185**, 689–694 (2007)